НПК МНІС ІП–2019

ЗБІРНИК НАУКОВИХ ПРАЦЬ МОЛОДИХ НАУКОВЦІВ І СТУДЕНТІВ

ЧАСТИНА 1

ПРИСВЯЧУЄТЬСЯ 30-РІЧЧЮ КАФЕДРИ КІБЕРБЕЗПЕКИ ТА КОМП'ЮТЕРНИХ СИСТЕМ І МЕРЕЖ ХМЕЛЬНИЦЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ

КбКСМ ХНУ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Хмельницький національний університет

Військовий інститут Київського національного університету ім.Тараса Шевченка

ПВНЗ “Університет економіки і підприємництва”

Вінницький національний технічний університет

Тернопільський національний економічний університет

Інтелектуальний потенціал - 2019

збірник наукових праць молодих науковців і студентів

Присвячується 30-річчю кафедри кібербезпеки та комп’ютерних систем і мереж Хмельницького національного університету

сформовано за матеріалами
Всеукраїнської науково-практичної конференції
молодих науковців і студентів «Інтелектуальний потенціал – 2019»

20-22 листопада 2019р.

Частина 1
Комп’ютерні системи та кібербезпека

Хмельницький
2019

ББК 74.480.278
С.88

«Інтелектуальний потенціал – 2019» - збірник наукових праць молодих науковців і студентів з нагоди 30-річчя кафедри кібербезпеки та комп'ютерних систем і мереж ХНУ/Колектив авторів – Хмельницький: ПВНЗ УЕП, 2019. – Ч.1: Комп'ютерні системи та кібербезпека. – 100 с.

ЗМІСТ

Кафедра кібербезпеки та комп'ютерних систем і мереж – 30 років в Хмельницькому національному університеті

В 2019 році колектив кафедри кібербезпеки та комп'ютерних систем і мереж Хмельницького національного університету відзначає тридцятирічний ювілей від часу заснування кафедри, яка за час існування пройшла досить великий шлях реорганізацій та перетворень і досягла на цьому шляху значних успіхів в різних напрямках діяльності.

Історію спеціалізованих кафедр з інформаційних технологій було започатковано в ХНУ у 1989 році створенням кафедри електронно-обчислювальних систем, яка нині носить назву кафедри кібербезпеки та комп'ютерних систем і мереж. Кафедру було створено при механічному факультеті. Першим завідувачем кафедри став кандидат технічних наук, доцент Бардаченко Віталій Феодосійович. На той час на кафедрі працювали 1 доцент та 3 асистенти. Під керівництвом Бардаченко В. Ф. викладачі кафедри проводили наукові дослідження в напрямку розробки і впровадження таймерних розрядно-аналогових обчислювальних пристроїв.

У 1991 році кафедра була включена до складу новоствореного факультету радіоелектроніки. Практично в той самий час доктор технічних наук, професор Бардаченко В. Ф. отримав пропозицію очолити власний науковий напрямок в Інституті кібернетики ім. В. М. Глушкова, що зумовило його переїзд у 2002 році до м. Київ, де він в подальшому обійняв посаду директора Центру таймерних обчислювальних систем Інституту кібернетики ім. В. М. Глушкова.

В жовтні 1992 року кафедру електронно-обчислювальних систем очолив кандидат технічних наук, старший науковий співробітник Локазюк Віктор Миколайович. Він поклав початок розвитку при кафедрі нової наукової школи, яка зайнялась питаннями технічної діагностики засобів сучасної обчислювальної техніки.

1996 рік в історії кафедри електронно-обчислювальних систем відзначився зміною її назви на кафедру комп'ютерних систем.

За час існування кафедри комп'ютерних систем на ній було ліцензовано відкриття магістратури зі спеціальності "Комп'ютерні системи та мережі", а також відкриття нової спеціальності "Системне програмування". Це зумовило значне збільшення колективу кафедри і станом на початок 2004 року на кафедрі працювало більше двадцяти співробітників. В зв'язку з цим, в ході реорганізації структури Хмельницького національного університету, у травні 2004 року кафедру комп'ютерних систем було розділено на дві кафедри шляхом відокремлення від неї нового підрозділу - кафедри системного програмування. Кафедру комп'ютерних систем в ході реорганізації перейменували в кафедру комп'ютерних систем та мереж. На основі двох зазначених кафедр та кафедри фізики було створено факультет комп'ютерних систем та програмування (вересень 2004 року), що увійшов до складу інституту телекомунікаційних і комп'ютерних систем Хмельницького національного університету.

Після розподілу кафедр доктор технічних наук, професор Локазюк В.М. очолив новостворену кафедру системного програмування, а виконуючим обов'язки завідувача кафедри комп'ютерних систем та мереж було призначено

кандидата технічних наук, доцента кафедри Хмельницького Юрія Владиславовича.

У 2005 році було проведено конкурс на заміщення посади завідувача кафедри комп'ютерних систем та мереж і кафедру очолив доктор технічних наук, професор Мясіщев Олександр Анатолійович.

Під головуванням Мясіщева О. А. на кафедрі було започатковано новий науковий напрямок - дослідження в області побудови складових комп'ютерних мереж на базі стека протоколів TCP/IP.

В 2017 році на кафедрі відкрито нову спеціальність "Кібербезпека" і кафедра отримала нову назву – кафедра кібербезпеки та комп'ютерних систем і мереж.

З вересня 2019 року за результатами проходження конкурсу завідувачем кафедри став кандидат технічних наук, доцент Кльоц Юрій Павлович, який починав своє знайомство з кафедрою у 1993 році студентом першого курсу.

На сьогоднішній день професорсько-викладацький склад кафедри кібербезпеки та комп'ютерних систем і мереж налічує 12 чоловік, в тому числі 2 професора з науковим ступенем доктор технічних наук іі 9 доцентів з науковим ступенем кандидат технічних наук.

Завдяки наполегливій і цілеспрямованій роботі співробітникам кафедри вдалося досягти значних успіхів в наукових дослідженнях: захищені 3 дисертації на здобуття наукового ступеня доктора технічних наук (Бардаченко В.Ф., Локазюк В.М., Петренко О.М.) та 11 дисертацій на здобуття наукового ступеня кандидата технічних наук.

Велика увага на кафедрі приділяється підтримці науково-дослідної роботи студентів.

Студенти кафедри, які досягли кращих результатів у навчанні та в студентській науковій роботі продовжують навчання в магістратурі та аспірантурі. На сьогоднішній день 12 випускників кафедри захистили дисертації на здобуття наукового ступеня кандидата наук (кандидати технічних наук Чешун В.М., Глущак С.В., Чорненький В.І, Кльоц Ю.П., Медзатий Д.М., Говорущенко Т.О., Лисенко С.М., Гнатчук Є.Г., Тітова В.Ю., Муляр І.В., Джулій А.В., кандидат економічних наук Більовський К.Е.). В 2018р. випускниця кафедри Говорущенко Т.О. першою захистила дисертацію на здобуття наукового ступеня доктора наук.

Для підтримки студентської наукової роботи на кафедрі кібербезпеки та комп'ютерних систем і мереж за роки її існування започатковане проведення трьох Всеукраїнських студентських олімпіад за підтримки МОН України.

У квітні 2002 року колективом кафедри впреше було проведено Всеукраїнську студентську олімпіаду з навчальної дисципліни "Технічна діагностика обчислювальних пристроїв та систем". Загалом в Хмельницькому національному університеті було проведено шість Всеукраїнських студентських олімпіад з навчальної дисципліни "Технічна діагностика обчислювальних пристроїв та систем" після чого, у відповідності до затверджених Міністерством освіти і науки України правил організації і проведення студентських олімпіад, право проведення відповідних Всеукраїнських студентських олімпіад було передано Національному технічному університету "Харківський політехнічний інститут".

За ідеологічної, технічної та спонсорської підтримки приватної компанії “OPEN SYSTEM” в 2006 році кафедрою започатковане проведення Всеукраїнської студентської олімпіади з навчальної дисципліни “Програмування мікропрограмних автоматів та мікроконтролерних систем”. Протягом 2006-2008 років було проведено шість Всеукраїнських студентських олімпіад з зазначеної навчальної дисципліни – 3 представницьких та 3 заочних олімпіади з використанням ресурсів мережі Internet. З 2009 по 2012 рік право проведення відповідних Всеукраїнських студентських олімпіад було передано Національному технічному університету України “Київський політехнічний інститут”, а з 2016 по 2018 рік олімпіада знову проходила в Хмельницькому національному університеті на базі кафедри кібербезпеки та комп’ютерних систем і мереж.

З 2009 року на кафедрі було започатковано ще один новий проект – Всеукраїнську студентську олімпіаду з навчальної дисципліни “Комп’ютерні мережі”, яка проводилася колективом кафедри, згідно з положенням МОН України, протягом трьох років.

З 2015 року на кафедрі щорічно проводяться Всеукраїнські конкурси наукових робіт студентів: 2015-2016 роки конкурси зі спеціальності "Інформаційні технології", з 2017 року"Інформаційні технології та системи". конкурси зі спеціальностей "Комп’ютерна інженерія" та "Інформаційні технології та системи".

Для презентації наукових досягнень і апробацій результатів молодими науковцями на кафедрі проведено цілий ряд Всеукраїнських науково-практичних конференцій: ”Інтелектуальний потенціал молоді в науці та техніці” 2009р., «Інформаційні технології і системи очима молоді» 2011р., «Інноваційні технології для EURO-2012» 2012р., «Молоді таланти в інформатизації суспільства» 2012р., «Інноваційні ідеї молодих вчених» 2013р., «Інтелектуальний потенціал» 2016, 2018 та 2019р.

Всеукраїнську науково-практичну конференцію молодих науковців та студентів «Інтелектуальний потенціал - 2019» присвячено 30-річчю кафедри кібербезпеки та комп’ютерних систем і мереж Хмельницького національноого університету. Для презентації власних доповідей на конференцію зареєструвалися більше 100 учасників, серед яких 46 доповідей буде зроблено в Хмельницькому національному університеті учасниками-гостями з 27-и Вищих навчальних закладів України.

Підсумки.

Протягом тридцяти років існування кафедри кібербезпеки та комп’ютерних систем і мереж її колектив пройшов складний шлях від маленького новоствореного підрозділу Хмельницького технологічного інституту до підрозділу Хмельницького національного університету з одним із найвищих показників за відсотком співробітників з науковими ступенями. За цей час працівниками кафедри реалізовано багато успішних наукових проектів і, без сумніву, її колектив має потужний потенціал для подальших творчих звершень.

Інформацію підготував Чешун В.М.

Моніторинг доступності ресурсів мережі

Авінов Д.С.
Науковий керівник – к.т.н., доц. Кльоц Ю.П.
Хмельницький національний університет

Постійний контроль за роботою локальної мережі, що становить основу будь–якої корпоративної мережі, необхідний для підтримки її в працездатному стані. Контроль – це перший етап, який повинен виконуватися при управлінні мережею. Зважаючи на важливість цієї функції її часто відокремлюють від інших функцій систем управління і реалізують спеціальними засобами. Такий поділ функцій контролю і власне управління корисно для невеликих і середніх мереж, для яких установка інтегрованої системи управління економічно недоцільна. Використання автономних засобів контролю допомагає адміністратору мережі виявити проблемні ділянки мережі, а їх відключення або реконфігурацію він може виконувати в цьому випадку вручну.

Процес контролю роботи мережі зазвичай ділять на два етапи – моніторинг і аналіз.

На етапі моніторингу виконується більш проста процедура – процедура збору первинних даних про роботу мережі: статистики про кількість циркулюючих в мережі кадрів і пакетів різних протоколів, стан портів концентраторів, комутаторів і маршрутизаторів і т. п.

Далі виконується етап аналізу, під яким розуміється більш складний і інтелектуальний процес осмислення зібраної на етапі моніторингу інформації, зіставлення її з даними, отриманими раніше, і вироблення припущень про можливі причини сповільненої або ненадійної роботи мережі.

Завдання моніторингу вирішуються програмними і апаратними вимірниками, тестерами, мережевими аналізаторами, вбудованими засобами моніторингу комунікаційних пристроїв, а також агентами систем управління. Завдання аналізу вимагає більш активної участі людини і використання таких складних засобів, як експертні системи, що акумулюють практичний досвід багатьох мережевих фахівців.

Всі засоби моніторингу та аналізу мереж, можна розділити на кілька великих класів:

Системи управління мережею (Network Management Systems) – централізовані програмні системи, які збирають дані про стан вузлів і комунікаційних пристроїв мережі, а також дані про трафік в мережі. Ці системи не тільки здійснюють моніторинг і аналіз, а й виконують в автоматичному чи напівавтоматичному режимі управління мережею – включення і відключення портів пристроїв, зміна параметрів мостів адресних таблиць мостів, комутаторів і маршрутизаторів і т.п. Прикладами

систем управління можуть служити популярні системи HPOpenView, SunNetManager, IBMNetView.

Засоби управління системою (System Management). Засоби управління системою часто виконують функції, аналогічні функціям систем управління, але відносно інших об'єктів. У першому випадку об'єктом управління є програмне і апаратне забезпечення комп'ютерів мережі, а у другому – комунікаційне устаткування. Разом з тим, деякі функції цих двох видів систем управління можуть дублюватися, наприклад, засоби управління системою можуть виконувати найпростіший аналіз мережевого трафіку. До найбільш відомих систем управління системами відносяться LANDesk, IBM Tivoli, Microsoft Systems Management Server, HP OpenView, Novell ZENworks і CA Unicenter.

Вбудовані системи діагностики і управління (Embedded Systems). Ці системи виконуються у вигляді програмно–апаратних модулів, які встановлюються в комунікаційне обладнання, а також у вигляді програмних модулів, вбудованих в операційні системи. Вони виконують функції діагностики і управління тільки одним пристроєм, і в цьому їх основна відмінність від централізованих систем управління. Прикладом засобів цього класу може служити модуль управління концентратором Distrebuted 5000, реалізує функції автосегментації портів при виявленні несправностей, приписування портів внутрішнім сегментам концентратора і деякі інші. Як правило, вбудовані модулі управління також виконують роль SNMP–агентів, які поставляють дані про стан пристрою системам управління.

Аналізатори протоколів (Protocolanalyzers). Представляють собою програмні або апаратно–програмні системи, які обмежуються на відміну від систем управління лише функціями моніторингу і аналізу трафіку в мережах. Хороший аналізатор протоколів може захоплювати і декодувати пакети великої кількості протоколів, що застосовуються в мережах – зазвичай кілька десятків. Аналізатори протоколів дозволяють встановити деякі логічні умови для захоплення окремих пакетів і виконують повне декодування захоплених пакетів, тобто показувати в зручній для користувача формі вкладеність пакетів протоколів різних рівнів один в одного з розшифруванням змісту окремих полів кожного пакета.

Відповідно до рекомендацій ISO можна виділити такі функції засобів управління мережею:

Управління конфігурацією мережі – полягає в конфігурації компонентів мережі, включаючи їх місце розташування, мережні адреси і ідентифікатори, управління параметрами мережевих операційних систем, підтримку схеми мережі: також ці функції використовуються для іменування об'єктів.

Обробка помилок – це виявлення і усунення наслідків збоїв у роботі мережі.

Аналіз продуктивності – допомагає на основі накопиченої статистичної інформації оцінювати час відповіді системи і величину трафіка, а також планувати розвиток мережі.

Управління безпекою – включає в себе контроль доступу та збереження цілісності даних. У функції входить процедура аутентифікації, перевірки привілеїв, підтримка ключів шифрування, управління правами. До цієї ж групи можна віднести важливі механізми управління паролями, зовнішнім доступом, з'єднання з іншими мережами.

Облік роботи мережі – включає реєстрацію і управління використовуваними ресурсами і пристроями. Ця функція оперує такими поняттями як час використання і плата за ресурси.

Типовими представниками засобів управління мережами є системи HPOpenView, SunNetManager і IBMNetView.

Останнім часом в області систем управління спостерігаються дві досить чітко виражені тенденції:

– інтеграція в одному продукті функцій управління мережами і системами;

– розподіленість системи управління, при якій в системі існує кілька консолей, які збирають інформацію про стан пристроїв і систем та видають керуючі дії.

Нині найуспішнішим сімейством стандартів є SNMP. Він лідирує за кількістю керованих систем (агентів). Керуючі системи (менеджери) зазвичай підтримують безліч стандартів, тому тут складно говорити про лідерство SNMP.

Майже всі успіхи SNMP пов'язані з особливостями процесу стандартизації в IETF:

– безкоштовні і вільно розповсюджувані;

– легко доступні в електронній формі;

– швидкий розвиток стандартів, продумані етапи стандартизації;

– на всіх етапах ведеться технічна експертиза;

– робочі групи очолюють технічні, а не політичні лідери;

– прототипи систем на основі стандартів демонструють їх придатність.

Протокол SNMP підтримують сотні виробників. Головні переваги – це простота, доступність, незалежність від виробників. Він розроблений для управління маршрутизаторами в мережі Internet і є частиною стека TCP/IP.

На сьогодні існує кілька стандартів на бази даних управляючої інформації для протоколу SNMP. Основними є стандарти MIB–I і MIB–II, а також версія бази даних для вилученого управління RMON MIB. Крім цього існують стандарти для спеціальних пристроїв MIB конкретного типу (наприклад, MIB для концентраторів або MIB для модемів), а також частки MIB конкретних фірм– виробників устаткування.

З перерахованих вище протоколів та стандартів саме протокол SNMP дозволяє розробити систему моніторингу доступності ресурсів мережі, що забезпечує мінімальне навантаження на мережу та вчасне інформування адміністраторів про втрату зв'язку з критичними вузлами.

Перелік посилань

1. Фейт С. TCP / IP. Архитектура. Протоколы. Реализация / Сидни Фейт. – Издательство Лори, 2016. – 424 с.

2. Эделман Д. Автоматизация программируемых сетей/ Джейсон Эделман, Мэтт Осуолт, Скотт С. Лоу. − Издательство : ДМК, 2019. – 616 с.

Контроль цілісності інформації за допомогою хешування

Акатов О.В.
Науковий керівник: ктн. доц. Огнєвий О.В.
Хмельницький національний університет

Більш надійними, ніж методи «парності», «контрольних сум», «циклічного контрольного коду» і «турбо-коду», можуть бути методи, побудовані на використанні односпрямованих криптографічних функцій хешування. Аналіз цілісності окремого об'єкта (тексту, файлу) може бути заснований на обчисленні хешу цього об'єкта за узгодженим алгоритмом і на наступному порівнянні його з початковим хешем об'єкта. Подібний аналіз використовують при синхронізації даних, при архівації, при резервуванні при здійсненні цифрового підпису, а також при інших процедурах.

Однак цей метод вразливий, тому що при навмисному порушенні цілісності інформації, особливо якщо порушення проводиться особою з санкціонованим доступом, може бути замінений і її контрольний хеш.

Окремим сформованим напрямком контролю цілісності даних є реєстрація часу надходження даних, що використовує засоби для виявлення порушення їх цілісності заднім числом – TSP (Time-Stamp Protocol). Цьому напрямку приділено увагу в багатьох роботах: від ранніх до однієї з останніх. Принцип реєстрації даних в цих роботах заснований на формуванні ланцюжка об'єднаних хешів (hash-chain based protocols for time-stamping and secure logging) за схемою, наведеною на рисунку 3, і на закріпленні реєстрації шляхом публікацій, що дозволяє виявляти порушення цілісності даних, вироблених заднім числом [1].

Метод полягає в тому, що реєструючу інформацію піддають хешуванню (отримують її хеш H), потім обчислюють об'єднуючий хеш H, враховуючий h і значення попереднього об'єднаного хеша. Кожен *N*-ий об'єднаний хеш публікують.

Засобам закріплення реєстрації засобів зберігання і обробки у зовнішньому інформаційному середовищі присвячений ряд робіт, в якій запропоновано проводити реєстрацію з використанням n-серверів, з яких в кожному акті реєстрації бере участь до обраних випадковим чином серверів, результати роботи яких аналізуються за заданим алгоритмом. Такий спосіб підвищує стійкість, наприклад, до DDoS-атак [2].

Для реєстрації даних оператору TSP – відповідно до схеми на рисунку 2 – направляють хеш цих даних і отримують у відповідь завірену електронними засобами посилку, що включає реєстрований хеш (хеш реєстрованих даних), час його реєстрації і криптографічу мітку, наприклад відповідний об'єднаний хеш (*H* на рисунку 1).

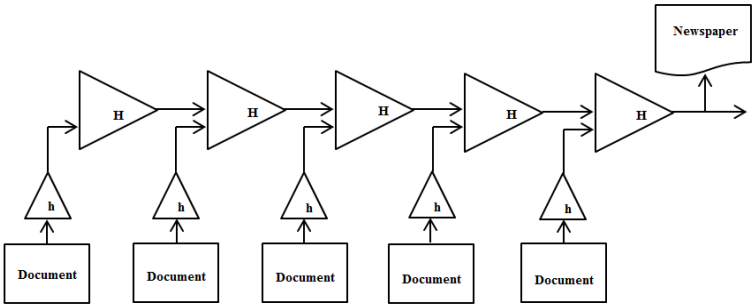

Рисунок 1- Лінійна схема об'єднаних хешів

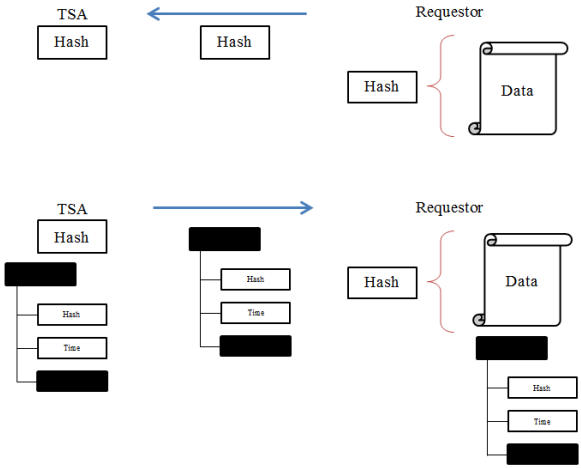

Рисунок 2 - Порядок реєстрації даних

Описаний вище спосіб TSP володіє значним потенціалом, частково використаним в даній роботі. Недоліком способу TSP є те, що контроль цілісності зареєстрованої в спеціальній базі даних інформації, може здійснювати тільки власник цієї БД по зберігаючій у нього системі хешів, а публікуючі окремі хеші (кожен *N*-ий, наприклад) не дають користувачеві можливостей для самостійного контролю цілісності інформації – ними безпосередньо може скористатися лише власник БД, якщо бажає. Якщо ж не бажає, то контроль цілісності виявиться недоступний для користувача – тобто контроль цілісності інформації залежить від волі обмеженого кола осіб, тобто необ'єктивний і вразливий [3].

Функції хешування повинні забезпечувати стиснення даних (отримання хеша), повинні просто обчислюватися і можуть бути безключовими (залежними тільки від повідомлення) або з секретним ключем (залежні від повідомлення і від секретного ключа). До безключових хеш-функцій відносяться коди виявлення змін (modification detection codes, MDC-коди), до яких пред'являються вимоги незворотності (обчислювальна неможливість відновлення даних по їх хешу, односпрямованість), стійкості до колізій першого роду (обчислювальна неможливість знаходження другого повідомлення з хешем, як у даного), стійкість до колізій другого роду (обчислювальна неможливість знаходження пари повідомлень з співпадаючими хешами). Такі хеш-функції називаються криптографічними. Вони оцінюються по відсутності кореляцій між вхідними та початковими бітами, по стійкості до близьких колізій (обчислювальна неможливість колізій, що відрізняються малою кількістю бітів), по стійкості до часткової односпрямованості (обчислювальна неможливість відновити частину початкового повідомлення), по можливості розтягування (можливість хешування коротких повідомлень) і ін. При цьому *n*-бітна хеш-функція вважається крипостійкою, якщо обчислювальна складність знаходження колізій для неї близька до 2, тобто до середнього числа атак «днів народження» для хеша довжиною в n розрядів. Атаки, які не залежать від алгоритму: атака "грубою силою", атака методом "дня народження", повний перебір ключів. При таких атаках вразливі всі алгоритми, єдина можливість протистояти їм - збільшити довжину хеш значення [4].

Список поширених алгоритмів хешування включає: Adler-32, CRC, SHA-1, SHA-2 (SHA-224, SHA-256, SHA-384, SHA-512), HAVAL, MD2, MD4, MD5, N-Hash, RIPEMD -160, Snefru, Tiger (TTH), Whirlpool, IP Internet Checksum (RFC 1071).

Схема алгоритму обчислення значення хеш-функції приведена на рисунку 3, де H_i – *i*-те наближення хеш-функції у вигляді рядка довжиною 256 біт (H_i - довільне); m_i – *i*-ий блок довжиною 256 біт, на які розбитий хешуючий рядок (доповнюється при необхідності нулями); f – крокова

функція хешування, яка відображає два блоки довжиною 256 біт в один блок довжиною 256 біт;

Після застосування функції f до всіх блоків m_i і відповідним проміжним значенням H_i її застосовують до довжини вхідного повідомлення по модулю 2^{256} з H_{n+1} і до контрольної суми $m_1 + m_2 + ... + m_n$ з H_{n+2}. В результаті отримують хеш вхідного повідомлення [5].

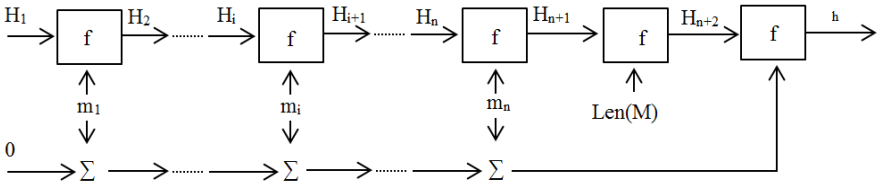

Рисунок 3 - Схема алгоритму хешування

Схема алгоритму хешування є варіантом ітераційного ланцюжка Меркле-Дамгарда, зображеного на рисунку 4.

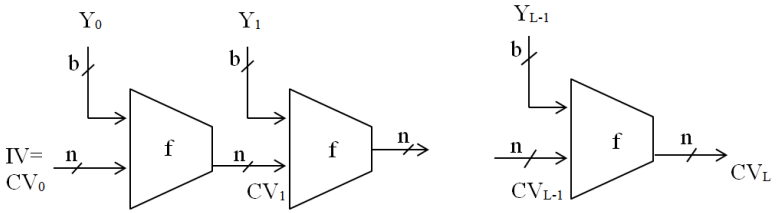

Рисунок 4 - Схема ітераційного ланцюжка Меркле-Дамгарда

Іншим варіантом розвитку цього алгоритму є алгоритм хешування BSA з двома ітераційними ланцюжками, на одного з яких подаються непарні за номером блоки Y_{2n+1}, на які розбитий хешуючий рядок, а на другий – парні Y_{2n}, а хеш отримують як конкатенацію результатів, отриманих за цими двома ланцюжках.

Перелік посилань

1.Петров, А. А. Компьютерная безопасность / А. А. Петров. — Криптографические методы защиты. - М.: ДМК. — 2000. — 448 с: ил.

2.Доля, А. Внутренняя ИТ-безопасность [Электронный ресурс] /А. Доля // Компьютер Пресс № 4'2005 НТМ. Режим доступа: http://www.compress.ru/article.aspx?id=10495&iid=430. - 2005.

3.Леваков, А. Анатомия информационной безопасности США / А. Леваков // Jet Info Информационный бюллетень, 6 (109). - 2002. - С. 29.

4.Суслопаров, А. В. Информационные преступления: диссертация / А.В. Суслопаров. - 2008.

5. Systems and methods for integrity certification and verification of contentconsumption environments: pat. US6931545. / Thanh Ta, Xin Wang. US. - 2000.

Багаторівнева архітектура комп'ютерної мережі

Войцехівський Б.І.
Науковий керівник – к.т.н., доц. Кльоц Ю.П.
Хмельницький національний університет

Комп'ютерна мережа − це сукупність комп'ютерів, які можуть здійснювати інформаційну взаємодію один з одним за допомогою комунікаційного устаткування і програмного забезпечення.

Головною метою об'єднання комп'ютерів в мережу є надання користувачам можливості доступу до різних інформаційних ресурсів і їх спільного використання.

Основною метою застосування багаторівневої архітектури при побудові мережі є забезпечення високої надійності, масштабованості (можливості розширення або перебудови мережі з мінімальними витратами), високої продуктивності.

У загальному випадку в мережах виділяємо такі рівні:

– ядро мережі;

– рівень агрегації;

– рівень доступу.

Приклад трирівневої ієрархічної моделі мережі показано на рисунку 1.

Завдання ядра мережі − високошвидкісна комутація трафіку. Пристрої, що входять до складу ядра мережі, виконують функції:

– високошвидкісну маршрутизацію/комутацію трафіку мережі;

– резервування на рівні апаратури і каналів;

– розділення навантаження по паралельних каналах;

– швидкого перемикання між основним і резервним каналами;

– ефективного використання смуги пропускання з'єднань.

Ядро мережі будується з модулів, утворених одним високопродуктивним пристроєм, із забезпеченням апаратного резервування. Побудова ядра мережі на базі спеціально підібраних комутаторів скорочує час простою мережі, як в разі відмови апаратного (за рахунок гнучких схем резервування), так і в разі програмних помилок або помилок оператора (за рахунок різноманітних механізмів пошуку несправностей).

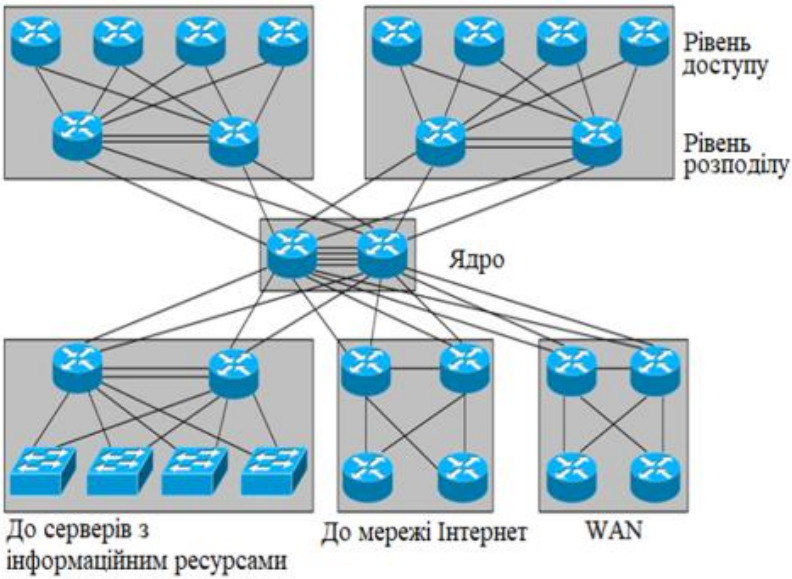

Рисунок 1 – Трирівнева ієрархічна модель побудови мережі

Рівень розподілу виконує сполучну функцію та функцію агрегації трафіку абонентів. Основна вимога до цього рівня − забезпечення резервування та оптимальний розподіл навантаження між паралельними сполуками (як в сторону рівня доступу, так в сторону ядра мережі).

Рівень доступу призначений для підключення робочих станцій і інших периферійних пристроїв (мережевих принтерів і ін.).

На рівні доступу реалізовано управління користувачами та робочими групами при зверненні до ресурсів об'єднаної мережі. Найбільша частина необхідних користувачам мережевих ресурсів повинна бути доступна локально.

Устаткування рівня доступу передбачається в вузлах, що мають не більше двох−трьох напрямків зв'язку (один−два напрямки із суміжними вузлами рівня доступу і один напрямок зв'язку до вузлів вищої ієрархії).

На рівні доступу забезпечується підключення абонентів системи передачі даних (сегментів різних інформаційних і керуючих комплексів, окремих користувачів і інших допоміжних виробничих автоматизованих систем). Устаткування рівня доступу об'єднує високошвидкісні канали локальних мереж з каналами, за якими передається трафік до суміжних вузлів рівня доступу і до вузлів рівня розподілу.

Основне функціональне призначення рівня доступу − підключення необхідної кількості призначених для користувача мережевих пристроїв до

локальної мережі відповідно до необхідних параметрами продуктивності, якості послуг, безпеки і надійності мережевої інфраструктури.

На рівні доступу функціонально вирішують такі основні завдання:

– організація взаємодії з суміжними вузлами рівня доступу і вузлами рівня розподілу,

– маршрутизація трафіку,

– контроль доступу до мережі (фільтрація пакетів),

– виконання інших функцій граничних пристроїв (класифікація, пріоритизація, формування мережевого трафіку, резервування мережевих ресурсів),

– можливість підтримки функціоналу маршрутизаторів.

Пристрої рівня доступу це, як правило, комутатори другого рівня (L2) моделі OSI, тобто без функції маршрутизації. Комутатори здійснюють первинне сегментування мережі (технологія WDM).

Рівень агрегації (розподілу) виконує сполучну функцію та функцію агрегації трафіку абонентів. Основна вимога до цього рівня полягає в забезпеченні резервування і оптимальному розподілі навантаження між паралельними з'єднаннями (як в сторону рівня доступу, так в сторону ядра мережі). Модулі, які використовуються для організації рівня розподілу, зазвичай, організовуються двома аналогічними комутаторами, що функціонують в режимі взаємного резервування.

Рівень розподілу − це точка ізоляції між шаром доступу до мережі та шарами ядра. Він контролює доступ до даних ядра, забезпечує надмірність пристроїв доступу. Перерозподіл маршруту, фільтрування маршрутів та узагальнення маршрутизатора виконуються на рівні розподілу. Розподіл зазвичай є межею між середовищами в мережі. Рівень розподілу забезпечує агрегацію маршрутів, що забезпечують узагальнення маршруту до ядра.

Устаткування рівня ядра передбачається в разі створення великомасштабних мереж з розвиненим рівнем розподілу, що має більше трьох−чотирьох вузлів, з необхідністю об'єднання вузлів рівня розподілу високошвидкісними каналами зв'язку, особливими вимогами по масштабованості (в разі перспективного збільшення кількості вузлів) і забезпечення надійності рівня розподілу.

На рівні ядра функціонально вирішуються такі основні завдання:

– підтримка функцій швидкодіючої комутації (маршрутизації) між окремими вузлами рівня розподілу,

– ізоляція наслідків зміни мережевої топології на рівнях розподілу і доступу,

– агрегація мережевого трафіку з рівня розподілу,

– функції відмовостійкості і балансування навантаження,

– підтримка функціоналу вузлів при застосуванні технології MPLS.

На рівні ядра проводиться тільки високошвидкісна комутація та

маршрутизація мережевого трафіку. Функції, що віднімають обчислювальні ресурси пристроїв ядра або збільшують затримку комутації пакетів, виносяться за межі ядра мережі. Для найбільшої ефективності роботи ядра мережі, а значить і всієї мережі необхідно забезпечення наступних стратегій:

– в ядрі мережі не повинні реалізовуватися мережеві правила,

– кожен пристрій ядра мережі повинен бути придатним для доступу до кожного вузла.

Як устаткування передачі даних рівня ядра використовуються високопродуктивні маршрутизатори, маршрутизовані комутатори (третього рівня).

Вузли рівня ядра організовуються з використанням резервованого обладнання або обладнання з резервованими модулями управління, комутації (маршрутизації) і блоками живлення.

Використання багаторівневої архітектури мережі, на відміну від традиційних, що базуються на використанні одного маршрутизатора та послідовності некерованих комутаторів, дозволяє суттєво зменшити службовий трафік, забезпечити ізоляцію та вищий рівень безпеки вузлів комп'ютерної мережі, значно зменшує часові витрати на пошук місця та визначення причин збоїв в роботі мережі та мережевого обладнання. Перехід від однорангової архітектури мережі до багаторівневої дозволяє оптимізувати використання каналів зв'язку та зменшити їх завантаження службовим трафіком.

Перелік посилань

1. Фейт С. TCP / IP. Архитектура. Протоколы. Реализация / Сидни Фейт. – Издательство Лори, 2016. – 424 с.

2. Эделман Д. Автоматизация программируемых сетей/ Джейсон Эделман, Мэтт Осуолт, Скотт С. Лоу. − Издательство : ДМК, 2019. – 616 с.

3. Куроуз Д. Компьютерные сети. Нисходящий подход. 6-е изд. / Куроуз Д., Росс К. – М.: 2016. — 912 с.

Метод оцінки ефективності керування багаторівневою системою на основі мультиагентного підходу

Єресько В.О.
Науковий керівник – к.т.н., доц. Бойчук В.О
Хмельницький національний університет

Існуюча модель управління БРС має ознаки жорсткої централізації і вузької спеціалізації і не може забезпечити всебічного взаємодії між собою органів управління. Це часто призводить до несвоєчас-ного та неадекватного, а часом нескоординованого реагування на ситуацію, що змінюється різних органів збору, обробки та аналізу інформації. Діяльність таких органів, у

багатьох випадках дублюється, відрізняючись лише тимчасовими і функціональними обмеженнями, що істотно збільшують документообіг, знижують наочність даних діяльності та ускладнють аналіз результативності, а фактори суттєво впливають на результативність діяльності, при оперуванні більшим обсягом інформації в організації, врахувати просто неможливо, тому що часові витрати на обробку інформації вручну потребують залучення додаткових ресурсів, а так само інформації, яка надходить не щодня, а відповідно до регламентів, які були збудовані ще для ситуації, коли кожен структурний підрозділ представляв собою автономну одиницю.

Розглянемо існуючу модель БРС (рис. 1) в контексті сформованої структури, що представляє собою сукупність органів управління, структурних одиниць.

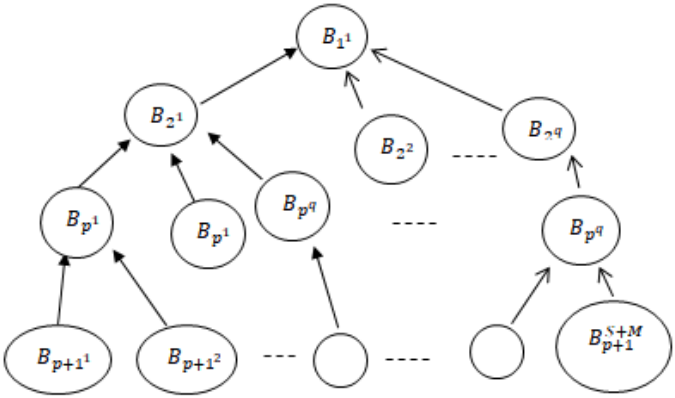

Рисунок 1– Ієрархічна система управління БРС

Будемо вважати, що: кожен орган може приймати рішення; рішення органів можна характеризувати кінцевим набором змінних, що приймають 25 числові значення; прийняття рішення органом зводиться до вибору деяких числових значень змінних з безлічі їх допустимих значень; якість прийнятих рішень оцінюється кінцевим числом показників ефективності. Тоді, ієрархічна модель управління складатиметься з приймаючих рішення органів B_p^q.

Нехай ОСО і АІ B_p^q має багато стратегій Y_p^q, елементами y_p^q у яких є точки, що відповідають різним допустимим рішенням. Пунктиром позначимо підпорядкованість ОСО і АІ територіально віддалених від вищих рівнів ієрархії. Безліч стратегій підпорядкованих органів визначається множинами стратегій вищих органів, а саме, якщо ОСО та АІ B_p^q підпорядковані органи B_{p+1}^S, B_{p+1}^{S+1}, …, B_{p+1}^{S+M} , То на прямому утворені множин, їх стратегій $Y_{p+1}^S\, x$,…, $Y_{p+1}^{M+S}\, x$ визначена вектор функція f_p^q, областю значень якої є безліч стратегій органу B_p^q за вибором оптимального управлінського рішення.

Тим самим набір стратегій органів нижнього рівня СМК Y_K^q, q = 1, ..., Q (K), в кінцевому рахунку, визначається стратегією Y_1^1 вищого органу системи B_1^1.

Стратегією $A_K^q \subset Y_K^q$ органу нижнього рівня СМК B_p^q є сукупність m точок Y_K^q, $A_K^q = \{(y_K^q)_i\}$, j = 1, ..., m.

Тоді, оптимальним рішенням в моделі СМК є сукупність стратегій A_K^q, q = 1, ..., Q (K) органів нижнього рівня, які забезпечують реалізацію "найкращих" значень вектора показників якості $\tilde{F}(y)$

Для органу B_1^1 на Y_1^1 повинен бути визначений його вектор показників ефективності F(y) характеризує ступінь досягнення цілей якості.

Стратегією $A_K^q \subset Y_K^q$ органу нижнього рівня СМК B_p^q є сукупність m точок Y_K^q, $A_K^q = \{(y_K^q)_i\}$, j = 1, ..., m. При m = 1 стратегія A_K^q складається з одного елемента і називається одноцентровою, при m> 1 - многоцентровою.

Оптимальним рішенням в моделі СМК МОС назвемо сукупність стратегій A_K^q, q = 1, ..., Q (K) органів нижнього рівня, що забезпечують реалізацію "найкращих" значень вектора показників якості $\tilde{F}(y)$.

Розглянемо різні варіанти дій органів управління в СМК БРС:

1. Органи B_K^q, p = 1, ..., K, q = 1, ..., Q (K) нижнього рівня СМК БРС задаються різними допустимими стратегіями якості, вектор-функції f_{K-1}^q, q=1, ..., Q (K -1) визначають стратегії органів вищого рівня системи управління, в кінцевому рахунку, відповідні стратегії Y_1^1 вищого органу системи B_1^1.

Після того, як безліч відповідних стратегій побудовано, оптимізується проста БРС B_1^1 і визначається, оптимальна стратегія Y_1^1; відновлюючи відповідні їй стратегії органів нижнього рівня, отримуємо оптимальне рішення в моделі СМК БРС. Тобто дана стратегія є багаторазовим повторенням через систему різних значень стратегій, прийнятих органами нижнього рівня, визначення оптимальних значень вектор-функції цілей якості F і вибір стратегій нижніх органів, які задають ці значення.

2. Від нижніх рівнів системи до верхніх повинно здійснюватися побудова кордонів множин стратегій якості аж до безлічі Y_1^1 вищого органу БРС B_1^1. Потім, від вищих органів ієрархії до нижчих проводиться прийняття рішень: орган B_1^1 оптимізує свій вектор ефективності F на безлічі Y_1^q і встановлює тим самим відповідні значення показників ефективності функціонування підлеглих йому органів по цілям качсетва, а нижні органи визначають показники ефективності підлеглих їм органів з умов забезпечення встановлених їм значень. В кінцевому рахунку, визначаються стратегії качсетва, а нижчестоящі органи визначають показники ефективності підлеглих їм органів з умов забезпечення встановлених їм значень: В кінцевому рахунку, визначаються стратегії Y_K^q, q = 1, ..., Q (K) органів нижнього рівня, які забезпечують оптимальне значення вектора ефективності вищого органу системи.

Розглянемо випадок, коли вектор-функції f_p^q, p = 1, ..., K, q = 1, ..., Q (p) монотонні, мають однакову для всіх органів розмірність ή і проведена структурна декомпозиція, коли кожна їх компонента залежить лише від відповідних компонент підлеглих органів.

З огляду на монотонності, вимога мінімізації компонент векторфункціі F призводить до вимоги мінімізації (або максимізації) компонент функцій ефективності всіх підлеглих органів; тому можуть розглядатися лише частини кордонів їх множин стратегій (допустимі стратегії - ДС). Якщо позначити через

Таким чином, побудова ДС вищого органу здійснюється шляхом послідовного побудови по ДС підлеглих йому органів і прийнятті рішення вищим органом СМК БРС, яке визначить відповідні рішення нижчестоящих органів.

Система буде замкнута, якщо задати (k -1) невідоме N^v і тоді її рішення визначить оптимальні стратегії всіх органів БРС.

Таким чином, якщо функції ефективності органів мають вигляд (1.6), то прийняття оптимального рішення буде полягати у виборі органом B_1^1 вектора N і незалежному визначенні органами нижнього рівня за своїми ***ДС*** і заданому N^v оптимальних стратегій БРС з рішення системи (1.4). знайдемо сукупність точок простору показників ефективності вищого органу, яким відповідає постійне значення вектора N. Без обмеження спільності, можна вважати, що $q_s^v(y_S^1, \dots, y_S^v) = 1$. Такий сукупністю є поверхню p = 0, що задовольняє системі рівнянь.

Для всіх підлеглих органів БРС одного рівня ціни виявляються однаковими і тому, коли вищестоящий орган може вказати оптимальні ціни N^v, тобто встановити сумарний критерій P, не розглядаючи своєї ДС, то немає необхідності проводити її синтез.

Розглянувши дії органів СО та АІ для БРС представленої моделі СМК і оцінивши їх цільові функції можна зробити висновок, що ефективна робота можлива лише в умовах провідної ролі вищих органів БРС, протее відсутність мотивації інших структурних одиниць зводить нанівець всі його зусилля, у всіх інших структурних одиницях рішення будуть запізнюватися, тому час на первинний аналіз на місці інформації, її передачу вищому органу, прийняття ним рішень і зворотний передачу неприпустимо зросте, а обсяг інформації переданої органу вищого керівництва не дозволить йому оперативно реагувати на виникаючі ситуації.

Отже, необхідно передбачити таку модель збору, обробки та аналізу інформації СМК БРС, в якій стратегії агентів визначені заздалегідь, ціни параметрів якості та ймовірні наслідки прораховані і для кожного ймовірного результату підготовлені стратегії відповідних ОС та ОАІ щодо попередження небажаних ситуацій.

До результатів роботи, що відрізняються науковою новизною, відносяться:

- алгоритм визначення стратегічних цілей розвитку багаторівневої системи в області якості, що дозволяє проводити вибір з усього комплексу альтернатив і критеріїв за рахунок уточнення значення пошуку власного стовпця в матриці парних порівнянь методу аналізу ієрархій;
- дискретна модель системи управління якістю багаторівневої системи, що відрізняється наявністю механізму отримання інформації від агентів, яка має слабку ступінь маніпульованості, що підтверджено результатами проведених досліджень;
- алгоритм вибору оптимальних управлінських рішень в багаторівневих системах, що відрізняється урахуванням оцінок верхніх і нижніх меж в багатовимірних критеріальних матрицях і забезпечує облік різних варіантів розбіжностей між реальними і прогнозованими процесами;
- модель оцінки успішності управління агентами, заснована на використанні модифікованих нечітких множин і дозволяє представити оцінки у вигляді нечітких множин, що характеризують складність процесу і зв'язку між агентами.

Перелік посилань

1. Майкл Вулдрідж, Вступ до мультиагентних систем, John Wiley & Sons Ltd, 2017, м'яка обкладинка, 366 сторінок,
2. Журнал автономних агентів та мультиагентних систем, Видавець: Springer Science + Business Media B.V., раніше Kluwer Academic Publishers B.V.
3. Хосе М. Відаль, Основи багатоагентних систем: з прикладами NetLogo.
4. Субботін С.О., Олійник А.А., Олійник О.О. Неітераційні, еволюційні та багатоагентні методи синтезу нечітких та нейронних мережевих моделей: Монографія / Під назвою Загальні. ред. SO Subbotin. - Запоріжжя: ЗНТУ, 2016. - 375 с.

Мультиагентний підхід в системі управління мобільними ресурсами

Заворотний М.В.
Науковий керівник - к.т.н., доц. Огнєвий О.В
Хмельницький національний університет

Управління мобільними ресурсами підприємств в реальному часі - актуальна і значуща задача, вирішення якої необхідно для широкого кола підприємств. Це передбачає автоматичну оперативну реакцію на незаплановані події з можливістю підбору або заміною ресурсів (наприклад, термінове замовлення, форс-мажорні ситуації, незаплановані ремонтні

роботи, різні відмови з боку інфраструктури). Одночасно при виконанні завдань управління ресурсами підприємства необхідно враховувати різноманітні критерії: максимальна швидкість виконання замовлення, рівномірне завантаження ресурсів, мінімальні ризики зриву термінів замовлень тощо. Це вимагає розробки нових моделей і методів, алгоритмів та програмних засобів, які дадуть можливість оперативно створювати, аналізувати і гнучко перебудовувати ресурси в режимі реального часу. Розвиток систем штучного інтелекту, методів об'єктно-орієнтованого програмування, мережевих технологій призвели до виникнення мультиагентного підходу.

Мультиагентні методи засновані на моделюванні поведінки групи агентів, дозволяють забезпечити пошук оптимальних рішень за короткий час, а також виключити необхідність розрахунку різних обмежень щодо досліджуваних залежностей.

Значний внесок у розвиток теорії і практики мультиагентних систем та технологій внесли такі автори як К.Лангтон, Р.Аксельрод, М. Вулдрідж, Н. Дженнінгс, Р. Хаммонд, Дж. Фербер, В.О. Філатов, В.Б. Тарасов, В.І. Городецький, М. Доріго, К. М. Пассіно, Д.А. Поспелов, О.Н. Гранічін, П.О. Скобелєв, О.М. Швєцов, С.О. Яковлєв та ін.

Однак, в умовах розвитку сучасного інформаційного простору дослідження організаційних і технічних аспектів сучасної інформаційно-комунікаційної сфери управління мобільними ресурсами підприємства лишаються досить актуальними. У зв'язку з складнощами і особливостями управління мобільними ресурсами сучасних підприємств останнім часом розвиток набуває мультианентний підхід до управління такими ресурсами.

Мультиагентий підхід принципово відрізняється від централізованого та схожий з процесами самоорганізації в живій природі та пропонує рішення складних завдань проводити в розподіленому середовищі взаємодіючих між собою агентів, які діють паралельно і асинхронно, спільно вирішуючи складні завдання шляхом самоорганізації [4].

Існує безліч визначень і класифікацій агентних систем, наприклад [1, 4]: під агентом мається на увазі автономна комп'ютерна програма чи програмний модуль, здатна діяти цілеспрямовано, автономно, безперервно і непередбачувано від змін зовнішнього середовища.

Інтелектуальний агент – це програмний модуль, здатний до оперативного аналізу даних, адаптації до умов, що змінюються, активного обміну інформацією з іншими агентами з метою досягнення поставлених користувачем завдань (рис.1).

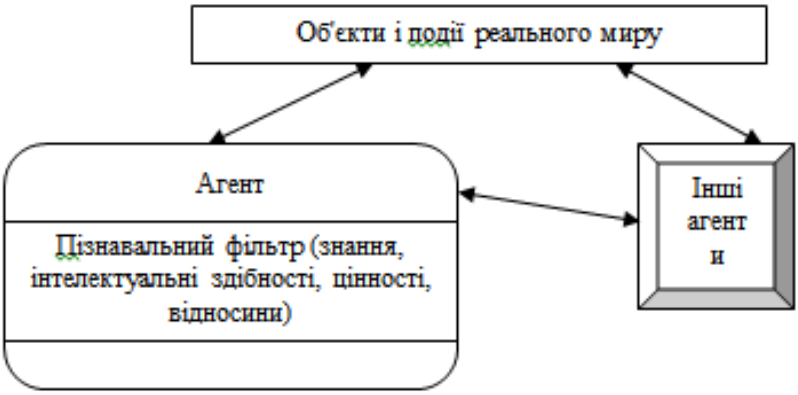

Рисунок 1 – Інтелектуальний агент

ПА можуть мати одну або декілька властивостей, наведених в табл. 1. Однією з основних властивостей, які асоціюються із програмними агентами, є автономність (Autonomy). Агент вважається автономним, якщо він здатний діяти без прямого управління людиною. Поняття автономності включає також наявність в агента декількох цілей, яких він може досягти. Наявність задач також сприяє тому, що агент одержує можливість планувати.

Таблиця 1 – Властивості програмних агентів

Властивість	Визначення
Автономність	Можливість діяти незалежно від користувача
Адаптивність	Здатність до навчання під час роботи
Комунікативність	Здатність до комунікації з користувачем або іншими агентами
Здатність до співробітництва	Працює з іншими агентами для досягнення мети
Персоніфікованість	Поводиться природно (проявляє емоції)
Мобільність	Можливість переміщатися по навколишньому середовищу

Агент необов'язково повинен мати всі перераховані властивості. У більшості випадків використовуються декілька з них.

До основних властивостей ПА можна віднести [3]:

– Автономність (Autonomy) - пов'язана зі здатністю ПА діяти цілеспрямовано для досягнення результату, без зовнішнього управління з боку інших систем та без прямого управління людиною. ПА контролює свої дії, робить висновки залежно від стану навколишнього середовища, має декілька цілей для досягнення.

– Комунікативність (Communicationability) - агенти визначаються соціальною поведінкою, тобто взаємодіють з іншими агентами для виконання спільного завдання, узгоджено вирішують конфлікти для досягнення консенсусу, спілкуються з навколишнім середовищем.

– Реактивність (Reactive) – ПА отримують події із зовнішнього середовища і своєчасно реагують на зміни в ситуації, при цьому їх поведінка коригується відповідно до змін і цілей.

– Проактивність (proactivity) полягає в прагненні агента досягти цілей безперервно покращуючи характеристики внутрішнього стану, проявляти ініціативність.

Можна відмітити такі властивості агента як цілеспрямованість (Goal-oriented), корпоративність (Collaborative), тимчасову безперервність (Temporal continuity), персонефіціюваність (Personality), гнучкість (Flexible), мобільність (Mobility) - можливість переміщення по навколишньому середовищу, адаптивність (Adaptability), запас міркувань або здатність до висновків (Capacity for reasoning), надійність (Trustworthiness).

За типом зв'язків виділяють наступні типи агентів синтагматичні - зв'язки між рівноправними агентами (дволанковий зв'язок), парадигматичні - зв'язки підпорядкування вищестоящому рівню (дволанковий зв'язок); ієрархічні - зв'язки підпорядкування вищестоящому агенту-координатору (багатоланкові зв'язки); субсидіарні - узгоджена дія периферійних агентів, що мають повноваження від центру управління (багатоланкові).

Отже ПА – це програма, яка знаходиться в середовищі, від якого отримує дані про події, що відбуваються, інтерпретує їх і виконує команди, що впливають на це середовище. Це визначення появилося у зв’язку з використанням програм не тільки для вирішення завдань, пов’язаних зі збором, обробкою та структуризацією інформації, але й для швидкого прийняття рішень в системах управління реального часу (real-time systems)

Агент може перебувати в одному із станів, згідно життєвому циклу МАС. Стан агента зазвичай визначається його статусом: ініційований, активний, зупинений, очікує, вилучений тощо.

У сучасних дослідженнях застосовують різні типии агентів, які утворюють мультиагентні або багатоагентні системи (МАС) [3].

МАС будується як система агентів, які можуть здійснювати інформаційну або інтелектуальну взаємодію один з одним за допомогою інформаційної мови ACL (Agent Communication Language). Відмінності між автоматизованими системами і інтелектуальними МАС проілюстровані в табл. 2.

Організаційно структура МАС визначається функціями агентів і нормами їх взаємодії, тому за організаційною структурою МАС виділяють такі типи агентів: агенти-виконавці, які підкоряються агентам-менеджерам, які необхідні для взаємодії всіх агентів та забезпечують доступ агентів до

загальних даних системи і управляє їх життєвим циклом; агенти-координатори, які відповідають за організацію взаємодії агентів; інтерфейсні агенти - для зв'язку із зовнішнім середовищем; канальні агенти - для забезпечення обміну інформацією в МАС [2]. Немає необхідності розглядати всі типи агентів, які можуть існувати в системі. В разі необхідності, в систему можна буде додати необхідні типи агентів.

Таблиця 2 – Автоматизовані і інтелектуальні системи

Ознака	Автоматизовані системи	МАС
Основні характеристики	Передбачуваність, повторюваність, ієрархічна структура	Гнучкість, чуйність, самоорганізація
Механізм досягнення основних характеристик	Задані алгоритми, пам'ять; інтеграція	Здатність робити припущення; знання; навчання, робота в мережі
Основні недоліки	Негнучкість	Ризик здійснення помилки
Механізми подолання недоліків модуляції	Модуляції	Розподіл інтелекту, повноцінне використання наявних знань, вміння вчитися на своєму досвіді
Області застосування	Стабільні середовища, довготривале виробництво, масове виробництво	Непередбачувані середовища, часто змінюється виробництво, індивідуальне виробництво, короткочасні періоди освоєння нової продукції

Архітектура МАС задає взаємодію агентів в системі. Середовищем існування МАС є агентні платформи. За минуле десятиліття було розроблено багато програмних реалізацій агентних платформ, кожна з яких має свої особливості, переваги та недоліки. Найбільш популярними є JADE, FIPA-OS, AOS, ZEUS, KADOMA, NOMADS, ARA, AGLETS, GRASSHOPPER, TRACY, AJANTA, LEAP, JACK, SEMOA. Деякі з них існують у вигляді комерційних проектів (таких як JACK) або проектів, які позиціонують, як проекти з відкритим вихідним кодом (JADE, ZEUS і ін.).

МАС зазвичай складається з наступних основних компонентів [5]:

– множини організаційних одиниць, серед яких підмножина агентів, які маніпулюють підмножиною об'єктів;

– множини завдань;

– середовище, тобто простір, в якому існують агенти і об'єкти;

– множини відносин між агентами;
– множини дій агентів (наприклад, операцій над об'єктами).

МАС автоматизують повний цикл управління ресурсами в реальному часі, включаючи:

– оперативну реакцію на важливі події;
– динамічне планування і адаптивне перепланування замовлень/ресурсів;
– взаємодію з клієнтами, менеджерами і виконавцями для узгодження прийнятих рішень через Інтернет або стільниковий телефон;
– моніторинг виконання побудованих планів і бізнес-процесів замовника;
– перепланування розкладів в разі неузгодженості між планом і фактом.

Отже одним з найбільш перспективних підходів для вирішення поставленого завдання є застосування мультиагентних технологій, найбільш придатних для пошуку балансів інтересів між усіма учасниками, врахування індивідуальних особливостей замовлень та мобільних ресурсів.

Перелік посилань

1. Гриценко В.І. Модель мультиагентної системи для е-бізнеса і технологія її программної реалізації / В. І. Гриценко, А. Я. Гладун, Ю. Д. Журавлев, М. В. Несен // Проблеми програмування. – 2004. – № 2,3. – С. 510–519.

2. Коновалов О. Ю. Агентні технології у розподілених обчислювальних системах / О.Ю. Коновалов // Наукові записки Українського науково-дослідного інституту зв'язку [Електронний ресурс]. – Режим доступу: http://nbuv.gov.ua/UJRN/Nzundiz_2013_2_15. – (дата звернення 18.10.2019).

3. Ковальський П.В. Інформаційна мультиагентна система випробування стійкості алгоритмів шифрування даних / П.В. Ковальський, П.О. Кравець // Вісник Національного університету «Львівська політехніка». – Сер.: Комп'ютерні системи проектування. Теорія і практика. – Львів : Вид-во НУ «Львівська політехніка». – 2008. – № 610. – С. 159-166.

4. Плескач В. Л. Агентні технології / В. Л. Плескач, Ю. В. Рогушина. – К. : КНТЕУ, 2005. – 337 с.

5. Примостка А.О. Особливості розробки та проектування мультиагентних систем / А.О. Примостка // Наукові записки Національного університету «Острозька академія». Серія «Економіка»: збірник наукових праць. – Острог: Видавництво НУ «Острозька академія», 2015. – Вип. 28. – С. 159–163.

Система контент-моніторингу соціальних мереж з питань кібербезпеки

Кальян Н.А., Матіїшин О.Т.
Науковий керівник – д.т.н, професор Ланде Д.В.
Інститут спеціального зв‘язку та захисту інформації
Національного технічного університету України
«Київський політехнічний інститут ім. Ігоря Сікорського»

Актуальність роботи. На цей час, в умовах гібридної війни з розвиненою інформаційною компонентою, на багатьох рівнях керування виникають питання урахування інформації, що з’являється у соціальних медіа. Відомо, що спеціальні тематичні інформаційні потоки є компонентою інформаційних протистоянь, зміст яких спрямований на реалізацію попередньо спланованих інформаційно-психологічних впливів на аудиторію шляхом впливу на установки і поведінку для досягнення заздалегідь визначених цілей. Така інформація, з одного боку, містить багато «шуму», дезінформації, а, з іншого боку, є самою оперативною (повідомлення в мережі Twitter у середньому на 6 годин випереджують повідомлення на веб-сайтах). На цей час існують можливості добування і комп'ютерної обробки даних із соціальних мереж, але до цього часу не існувало доступних бюджетних рішень проблеми цільового інформування корпоративних користувачів на основі інформації із соціальних мереж.

Мета роботи – створення технологічних засад та інструментальних засобів контент-моніторингу соціальних мереж з питань кібербезпеки, побудова діючого макету корпоративної інформаційно-аналітичної системи на основі моніторингу соціальних мереж із максимальним застосуванням компонент відкритого доступу.

Завдання полягає у розв’язанні часткових поставлених задач:

1. Провести аналіз існуючих підходів до агрегації тематичних новин.
2. Запропонувати та обгрунтувати підходи до побудови корпоративної системи контент-моніторингу соціальних мереж.
3. Створити комплекс інструментальних засобів контент-моніторингу соціальних мереж з вибраних питань, зокрема, кібербезпеки.
4. Розробити/адаптувати серверні додатки, що реалізують функції аналізу і прогнозування тематичних інформаційних потоків.
5. Розробити/адаптувати додатки користувачів, що реалізують персоналізацію інформаційного забезпечення.

Об’єкт роботи – методи побудови корпоративних систем інформаційної підтримки прийняття рішень, зокрема, системи контент-моніторингу соціальних мереж з питань кібербезпеки.

Предмет роботи – засоби контент-моніторингу соціальних мереж з питань, що відносяться до вибраної предметної галузі.

Сучасні відкриті мережеві ресурси, веб-сайти, соціальні мережі перетворюється в даний час в основне джерело і ефективний інструмент для конкурентної розвідки. Вони дозволяють в режимі реального часу не тільки відслідковувати дії компаній-конкурентів, але і виявляти останні тенденції по необхідній тематиці [1].

Сучасні методи контент-моніторингу – це адаптація концепції глибинного аналізу текстів (Text Mining) і класичних методів контент-аналізу до умов формування і розвитку динамічних інформаційних масивів, наприклад, потоків інформації в мережі Інтернет [2].

Актуальним підходом, що пропонується у цій роботі, до вирішення проблеми створення такої корпоративної системи є одночасне застосування методів і засобів інформаційного пошуку і агрегування інформаційних потоків. Це обумовило створення системи-контент моніторингу як складову системи підтримки рішень з предметної області "кібербезпека" на основі аналізу динамічних інформаційних текстових потоків з мережевих ЗМІ.

У межах роботи побудовано та досліджено Діючий макет системи контент-моніторингу соціальних мереж, автоматичної обробки динаміки і повних текстів із соціальних мереж за певний період часу пов'язаних із тематикою «кібербезпека», складові якого наведено на Рис. 1. Збирання інформації здійсюється у режимі пошуку в соціальній мережі. Запит (ключова фраза для пошуку у відповідній соціальній мережі) считується програмою із пеціального конфігураційного файлу.

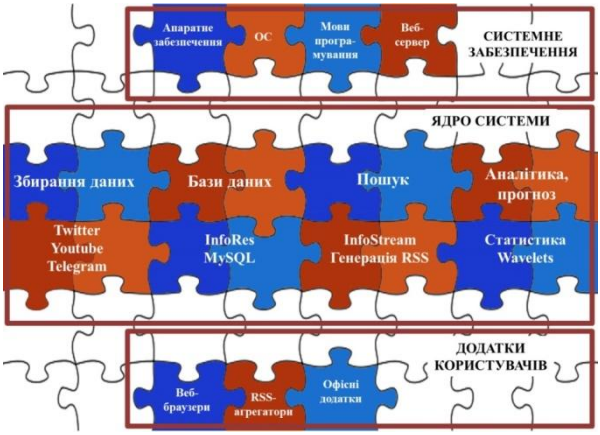

Рисунок 1 - Складові системи контент-моніторингу соціальних мереж

Далі йде здійснення пошуку і виведення записів, що відповідають запитам. Після цього здійснюється запис унікальних записів до БД сервера.

Для контролю (або резервного копіювання) може здійснюватись вивід усіх записів з БД.

Аналіз існуючих підходів до агрегації тематичних новин приводить до необхідності і можливості створення комплексу інструментальних засобів контент-моніторингу соціальних мереж з вибраних питань, зокрема, кібербезпеки.

Макет включає сучасні засоби персоналізації, надання доступу до баз даних в режимі онлайн, у тому числі з мобільних пристроїв, для чого широко застосовуються можливості форматів RSS. Обґрунтовано вибір «готових» програмних компонентів, описані засоби власної розробки (сканери соціальних мереж, засоби формування динамічних RSS-каналів), наведені результати їх інтеграції у єдиний програмно-апаратний комплекс.

У результаті розробки проекту запропоновано та обґрунтовано методику і засоби створення системи контент-моніторингу соціальних мереж за визначеною проблематикою, вибору релевантної інформації із соціальних мереж, реалізацію інформаційно-пошукового механізму їх уточнення користувачами, збереження запитів як RSS-каналів, ведення персональних баз даних у середовищі клієнтських додатків – RSS-агрегаторів (зокрема, FeedDemon 3.5 [3], Feedreader 3.14 [4], RSS Guard 3.4.1 [5]).

Практичне значення отриманих результатів полягає в створенні діючого макету системи контент-моніторингу соціальних мереж з питань кібербезпеки, готового до застосування в якості компоненти систем підтримки прийняття рішень щодо інформаційної і кібернетичної безпеки.

Усі функціональні блоки діючої моделі системи контент-моніторингу соціальних мереж за вибраною тематикою, які відображені на Рисунку 1, можна розглядається з позиції узагальненої моделі контент-моніторингу, а саме: системи збору інформації, інформаційного пошуку, вибіркового розповсюдження, а також аналізу і прогнозування.

Перелік посилань

1. Додонов О.Г., Ланде Д.В., Прищепа В.В., Путятін В.Г. Конкурентна розвідка в комп'ютерних мережах. Київ: ІПРІ НАН України, 2013. 12 с.

2. Додонов А.Г., Ланде Д.В., Циганок В.В. Розпізнавання інформаційних операцій. Київ: Інжиніринг, 2017. 69 с.

3. Bradbury N. The Last Version of FeedDemon is Here, and it's Free. URL: http://nickbradbury.com/2013/06/20/the-last-version-of-feeddemon-is-here-and-its-free/ (дата звернення: 05.02.2019)

4.Feedreader 3.14 released. URL: http://www.feedreader.com/feedreader/releases/3.14 (дата звернення: 05.02.2019)

5. Serea R. RSS Guard 3.3.5. URL: https://www.neowin.net/news/rss-guard-335/ (дата звернення: 05.02.2019)

Оцінка часу передачі даних за інформаційною взаємодією

Кізюн Б.М.
Науковий керівник – к.т.н., доц. Джулій В.М.
Хмельницький національний університет

При оцінці ймовірносно-часових характеристик доставки даних застосовують математичний апарат систем масового обслуговування - СМО. Основний вид взаємодії вузлів - послідовне поетапне. В разі встановленого з'єднання - це ланцюжок сенсорних пристроїв. На рис. 1 приведено графічне позначення СМО.

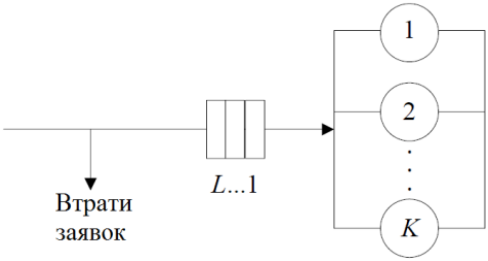

Рисунок 1 - Система масового обслуговування

СМО задається наступними параметрами: $A(t)$ - функцією розподілу інтервалу часу між надходженнями заявок, $0 \le t$; $B(t)$ - функцією розподілу часу обслуговування заявок; K - кількістю каналів обслуговування, $K \ge 1$; L - довжиною черги, $0 \le L \le \infty$. Характеристики, які оцінюються за допомогою СМО: $P_в$ – ймовірність втрати заявки; $T_{оч}$ – середня тривалість очікування; L – середня кількість заявок в черзі; M – середня кількість заявок в СМО.

В теорії СМО для оцінки продуктивності застосовується апарат перетворення Лапласа-Стілтьєса (ПЛС). Перетворенням Лапласа-Стілтьєса випадкової величини x функції розподілу $B(x)$ називається функція $\beta(s)$, яка визначається наступним чином

$$\beta(s) = \int_0^\infty e^{-sx} dB(x),$$

де $s \ge 0$ - параметр ПЛС.

Якщо x - неперервна випадкова величина, то ПЛС

$$\beta(s) = \int_0^\infty e^{-sx} B(x)\, d(x),$$

Для оцінки ймовірнісно-часових характеристик використовуються відомі властивості ПЛС:

1. Перетворення Лапласа-Стілтьєса суми випадкових величин дорівнює добутку перетворень Лапласа-Стілтьєса кожної з цих величин, тобто якщо дві незалежні випадкові величини мають ПЛС $\beta_1(s)$ і $\beta_2(s)$ їх функцій розподілів, то ПЛС функції розподілу суми цих величин є $\beta_1(s)\beta_2(s)$.

2. Якщо B_k є k-й момент випадкової величини відносно початку координат, то $B_k = (-1)^k \frac{d^k \beta(s)}{ds^k}|_{s=0}$ тобто моменти випадкової величини визначаються диференціюванням в нулі (при s = 0) відповідне число раз перетворення Лапласа-Стілтьєса функції розподілу цієї величини. Перший центральний момент визначає математичне очікування випадкової величини, $\bar{t} = \beta = \frac{d\beta(s)}{ds}$, а другий центральний момент необхідний для знаходження дисперсії випадкової величини

$$\sigma^2 = B_2 - B_1^2 = \frac{d^2\beta(s)}{ds^2}|_{s=0} - \left[-\frac{d\beta(s)}{ds}|_{s=0}\right]^2.$$

3. $\lim_{s\to 0} \beta(s) = \lim_{t\to\infty} B(t)$.

4. Величина $e^{-sx}B(x_i)$ – є ймовірність складної події, що складається в тому, що випадкова величина не перевищить значення x_i (співмножник $B(x_i)$), а крім того, за час [0, x_i] не відбудеться жодної «катастрофи» (співмножник e^{-sx}). Параметр s розглядається як інтенсивність «катастроф». Інтегрування по всьому діапазону дає $\int_0^\infty e^{-sx} dB(x) = \beta(s)$.

Таким чином, імовірнісний сенс перетворення Лапласа - Стилтьеса при оцінці $t_{пд}$ полягає в тому, що воно визначає ймовірність того, що за час $t_{доп}$ будуть передані всі дані. Розподіл часу передачі даних в Інтернет речей описується в термінах перетворення Лапласа-Стілтьєса. В Інтернеті речей одночасно передаються між різними її елементами K незалежних пуассонівських потоків даних різних класів інтенсивності λ_k, k = 1, ..., K і обслуговуються за довільним законом, тобто кожний сегмент маршруту є СМО типу M|G|1. Для системи M|G|1 відомо рівняння Поллачіка-Хинчина для перетворення Лапласа-Стілтьєса (ПЛС) функції розподілу часуочікування $W(s)$

$$W(s) = \frac{s(1-\rho)}{s - \lambda + \lambda\beta(s)},$$

де ρ - коефіцієнт завантаження каналу; λ - інтенсивність надходження пакетів в канал; $\beta(s)$ - ПЛС часу обробки пакета на СП. Для обчислення $W(s)$ необхідно знайти ПЛС функції розподілу часу обробки пакета $\beta(s)$. Для випадкової величини t - часу обробки пакета з експоненційної функцією розподілу

$$B_k(t) = 1 - e^{-\mu k t},$$

де $\mu_k = C / l_k$ - пропускна здатність СП, в пакетах/с, C - пропускна здатність СП, в бітах/с, l_k - довжина пакета даних k-го класу в бітах, $k = 1, ..., K$.

Перетворення Лапласа-Стілтьєса

$$\beta(s) = \frac{\mu}{(\mu + s)} = \frac{1}{(1 + s \cdot t)}.$$

Для випадкової величини t з рівномірною функцією розподілу

$$B_k(t) = \frac{t - \alpha}{b - \alpha}, \text{ при } \alpha \le t \le b,$$

Перетворення Лапласа-Стілтьєса $\beta(s) = \dfrac{e^{-s\alpha} - e^{-sb}}{s(b - \alpha)}$.

Відповідно до розглянутих властивостей ПЛС тривалість передачі даних k-го класу від вузла i до вузла j визначається як

$$\beta_k(t) = \prod_{d=1}^{N} W_d(t)\,\beta_d(t),$$

де $\beta_d(t)$ - ПЛС функції розподілу часу обробки пакета в d-му каналі маршруту; $W_d(t)$ - ПЛС функції розподілу часу очікування $W(s)$ в d-му каналі маршруту. Таким чином, даний розподіл буде функцією від векторів інтенсивностей надходження і обслуговування пакетів $\beta_k(t) = f(\lambda_k, \mu_k)$.

Модель дає розподіл часу перебування даних k-го класу в мережі Інтернет речей, при заданому векторі ймовірностей виникнення помилок на елементах маршруту для всіх пар взаємодіючих СП. ПЛС дозволяє при обмеженнях $t_{доп}$ на час $t_{пд}$ деякого маршруту визначити можливе навантаження на цей маршрут і здійснити вибір відповідного алгоритму самоорганізації мережі.

Перелік посилань

1. Кутузов, О. И. Инфокоммуникационные сети. Моделирование и оценка вероятностно-временных характеристик [Текст] : монография / О. И. Кутузов, Т. М. Татарникова - СПб. : ГУАП, 2015. - 381 с.

2. Риз, Дж. Облачные вычисления. / Дж. Риз: пер. с англ. —СПб.: БХВ-Петербург, 2011. —288 с.: ил.

3. Тархов, Д. А. Нейросетевые модели и алгоритмы : справочник / Д. А. Тархов. — Москва : Радио- техника, 2014. — 349 с. : ил.

Модифікована архітектура системи автоматизованого тестування

Коваленко О.О.

Науковий керівник – к.т.н., доц. Джулій В.М.

Хмельницький національний університет

Типова архітектура систем автоматизованого тестування вимагає значної участі з боку інженерів по тестування і адміністраторів. Функція підтримки цілком і повністю покладається на людину. Даний підхід має ряд недоліків:

1. Для якісного аналізу стану тестів необхідно аналізувати великий обсяг даних (мільйони тестових завдань і десятки мегабайт файлів звітів). Людина погано пристосована для подібної роботи.

2. Контроль за станом системи потрібно проводити якомога частіше, щоб максимально швидко виявляти проблему і виправляти її до того, як вона вплине на велику кількість тестових завдань. З урахуванням великого обсягу даних і цілодобового режиму тестування, необхідно організовувати позмінну роботу великої кількості інженерів.

3. Рутінний характер роботи важкий для людини і демотивує інженерів, які беруть участь в підтримці процесу тестування.

4. В характері роботи закладено протиріччя:

- виявлення проблем не вимагає високої кваліфікації;
- виправлення виявлених проблем, вимагає широких знань і високої кваліфікації.

У зв'язку з цим персонал, який здійснює підтримку, розбитий на дві команди:

- виявлення проблем і спроба вирішення по шаблонах покладається на команду безпосередньої підтримки, що складається з менш кваліфікованих фахівців і працюють позмінно;

- рішення проблем, не вирішених на першому етапі, вирішують більш кваліфіковані фахівці.

В результаті потрібна велика команда різних фахівців для підтримки системи. Зважаючи на характер роботи існує велика текучка і як наслідок, високі витрати на навчання. При цьому характер робіт по підтримці процесу тестування ідеально підходить для автоматизації: обробка великих обсягів даних, велика кількість повторень, безперервна цілодобова робота, наявність готових сценаріїв обробки помилок.

В рамках даної роботи пропонується впровадити інтелектуальну систему, що виконує функції команди безпосередньої підтримки, для підвищення ефективності роботи САТПК. В результаті відбудеться перехід до використання системи, зазначеної на рис. 1, в якій інтелектуальний модуль автоматизує ряд рутинних операцій, які раніше вимагали участі персоналу.

Модифікована система автоматизованого тестування зображена на рис. 1.

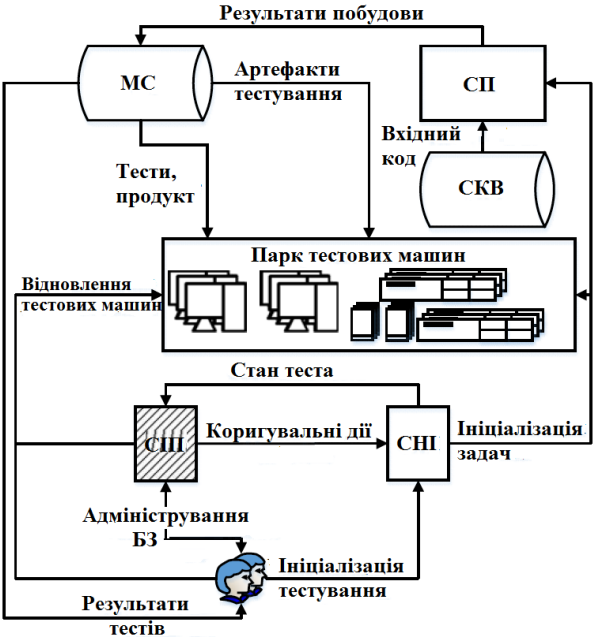

Рисунок 1 - Модифікована архітектура систем автоматизованого тестування

Модифікована система складається з наступних частин:

1. Парку тестових машин, під яким слід розуміти всі пристрої, використовувані для тестування програмного забезпечення. Це можуть бути серверні платформи, настільні комп'ютери, мобільні системи, планшети, смартфони. Крім того, можуть використовуватися системи віртуалізації. Наприклад, Android емулятори або віртуальні машини VMWare. Парк тестових машин може перебувати в одному місці, так і бути географічно розподіленим.

2. Системи розподілу завдань, в якості якої використовується одна з систем безперервної інтеграції (СБІ). Крім розподілу тестових завдань вона дозволяє централізовано адмініструвати тестові машини. Дані підсистеми побудовані на клієнт серверній архітектурі. При цьому є один сервер і на кожній тестовій машині працює клієнтський додаток, який обробляє запити від сервера і здійснює запуск додатків, відповідних завданню, отриманої від сервера, контролює їх виконання і збирає інформацію про процес виконання на конкретному тестовому пристрої.

3. Системи побудови (СП), що складається з декількох серверів, для побудови тестованого програмного забезпечення і тестів для всіх цільових платформ.

4. Системи контролю версій вихідного коду (СКВ), яка містить вхідний код програмних засобів, що використовуються для тестування, в тому числі:

- тести, розроблені для тестування продукту;

- система емуляції оточення - дозволяє підготувати тестову машину для виконання тестів. Зазвичай включає в себе установку або емуляцію установки: програми, від яких залежить тестоване програмне забезпечення; програми, від яких залежать тести; саме тестоване програмне забезпечення; програми, що забезпечують тестування.

5. Мережеве сховище (МС), яке служить для зберігання програм, необхідних для забезпечення тестування, скомпільовані версії тестів та тестового програмного забезпечення, а також артефактів тестування.

6. Тестових інженерів і системних адміністраторів, які забезпечують роботу системи.

7. Модуля системи інтелектуальної підтримки (СІП), який бере на себе частину функцій, раніше виконуваних персоналом.

Як вже говорилося вище, існуюча архітектура автоматизувала всі однозначно визначені послідовності дій. Решта дії вимагали аналізу ситуації і виконувалися інженерами по тестуванню.

У процесі модифікації продукційної моделі до предметної області автоматизованого тестування була проаналізована статистика виконання тестів в системі автоматизованого тестування. В результаті були виділені: множина об'єктів в предметної області (дана множина відображається в базі знань через стан об'єктів) – збірка продукту, збірка тестів, тестові пристрої, тестова задача і т. д .; множина можливих станів об'єктів предметної області відображається на множині умов (*A*) - для тестового завдання це: не запущений, в черзі на виконання, виконується, завершив виконання; множина операцій, що призводять об'єкти в певний стан, відображається на множині допустимих дій - для тестової завдання це: перезапуск, перезапуск на іншому тестовому пристрої, відправка повідомлення розробнику тесту і т.д. Поряд з перевагами, у продукційних систем є ряд недоліків. До недоліків продукційної моделі можна віднести наявність конфліктів двох типів:

1. Формальні конфлікти, які виникають, коли в процесі життєвого циклу в базі знань виникають кілька правил з однаковими лівими частинами, але різними правими.

2. Конфлікти, що виникають при інтеграції проектів високого рівня складності, коли існує кілька різних підходів до вирішення однієї і тієї ж задачі.

Обидва типи конфліктів вносять неоднозначність в процес прийняття рішення. У разі конфліктів першого типу можна ввести в розгляд додатковий параметр об'єкта, тим самим диференціюючи два стани і правила. Але в реальних системах не завжди можна отримати достатню деталізацію стану об'єкта. В результаті одні й ті ж «симптоми» можуть описувати різні «реальні» стани об'єкта і вимагати різні дії. Тому неможливо гарантувати відсутність формальних конфліктів в базі знань. Незважаючи на те що виникнення конфліктів альтернативних рішень малоймовірно, тому що в даному випадку предметна область досить обмежена, при цьому за рахунок розбиття всієї предметної області на фрагменти, відповідні типові завдання, досягається зменшення ймовірності виникнення подібних конфліктів, але виключити їх виникнення не можна.

Таким чином, наявність конфліктів обох видів є невід'ємним властивістю продукційної моделі баз знань. А так як конфлікти можуть виникати протягом усього життєвого циклу виробничої системи, що обумовлено тим, що очікується поява нових знань про предметну область в процесі роботи СІП, то потрібно формалізувати й автоматизувати засоби вирішення подібних ситуацій.

Нова архітектура, за рахунок автоматизації прийняття рішень, дозволяє автоматизувати, які раніше не були автоматизовані, операції і знизити навантаження на персонал, що дозволить інженерам з тестування сфокусуватися на покращенні якості тестування за рахунок поліпшення самих тестів і збільшення тестового покриття.

Перелік посилань

1. Проскурин В. Г. Защита программ и данных: учебное пособие / В. Г. Проскурин, С. В. Крутов, И. В. Мацкевич. – М.: Академия, 2011. – 198 с.

2. Советов, Б. Я. Моделирование систем : учебник для бакалавров / Б. Я. Советов, С. А. Яковлев. –7-е изд. – М. : Издательство Юрайт, 2015. – 343 с.

3.Тархов, Д. А. Нейросетевые модели и алгоритмы : справочник / Д. А. Тархов. – М. : Радио- техника, 2014. – 349 с.

Вдосконалення організації інформації в самоорганізованих мережах

Ковпа Д.М.

Науковий керівник – к.т.н.,доц. Хмельницький Ю.В.

Хмельницький національний університет

Проведений аналіз методів та моделей управління програмно - керованими мережами показав, що проблема забезпечення якості обслуговування займає досить важливе місце у наукових працях закордонних і вітчизняних дослідників. Значна увага науковців приділяється до адаптивності програмно - керованої мережі в умовах обслуговування мультисервісного потоку даних, яка реалізується завдяки можливостям протоколу Open-Flow. Це дає можливість здійснювати оперативний моніторинг функціональних параметрів пристроїв передачі даних у мережах і можливість їх динамічно їх програмувати. В рамках аналізу складових елементів системи управління SDN архітектури, що забезпечують якість

обслуговування потоків у програмно - керованими мережах є деякі недоліки. Це, незважаючи на вищий рівень оперативності обчислення маршрутів та аналіз топології мережі, сама маршрутизація здійснюється за допомогою традиційних алгоритмів, що використовуються існуючими протоколами - OSPF та EIGRP. Різнорідність самої апаратної реалізації пристроїв передачі даних мережі призводить до того, що різні види комутаторів можуть не підтримувати деякі функції чи підтримувати їх із обмеженою продуктивністю. У процесі роботи мережі це може суттєво вплинути на пропускну здатність окремих потоків передачі даних чи цілих доменів мережі. Сама маршрутизація потоків передачі здійснюється за критерієм якості обслуговування чи за критерієм рівномірного завантаження мережевих ресурсів мережі.

Підвищення якості обслуговування здійснюється із врахуванням класифікації потоку передачі згідно ITU-T, що досить суттєво обмежує можливість управління потоком даних. Більшість моделей не враховують характеристик мультисервісного потоку даних, що призводить до погіршення якості обслуговування та підвищення ймовірності блокування каналів передачі мережі. Ще одна категорія методів базується на потокових аналітичних моделях для оптимізації мережі. Ці методи використовують моделі балансування навантаження потоків передачі даних, які не враховують чутливості потоку даних до перемішування порядку пакетів, погіршення затримки та тимчасового розриву з'єднання.

Відсутність можливості здійснювати диференційоване управління для окремих потоків даних окремих клієнтів мережі та врахувати їхні вимоги щодо якості, призводить до низької ефективності каналу маршрутизації, неоптимального розподілу навантаження мережі, погіршення якості обслуговування високо пріоритетних потоків даних. Самі ж засоби контролю за процесом передачі окремих потоків передачі даних відсутні, внаслідок чого система управління мережею не має змогу визначити погіршення якості обслуговування для цих потоків даних, а тому не зможе гарантувати рівень якості, узгоджений у сервісі. Враховуючи проведений аналіз, можна дійти до висновку, що існуючі моделі управління інформаційним потоком у програмно - конфігуруємо мережах не завжди враховують вимоги окремого клієнта, а диференціюють потоки лише за класами потоку інформації. Не використовують актуальні параметри управління та обслуговування як окремих каналів, так й індивідуальних потоків окремого клієнта мережі та не завжди можуть справитися із перевантаження елементів телекомунікаційної

мережі, вузлів чи каналів. Проводить маршрутизацію потоків даних мережі, не диференціюючи їх за чутливістю до перемішування порядку пакетів та розриву з'єднання [1].

Розглянемо архітектуру SDN (рис. 1), що складається із рівня програми, рівня мережевого устаткування та рівня даних. Мережеві комутатори стають простими пристроями пере адресації, а логіка управління в архітектурі SDN реалізована в логічно централізованому контролері.

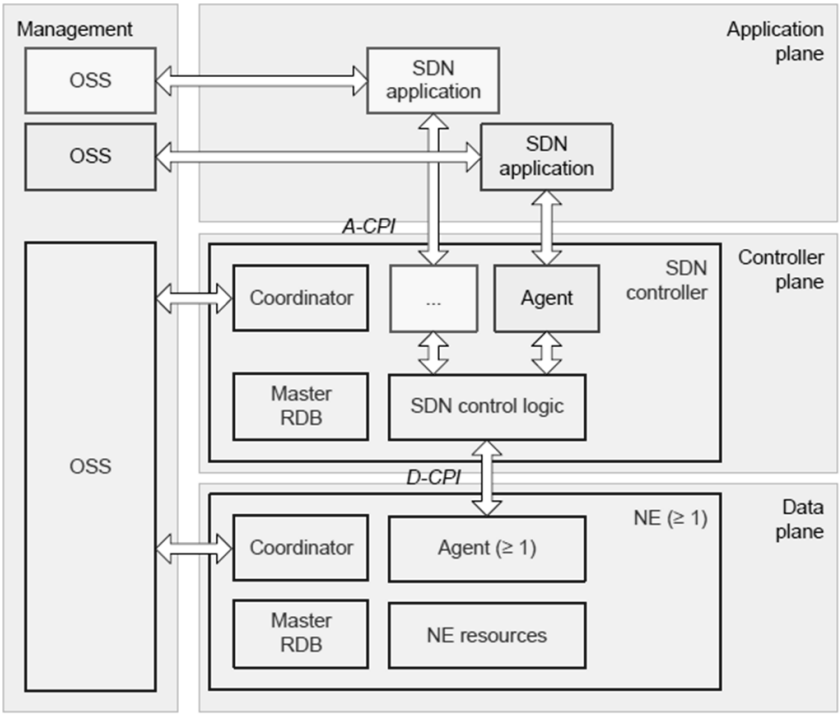

Рисунок 1 – Класична модель архітектура SDN мережі

SDN контролер управляє ресурсами даних через D-CPI інтерфейс. A-CPI інтерфейс використовується для реалізації зв'язку між додатками та контролером тому функція управління відбувається через інтерфейс управління. Така конфігурація може підтримувати різні мережеві додатки. На відміну від традиційних мереж у мережах SDN маршрутизатор виконує лише функції передачі. Відмінність передачі пакетів у традиційних мережах та мережах SDN представлено на рис. 2 [2].

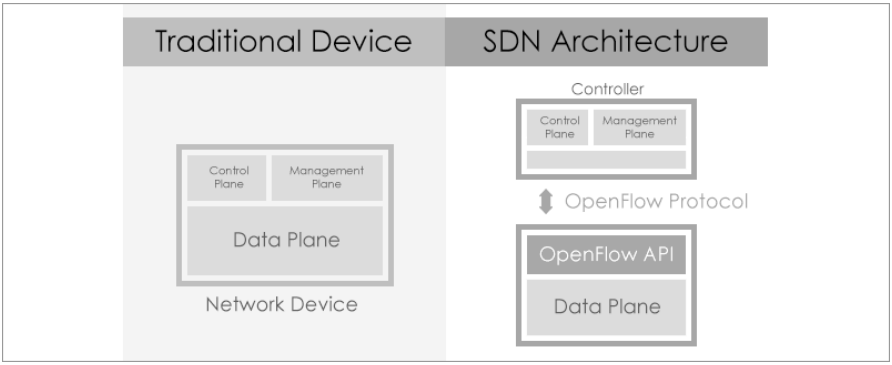

Рисунок 2 – Порівняння традиційної та SDN архітектури

При аналізі моделей та алгоритмів функціонування системи управління SDN архітектури розглянемо співвідношення до її традиційних мереж. Усі сучасні інформаційні технології висувають досить великі вимоги до гнучкості та масштабованості комп'ютерних мереж і очікується, що SDN мережі допоможуть вирішити цілий ряд наявних проблем, сприятимуть створенню автоматизованих, програмованих, гнучких та економічних мережевих інфраструктур, однак ця стратегія у різних провідних виробників помітно розрізняється. За оцінками GARTNER[1], на мережеву інфраструктуру припадає приблизно 17% IT - бюджету. При цьому вона далеко не завжди може адаптуватися до змін потреб бізнесу. Тому нові тенденції - віртуалізація, хмарні обчислення, мобільність користувачів, зростання обсягів передачі - змінюють вимоги до мережевих інфраструктур. Будуть виникати питання чи зможуть сьогодні мережеві продукти забезпечити підтримку майбутніх додатків і сервісів, якою мірою розвиток мережі буде прив'язаний до продуктів обраного виробника тощо. Пріоритетність рішень, архітектура традиційного мережевого обладнання робить цю прив'язку дуже міцною. Деякі виробники навіть характеризують поточну ситуацію у мережевий галузі як революційну.

Ряд експертів [2] в якості рецепту усунення розкритих в мережах проблем називають перехід до архітектури програмно-керованих мереж (Software-Defined-Networking, SDN). Архітектура SDN обіцяє істотно послабити залежність від замовників технологій конкретного виробника. В архітектурі SDN вся логіка управління мережевими пристроями виноситься в так звану «площину управління», яка реалізується програмним чином.

Конструктивно контролери в архітектурі SDN можуть будуватися на базі фізичних або віртуальних вузлів. В архітектурі SDN управління мережевими пристроями, як правило, здійснюється по протоколу Open-Flow. Головна ідея архітектури SDN - відділення функцій передачі даних від функцій управління. У традиційних комутаторах та маршрутизаторах ці процеси зв'язані. У архітектурі SDN мережа, що складається із безлічі пристроїв різних виробників, постає для застосування як один логічний комутатор. Архітектура SDN дозволяє адміністраторам програмувати мережу як єдине ціле, а не займатися окремими комутаторами, які можуть просто виконувати інструкції контролера. Реалізація такої концепції значно спрощує експлуатацію та функціонування мережі, її конфігурацію. Комутатори можуть бути простими та дешевими. Характеристики мережі можна оперативно змінювати у режимі реального часу, скорочуються терміни впровадження нових додатків та сервісів. Програмні інтерфейси (API), контролери дозволяють розробникам створювати додатки для управління такою мережею. Ці програми можуть виконувати найрізноманітніші функції, причому для цього не потрібно знати особливості роботи конкретних мережевих пристроїв. З точки зору виробників, такий підхід не повинен викликати ентузіазму у розробників мережевого устаткування, які багато років удосконалювали унікальні функції своїх комутаторів та маршрутизаторів. Можливість використання простих та дешевих комутаторів, створення додатків сторонніми розробниками за рахунок відкритих API підриває бізнес таких компаній, позбавляє їх джерела додаткової вартості. Проте великі замовники, включаючи провідних операторів зв'язку та провайдерів, вже перейнялися ідеями архітектури SDN, а виробники мікросхем комутаторів оголосили про підтримку Open-Flow, тому поставники не можуть залишатися осторонь.

Реалізація необхідної функціональності є індивідуальною та змінюється від одного виробника до іншого та може привести до змін продуктивності у реальній мережі. Проблеми виникають уже тоді, коли програмно - керована мережа налаштована та окремі комутатори починають створювати вузькі місця, погіршуючи продуктивність та якість обслуговування мережі. Такі недоліки можна врахувати на стадії проектування програмно – керованої мережі та розробки програмного забезпечення для контролера. Тому необхідно, щоб розробники додатків для програмно – керованих мереж змогли отримати інформацію про обмеження продуктивності конкретного комутатора із метою забезпечення стабільної та надійної роботи мережі. Завдяки такій системі розробники зможуть отримати

необхідні дані та можливість охарактеризувати продуктивність конкретного комутатора. Комутатор, залежно від апаратної реалізації, може надавати доступ до окремих параметрів. Специфікація Open-Flow містить загальний алгоритм роботи та список параметрів доступних для моніторингу:

- статистика потоків передачі - тривалість існування, пріоритет, кількість оброблених пакетів, байт;
- статистика таблиць потоків даних - кількість правил, кількість оброблених пакетів, байт;
- статистика портів передачі - кількість переданих чи відкинутих пакетів, байт, помилок передачі та колізій);
- статистика черг передачі - довжина черги, кількість переданих пакетів, байт, кількість відкинутих пакетів через переповнення;
- інші специфікації - STP, збірка сегментів IP пакетів тощо.

Основним компонентом розглянутої системи моніторингу є додаток моніторингу, який встановлюється на фізичному сервері. Цей додаток виконує ключову роль збору, форматування та представлення інформації у форматі, зручному для користувача і для подальшої обробки цієї інформації іншими додатками. База даних зберігає інформацію, необхідну для роботи додатка, наприклад, для налаштування комутаторів, моделі та технічні особливостей того чи іншого комутатора. Ці дані заносяться в базу даних адміністратор програмно – керованої мережі. У базі даних зберігаються зібрані дані моніторингу та оброблені статистичні характеристики поведінки мережі в певні періоди.

Враховуючи проведений аналіз, можна дійти до висновку, що існуючі моделі управління інформаційним потоком у програмно - конфігуруємо мережах не завжди враховують вимоги окремого клієнта, а диференціюють потоки лише за класами потоку інформації. Розглянута схема розробки мереж SDN показує, що різні організації при створені мереж SDN, ставлять за мету формування такої архітектури мережі та устаткування, що припускає відділення площини управління від площини передачі та докладають значних зусиль до подолання виникаючих проблем, пов’язаних із складнощами міграції від традиційних мереж до архітектури SDN. Це в подальшому надасть можливість користувачам отримувати послуги із необхідною якістю, надійністю, достовірністю та прогнозувати ризики і стабільність якості функціонування системи управління такої архітектури.

Перелік посилань

1. Климаш М.М. Система моніторингу пакетної затримки в

програмно-конфігурованих телекомунікаційних мережах / М.М.Климаш, М.О.Селюченко, О.М.Панченко // X Міжнародна науково -технічна конференція «Проблеми телекомунікацій» ПТ-2016: Збірник матеріалів конференції (м. Київ, 19-22 квітня 2016 р.). - К.: НТТУ «КПІ», 2016. - С.345-347.

2. Стєклов В.І. Проектування телекомунікаційних мереж. Підручник для студ. вищ. навч. закл. за напрямком "Телекомунікації"/ В.І. Стєклов, Л.Н.Беркман //-К.: Техніка, 2002.-792 с.

Підвищення тестопридатності цифрових об'єктів діагностування на основі послідовної структурної декомпозиції

Козенюк О.М.
Науковий керівник – к.т.н.,доц. Чешун В.М.
Хмельницький національний університет

Через неможливість комплексного вирішення всіх задач технічної діагностики, що виникають при організації експериментів з тестових перевірок будь-якого сучасного цифрового об'єкта діагностування (ЦОД), постановку експерименту розбивають на підзадачі, кожна з яких орієнтована на вирішення певних завдань і має обмежений спектр застосування. Для вирішення зазначених завдань також використовуються спеціалізовані методи, орієнтовані саме на даний клас задач і, відповідно, з обмеженою сферою застосування.

Таким чином, при постановці будь-якого завдання технічної діагностики сучасних ЦОД потрібно обмежити сферу актуальності задач даного завдання і уточнити вимоги щодо очікуваних результатів. Аналогічно, обмежити сферу актуальності задач і уточнити вимоги щодо очікуваних результатів необхідно при формуванні опису методу структурної декомпозиції цифрових об'єктів діагностування.

Першочергово визначимо загальні положення методу:

- метод структурної декомпозиції ЦОД орієнтований на застосування на завершальному етапі проектування схем функціонально-завершених цифрових вузлів або пристроїв;
- метод призначений для забезпечення рівня тестопридатності ЦОД відповідно висунутим вимогам щодо глибини їх діагностування;
- метод орієнтований на декомпозицію і аналіз структурної організації ЦОД з метою визначення мінімально-достатнього набору контрольних точок для реалізації діагностичних перевірок із заданою точністю, а також шляхів і способів транспортування діагностичних даних (сигналів) в структурі об'єкта діагностування;
- за результатами застосування методу можуть бути сформульовані

вимоги щодо забезпечення доступу до мінімально-достатнього набору контрольних точок для реалізації діагностичних перевірок, а також рекомендації щодо структурної реорганізації схеми об'єкта діагностування для збільшення його тестопридатності;

- вихідними даними для реалізації методу є інформація про структурну організацію ЦОД як сукупності інтегральних компонентів $w_h \in W$ з фіксованим набором контактів і зв'язків між ними, а також діагностичні тести, необхідні для відокремленої перевірки кожного компонента $w_h \in W$ із заданою точністю.

Деталізуємо наведені базові вимоги.

Першочергово, метод структурної декомпозиції ЦОД орієнтований на застосування на завершальному етапі проектування схем функціонально-завершених цифрових вузлів або пристроїв, які при застосуванні методу розглядаються в якості ЦОД, і призначений для забезпечення рівня тестопридатності створення цифрових вузлів (пристроїв) відповідного висунутим вимогам щодо глибини діагностування.

Приведення структури ЦОД до умов тестопридатності проводиться з позицій забезпечення необхідної глибини пошуку несправностей, тому першочергово слід ввести обмеження щодо глибини діагностування, на яку буде зорієнтовано метод.

Найпростішим варіантом перевірки ЦОД є їх перевірка за принципом «справний-несправний», але така перевірка, поряд з максимальною простотою реалізації, є малоінформативною і не може бути прийнята за обмеження глибини пошуку несправностей для методу структурної декомпозиції ЦОД, оскільки, фактично, жодним чином не вказує на місце виникнення і тип несправності в складі ЦОД.

В розгляді будь-яких ЦОД при їх тестовому діагностуванні важливою є структурна будова досліджуваного об'єкта, яка, фактично, може бути представлена у вигляді множини окремих блоків (вузлів, схем або інших структурних одиниць тощо), на пошук несправностей орієнтовані процедури діагностування.

Не має потреби і занадто завищувати вимоги щодо глибини діагностування цифрових об'єктів, зокрема, не доцільно заглиблюватись при пошуку місця виникнення дефектів і зумовлених ними несправностей у внутрішню будову інтегральних компонентів з двох основних причин:

- пошук дефектів у внутрішній структурі інтегральних компонентів, що є складовими частинами ЦОД, для більшості задач технічної діагностики не є необхідним, оскільки ремонт або заміна складових частин в середині інтегрального компонента не є можливими;

- пошук дефектів із заглибленням у внутрішні вузли інтегральних компонентів призводить до невиправданого багатократного збільшення складності застосовуваних тестів і до надмірного збільшення вартості

синтезу таких тестів.

На сьогоднішній день, як правило, вважається, що глибина діагностування може вважатися досягнутою, якщо пошук несправностей ЦОД проводиться до рівня окремих інтегральних компонентів (мікросхем), при чому самі зазначені інтегральні компоненти проходять повну діагностичну перевірку для встановлення діагнозу за принципом «справний-несправний».

Отже, при розгляді методу структурної декомпозиції ЦОД введемо за необхідну вимогу щодо глибини діагностування ЦОД забезпечення можливості їх перевірки із локалізацією місця виникнення несправностей з точністю до інтегрального компонента та класифікацією стану перевірених компонентів за принципом «справний-несправний». Ознакою справності ЦОД в цілому будемо вважати проходження всіма компонентами, наявними в їх структурі, всіх передбачених планом експерименту діагностичних перевірок з діагнозом «справний».

За висновками, що були зроблені при обґрунтуванні математичної моделі методу структурної декомпозиції ЦОД, такий підхід є виправданим з цілого ряду причин:

- більш глибока деталізація структурної будови цифрового пристрою для більшості задач технічної діагностики не є необхідною, оскільки ремонт або заміна складових інтегрального компонента не є можливими;
- деталізація структурної будови цифрового пристрою з точністю до інтегрального компонента дозволяє перейти до спрощених моделей його представлення із обґрунтованим збереженням можливостей проведення діагностичних випробувань відповідно до заданих вимог із застосуванням методу структурної декомпозиції ЦОД;
- розгляд ЦОД як сукупності інтегральних компонентів дозволяє зводити задачу діагностування цифрового об’єкта з рівня «чорного ящика» до рівня взаємопов’язаної сукупності інтегральних компонентів з певними відомими характеристиками;
- для інтегральних компонентів в діагностичних бібліотеках з більшою вірогідністю можуть бути наявні дані щодо типів можливих несправностей та статистичних ймовірностей їх виникнення, ніж для цілого пристрою (особливо на етапі проектування схем зазначених пристроїв, де і планується застосовувати метод структурної декомпозиції ЦОД). За відсутності безпосередніх даних про несправності наявного інтегрального компонента, відповідні дані можуть бути прогнозовані експертами або експертними системами на підставі накопиченої діагностичної інформації про аналогічні або споріднені компоненти;
- для інтегральних компонентів фірмою-виробником можуть надаватися перевірочні тести або технічна документація, достатня для створення таких тестів, що спрощує перехід від покомпонентних тестів до

структурного тестування, особливо із застосуванням пропонованого методу структурної декомпозиції ЦОД. За відсутності подібних тестів від виробника, їх, знову ж таки, можна створити з діагностичних бібліотек на підставі накопиченої діагностичної інформації про тестування аналогічних або споріднених компонентів, що є значно простішим, ніж розробка тестів для ЦОД одразу в цілому.

Виходячи з вже визначених положень, що метод структурної декомпозиції ЦОД призначений для забезпечення рівня тестопридатності створюваних цифрових вузлів (пристроїв) відповідного висунутим вимогам щодо глибини діагностування, а також з того, що в даній кваліфікаційній роботі глибина діагностування обмежується перевіркою справності інтегральних компонентів, які входять до складу ЦОД, з локалізацією місця виникнення несправності з точністю до компонента, для реалізації методу обов'язковою є наявність опису множини структурних складових ЦОД (з точністю до компонента).

Згідно з математичною моделлю методу, загальне структурне представлення ЦОД буде відображається множиною структурних складових ЦОД W:$\{w_1,w_2,\ldots,w_h,\ldots,w_n\}$.

Таким чином, якщо будь-який елемент $w_h \in W$ є представленням неподільного інтегрального компонента в складі досліджуваного ЦОД, то загальна кількість елементів цієї множини $n=|W|$ має відповідати числу інтегральних компонентів (мікросхем), з яких зібрано схему об'єкта діагностування.

При реалізації методу можна враховувати, що структурні складові ЦОД діляться два структурні складові, що являють собою схеми комбінаційного типу, і структурні складові, що являють собою схеми з пам'яттю або, в альтернативній класифікації, схеми послідовнісного типу.

З урахуванням поділу множини структурних складових ЦОД на схеми комбінаційного типу і на схеми з пам'яттю, в складі множини можна відокремити дві різнотипних підмножини: $W_к$:$\{w_{к\,1},w_{к\,2},\ldots,w_{к\,i},\ldots,w_{к\,m}\}$ - множина структурних складових ЦОД комбінаційного типу; W_n:$\{w_{n\,1},w_{n\,2},\ldots,w_{n\,i},\ldots,w_{n\,n\text{-}m}\}$ - множина структурних складових ЦОД послідовнісного типу (структурних складових з пам'яттю). Нажаль, переважна більшість інтегральних компонентів сучасних ЦОД є структурними складовими з пам'яттю, тому акцентувати увагу на поділі множини структурних складових ЦОД W на підмножину структурних складових комбінаційного типу $W_к$ і підмножину структурних складових послідовнісного типу W_n в розгляді сучасних ЦОД не доцільно.

В окремих випадках до числа структурних складових ЦОД відносять також з'єднувальні елементи (лінії зв'язку, роз'єми, слоти тощо), але зазначені елементи не виконують функцій перетворення даних і відіграють роль контрольних точок об'єкта діагностування, які задіюються та

перевіряються при перевірці інших структурних складових ЦОД $w_h \in W$, що зумовлює недоцільність загромадження математичної моделі методу структурної декомпозиції ЦОД елементами окремого опису з'єднувальних елементів.

Оскільки метод структурної декомпозиції ЦОД призначений для забезпечення рівня тестопридатності створюваних цифрових вузлів (пристроїв), відповідного висунутим вимогам щодо глибини діагностування, а в даній кваліфікаційній роботі глибина діагностування обмежується перевіркою загальної справності інтегральних компонентів, що входять до складу ЦОД, з локалізацією місця виникнення несправності з точністю до компонента, і при цьому сукупність компонентів представлена в реалізації методу множиною структурних складових ЦОД W, то діагностування подібного об'єкта зводиться до перевірки справності всіх структурних складових ЦОД $w_h \in W$. Для перевірки справності всіх структурних складових $w_h \in W$ ЦОД виникає потреба у застосуванні тестів, здатних виконати перевірку наявних компонентів $w_h \in W$ за принципом «справний-несправний». В ідеальному варіанті це має бути комплексний тест, адаптований на пошук несправних компонентів в структурі саме досліджуваного типу ЦОД.

Оскільки метод структурної декомпозиції ЦОД орієнтований на застосування на завершальному етапі проектування схем функціонально-завершених цифрових вузлів або пристроїв, які при застосуванні методу розглядаються в якості ЦОД, то говорити про наявність на цьому етапі комплексного тесту з адаптацією на пошук несправних компонентів в структурі досліджуваного новоствореного типу ЦОД апріорі неможливо. Більш того, існує велика імовірність того, що такий комплексний тест без адаптації самої схеми ЦОД для забезпечення тестопридатності до проведення діагностичних випробувань із заданою глибиною діагностування створити неможливо, що і стало підставою для розробки методу структурної декомпозиції ЦОД.

В той же час, існує велика імовірність наявності типових тестів, орієнтованих на перевірку справності різних видів інтегральних компонентів $w_h \in W$ за умов їх відокремленого діагностування, тобто, для покомпонентної перевірки складових ЦОД.

Тести, орієнтовані на перевірку справності інтегральних компонентів $w_h \in W$, можуть бути отримані різними способами:

- перевірочні тести для стандартних інтегральних компонентів можуть надаватися фірмою-виробником;

- перевірочні тести для стандартних інтегральних компонентів можуть бути отримані з діагностичних бібліотек;

- за відсутності перевірочних тестів, їх, знову ж таки, можна створити на підставі технічної документації від виробника, або на підставі накопиченої діагностичної інформації про тестування аналогічних або споріднених

компонентів з діагностичних бібліотек, що є значно простішим, ніж розробка тестів для ЦОД одразу в цілому.

Оскільки завдання синтезу перевірочних тестів, незалежно від того, призначаються вони для тестування інтегральних компонентів, які входять до складу досліджуваних ЦОД, або ж для загального тестування досліджуваних ЦОД, не входить до числа задач методу структурної декомпозиції ЦОД, при розгляді методу будемо спиратися на положення, що тести, орієнтовані на перевірку справності інтегральних компонентів $w_h \in W$, є в наявності.

Згідно з математичною моделлю методу структурної декомпозиції ЦОД, тести, призначені для перевірки справності інтегральних компонентів $w_h \in W$ позначаємо $t[w_h]$, сукупність тестів відповідного призначення в математичній моделі ідентифікується як $T[w_h]:\{t[w_h]_1, t[w_h]_2, \ldots, t[w_h]_j, \ldots, t[w_h]_d\}$.

Для кожного тесту $t[w_h]_j$ має бути співставлено у відповідність певний набір сигналів відповідних реакцій ЦОД $r[w_h]_j$, який є очікуваним за умов правильного спрацьовування ЦОД на тест $t[w_h]_j$. З останнього ствердження можна зробити висновок, що кожній множині тестів $T[w_h]$, призначених для перевірки технічного стану компонента $w_h \in W$, має бути співставлена множина відповідних реакцій зазначеного компонента ЦОД на тести $t[w_h]_j \in T[w_h]$ за умови його перебування в справному стані $R[w_h]:\{r[w_h]_1, r[w_h]_2, \ldots, r[w_h]_j, \ldots, r[w_h]_d\}$.

Сукупність множини $T[w_h]:\{t[w_h]_1, t[w_h]_2, \ldots, t[w_h]_j, \ldots, t[w_h]_d\}$ тестів $t[w_h]_j \in T[w_h]$, призначених для перевірки справності інтегрального компонента $w_h \in W$, і множини $R[w_h]:\{r[w_h]_1, r[w_h]_2, \ldots, r[w_h]_j, \ldots, r[w_h]_d\}$ відповідних реакцій $r[w_h]_j \in R[w_h]$ зазначеного інтегрального компонента $w_h \in W$ на тести $t[w_h]_j \in T[w_h]$ утворюють комплект тестових діагностичних даних, які є достатніми для перевірки справності інтегрального компонента $w_h \in W$ за умов його відокремленого діагностування, тобто, для перевірки інтегрального компонента $w_h \in W$ засобами покомпонентного діагностування за відсутності його зв'язку і взаємовпливів з іншими елементами.

Оскільки вилучення всіх складових частин (інтегральних компонентів $w_h \in W$) ЦОД не передбачено їх конструктивною організацією, забезпечити ідеальні умови для перевірки інтегральних компонентів $w_h \in W$ засобами покомпонентного діагностування за відсутності їх зв'язку і взаємовпливів з іншими елементами неможливо. Також неможливо забезпечити доступ до всіх потрібних для діагностування контактів інтегральних компонентів $w_h \in W$ в структурі зібраного в єдиний модуль ЦОД, тому виникає задача забезпечення можливості діагностування інтегральних компонентів $w_h \in W$ в структурі зібраного в єдиний модуль ЦОД із застосуванням штатних контрольних точок, які передбачені схемою об'єкта діагностування (як

правило, це контакти крайових з'єднувачів і засобів з'єднання з іншими модулями системи).

При такому підході основною задачею постановки діагностичного експерименту є визначення такої множини створюваних для ідентифікації технічного стану досліджуваного ЦОД тестів T, щоб на вхідних контрольних точках будь-якого компонента $w_h \in W$ об'єкта діагностування можна було сформувати будь-який з перевірочних тестів $t[w_h]_j \in T[w_h]$ з подальшим транспортуванням отримуваних на вихідних контактах компонента $w_h \in W$ відповідних реакцій $r[w_h]_j \in R[w_h]$ на вихідні контрольні точки об'єкта діагностування $q_{вих.\,i} \in Q_{вих}$ у вигляді певних характерних значень вихідних реакцій ЦОД $r[w_h]_j$.

Сучасні цифрові пристрої як об'єкти діагностування не відрізняються адаптованістю для проведення тестових перевірок і для транспортування використовуваних при цьому сигналів діагностичних даних. Через це зачасту виникають ситуації, коли діагностичні сигнали або взагалі не можуть бути транспортовані між двома віддаленими контактами в структурі ЦОД (доступною у випробуваннях контрольною точкою об'єкта діагностування і контактом інтегрального компонента в його складі), або ж це потребує таких складних і тривалих операцій перетворення і транспортування сигналів, що діагностичні перевірки стають не рентабельними за часовими або іншими критеріями собівартості. Відповідно, побудова комплексного тесту (визначення множини утворюючих його тестів T), за потреби транспортування діагностичних сигналів між контактами інтегральних компонентів $w_h \in W$ і штатними контрольними точками ЦОД через інші компоненти $w_h \in W$, також може бути недосяжною або нерентабельною.

Не заглиблюючись в задачі синтезу тестів, які не є предметом досліджень методу структурної декомпозиції ЦОД, уточнимо задачу методу з урахуванням останніх висновків.

Оскільки транспортування необхідних діагностичних сигналів між контактами досліджуваних інтегральних компонентів $w_h \in W$ і штатними контрольними точками ЦОД через інші компоненти $w_h \in W$ може бути неможливим або нерентабельним, множину штатних контрольних точок об'єкта діагностування необхідно додатковими контрольними точками в його структурі таким чином, щоб за мінімальних змін була досягнута можливість транспортування необхідних діагностичних сигналів в процесі перевірки будь-якого інтегрального компонента $w_h \in W$ між ним і доступними контрольними точками.

Отже, пропований метод структурної декомпозиції ЦОД орієнтований на застосування на завершальному етапі проектування схем цифрових вузлів або пристроїв і має за мету забезпечення їх тестопридатності відповідно до заданих вимог щодо глибини діагностування через визначення мінімально-достатнього набору контрольних точок для реалізації діагностичних

перевірок, а також шляхів і способів транспортування діагностичних даних (сигналів) в структурі об'єкта діагностування.

Для застосування методу структурної декомпозиції ЦОД і визначення мінімально-достатнього набору контрольних точок для реалізації діагностичних перевірок необхідно сформувати загальну множину наявних контактів ЦОД, які можуть бути використані в якості контрольних точок при виконанні діагностичних випробувань. Такими контактами автоматично стають всі штатні контрольні точки ЦОД, якими є контакти комутаційні роз'ємів і з'єднувачів, а також до числа таких контактів з можливістю їх переведення в категорію контрольних точок слід віднести сигнальні контакти (лінії зв'язку) всіх наявних в складі об'єкта діагностування інтегральних компонентів $w_h \in W$.

Відповідно до запропонованої математичної моделі методу, загальна множина всіх перспективно-доступних для використання в тестових випробувань контрольних точок ЦОД визначена як $Q_{кт}$:$\{q_1,q_2,\ldots,q_i,\ldots,q_p\}$.

Оскільки кількість контактів інтегральних компонентів $w_h \in W$ в складі ЦОД може бути досить великою, для зменшення складності реалізації методу структурної декомпозиції ЦОД використаємо спосіб узагальненого представлення контрольних точок, який використовується в структурно-логічних моделях ЦОД і базується на двох положеннях:

- пари контактів двох інтегральних компонентів $w_h \in W$ і $w_g \in W$, які з'єднані однією лінією зв'язку, в сукупності із зазначеною лінією зв'язку розглядаються як одна контрольна точка на зазначеній лінії зв'язку;

- всі контакти інтегральних компонентів $w_h \in W$, які з'єднані однією лінією зв'язку з штатною контрольною точкою об'єкта діагностування вважаються приналежними цій контрольній точці і відокремлено не розглядаються;

- групи контактів інтегральних компонентів $w_h \in W$, що мають спільне призначення і використовуються завжди групою в однакових режимах роботи, розглядаються як одна групова контрольна точка.

На підставі наведених положень методу структурної декомпозиції цифрових об'єктів діагностування можна перейти його алгоритмічної реалізації.

Перелік посилань

1. Лєнков Є.С. Узагальнена математична модель процесу технічного обслуговування і ремонту складної техніки / Є.С. Лєнков // Вісник Хмельницького національного університету. Технічні науки. – Хмельницький : ХНУ, 2017. – № 2 (247). – С. 186-191.

2. Гунченко Ю.О. Дослідження структури цифрового об'єкта діагностування на основі граф-моделі / Ю.О. Гунченко, Є.С. Лєнков, В.М. Чешун, С.О. Прокопчук //Сучасна спеціальна техніка. Науково практичний

журнал. – Харків,2016. – Вип.№2(45),2016р. – С53-58.

3. Стецюк О.І. Функціональний підхід в діагностуванні цифрових процесорів і елементів пам'яті / Р.В. Кравчук, О.І. Стецюк, В.М. Чешун // Міжнародний науково-технічний журнал «Вимірювальна та обчислювальна техн іка в технологічних процесах». – Хмельницький: ХНУ, 2018. – Вип. №2 (62) – С.106-109.

4. Кушнєрова Н.І. Вибір та обґрунтування методу тестового діагностування елементів системи попередження нештатних ситуацій на борту повітряного судна / Н.І. Кушнєрова // Системи управління, навігації та зв'язку – Полтава : ПНТУ, 2013. – Вип. 1 (25). – С. 86-89.

5. Шевченко В.В. Визначення технічного стану цифрових типових елементів заміни за допомогою електромагнітного методу діагностування / В.В. Шевченко // Сучасні інформаційні технології у сфері безпеки та оборони. – 2015. – № 1. – С. 131-135.

6. Кон Е.Л. Подходы к тестовому диагностированию цифровых устройств / Е.Л. Кон, В.И. Фрейман // Вестник ПНИПУ. Электротехника, информационные технологии, системы управления. – Пермь: ПНИПУ, 2012. – № 6. – С. 231-241.

7. Волков Ю.В. Системы технического диагностирования, автоматического управления и защиты: учебное пособие. Часть 1 / Ю.В. Волков – СПб. : ВШТЭ СПбГУПТД., 2016. – 115 с.

8. Тюрин С.Ф. Разработка контрольных и диагностических тестов для КМОП элементов с избыточным базисом / С.Ф. Тюрин, О.А. Громов // Приволжский научный вестник. – Ижевск : ИЦНП, 2013. –No 1 (17). – С.13-21.

9. Wu Chi-Feng Fault simulation and test algorithm generation for random access memories / Chi-Feng Wu, Chih-Tsun Huang, Kuo-Liang Cheng, Cheng-Wen Wu //IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2002. – Vol.: 21, Issue: 4. P. 480-490.

10. Li Jin-Fu March-based RAM diagnosis algorithms for stuck-at and coupling faults / Jin-Fu Li, Kuo-Liang Cheng, Chih-Tsun Huang, Cheng-Wen Wu //IEEE Trans. on Fuzzy Systems. 2002. – Vol. 10, Issue 2. – P. 155-170.

11. Bushnell M. Essentials of Electronic Testing for Digital, Memory & Mixed-Signal VLSI Circuits / M. Bushnell, V. Agrawal – Kluwer Academic Publishers, 2000 – 695 p.

12. Rayudu K. V. B. V. Functional testing technique for Microprocessor Interface board / K. V. B. V. Rayudu //2015 International Conference on VLSI Systems, Architecture, Technology and Applications (VLSI-SATA) – P. 1-5.

Вставлення маскуючих елементів у відкритий текст

Лабунський В.А.
Науковий керівник – к.т.н.,доц. Муляр І.В.
Хмельницький національний університет

Ефективність та надійність шифрів необхідно розглядати крізь призму часу. Існує велика кількість цікавих шифрів, розроблених в минулі століття, які тривалий час вважали неефективними через складність і трудоємність виконання арифметичних перетворень, невисоку продуктивність роботи криптографа тощо. На сучасному етапі розвитку криптографії варто враховувати, що шифрування інформації і дешифрування виконується з використанням обчислювальної техніки, що дозволяє актуалізувати використання шифрів. Необхідно також брати до уваги, що зловмисник завжди може отримати шифрований текст, також використовуючи засоби обчислювальної техніки.

Ефективність криптосистеми (алгоритм шифрування та дешифрування, або шифр) в значній мірі визначається трудоємністю і часом, який витрачається на шифрування та дешифрування тексту. Надійність криптосистеми визначається часом, який злодій витратить для розкриття алгоритму шифрування і дешифрування, а також щоб знайти ключ шифру. Очевидно, що ефективність і надійність забезпечити одночасно складно - ідеальних шифрів не існує. Необхідно врахувати, що конкретні ситуації висувають свої вимоги до криптосистем. Наприклад, біржова інформація перестає бути таємною через пару десятків хвилин, але повинна бути зашифрована і передана за лічені секунди. Іноді інформація повинна зберігатися десятиліттями, проте відсутні вимоги до швидкості її шифрування [1].

Значна кількість методів злому шифрів основана на класичних дослідженнях шифрованих текстів: частотному аналізі використання одного символу; частотному аналізі біграм; частотному аналізі триграм і характеристиках повторень трьох, чотирьох і більше символів тексту. Основні криптоаналізи базуються на ґрунтовних дослідженнях популярних шифрів. В багатьох країнах світу на початку 21 століття були сформовані кібервійська (кібердивізії), які використовуються як для захисту власних інформаційних систем, так і для злому систем опонентів із складними сучасними шифрами.

Окремо варто закцентувати на використанні нових методів злому таких шифрів кібердивізіями: якщо раніше складно було здійснити злом шифру методом перебору декількох мільярдів можливих варіантів ключів, то сучасні складні шифри зламуються кібервійськами за допомогою масованих атак в організованій структурі, яка використовує тисячі чи десятки тисяч одночасно працюючих комп’ютерів у синхронізованій розподіленій системі

пошуку ймовірних ключів. Тому сучасні комп'ютерні шифри, які стають проблемою для окремого криптоаналітика, що ставить за мету зламати шифр, не складають великих проблем для кібердивізій - стоїть питання лише в часовому і інших ресурсах.

В теорії можна створити абсолютно надійні шифри, які дійсно існують. Водночас, стоїть питання зручності їх використання, продуктивності та собівартості таких шифрів. Як стверджував К. Шеннон [2], для створення таких шифрів краще забезпечити приховування власне факту передачі шифрованої інформації, тобто передавати безперервно текст з рівномірно розподіленою частотою вживання символів, на який одночасно накладати відкритий текст, який треба передати. Тобто система синхронно "накладає" і "знімає" шифровані дані на оригінальні. Якщо передавання інформації такою системою організовано цілодобово і є неперервне в часі - через великий об'єм інформації така система дорога у використанні, але практично не може бути взламана навіть кібервійськами [3].

Традиційними стали спроби щодо науково-дослідних розробок у напрямку створення нових шифрів з високими якостями, і, відповідно, пошуком нових методик їх злому. Значна частина відомих шифрів маю свою, вже відому, методику злому, що зумовлено цілим рядом факторів. По-перше, в багатьох країнах світу зняли заборону на публікацію дослідних матеріалів з криптографії, криптології, тобто доступ до таких досліджень є публічним. Подруге, ряд країн сформував кібервійська, які вирішують задачі з криптографії, в першу чергу щодо пошуків шляхів взлому нових шифрів. Використання кібервійськ, які при зломі шифрів застосовують багатовекторну масовану атаку, вимагає нових методів злому, і, відповідно, захисту від таких методів.

Захищаючи інформацію, важливо не лише правильно вибрати шифр і ключ шифрування та тримати їх захищеними, але й приховати криптографічну систему. Як правило, навіть найнадійніша система захисту не може забезпечити необмежене у часі збереження конфіденційності інформації, поготів якщо стоїть задача передавати інформацію відкритими каналами зв'язку, забезпечити її отримання легальними абонентами, а потім дешифрувати.

Досліджуються методи захисту інформації, які окрім забезпечення безперешкодної криптографічної роботи, забезпечують приховування методу шифрування. Для пошуку, зокрема, таких методів шифрування використовуються відомий інструмент дослідження шифрів - аналіз статистичних характеристик шифрованих текстів (які сформовані одним методом і одним ключем) для одного символу, біграм і триграм. Також необхідну інформацію можна отримати, здійснивши аналіз повторень в шифрованому тексті, в тому числі розривчастих. Власне проти таких методів злому здійснювалися дисертаційні дослідження щодо забезпечення

ефективності нових методів шифрування інформації із використанням маскуючи елементів.

Для визначення можливостей використання маскуючих елементів у блокових шифрах доцільно зупинитися на найбільш вживаних шифрах та їх особливостях, звернувши увагу на вимоги до алгоритмів шифрування.

У 1973 році Національне бюро стандартів США (NBS) опублікувало вимоги до криптографічного алгоритму, який міг би використовуватися в якості стандарту. Були сформульовані такі вимоги до алгоритму [4, 13, 29]:

1. Алгоритм повинен забезпечити високий рівень безпеки.
2. Алгоритм повинен бути повністю визначений і зрозумілий.
3. Безпека повинна базуватися на ключі, не залежачи від збереження в таємниці алгоритму.
4. Алгоритм повинен бути доступний усім користувачам.
5. Алгоритм повинен дозволяти виконувати адаптацію до різноманітних варіантів використання.
6. Алгоритм повинен надавати можливість перевірки.
7. Алгоритм повинен бути дозволений для експорту.

Звичайно, що нові способи шифрування інформації повинні відповідати більшості із перелічених вимог, тоді криптографічний алгоритм можна зарахувати до категорії стійких і перспективних.

Основна перевага блокових шифрів забезпечується за рахунок поєднання в процесі шифрування процедур перестановок і підстановок. При розмірі блоку перетворення пару десятків біт стійкість забезпечується величезним обсягом варіантів, які повинні розглядатися при пошуку ключа і алгоритму. Водночас, при розмірі блоків шифрування 128 біт та більше реалізація мережі Фейстеля на 32-розрядних архітектурах може викликати певні труднощі, тому прийнято використовувати модифіковані варіанти цієї конструкції. Зазвичай використовуються 4-х гілкові мережі. Інформаційний блок ділиться на дві рівні частини $Ь_0$ та $Я_0$, які за тактами перетворюються та переставляються.

У новому методі шифрування інформації виконують поділ символів відкритого тексту (ВТ) на блоки по ц символів у блоці, які утворюють матрицю-стовпець, а ключ формують з $ц^2$ кількості символів, які записують як квадратна матриця цхц. Символи шифрованого тексту (ШТ) формуються в процесі поблокового перемноження матриці-стовпця і квадратної матриці ключа шифрування, які попередньо перетворюють у відповідні числа по модулю п, де п - кількість символів ВТ (потужність алфавіту відкритого тексту). Дешифрування шифрованого тексту виконують поділом символів ШТ на блоки (по ц символів у блоку) і перемноженням матриці-стовпця і квадратної матриці-ключа дешифрування, які перетворюють у відповідні числа по модулю п, де п - кількість символів ВТ.

Згідно запропонованого методу, перед множенням на матрицю-ключ

шифрування у відкритий текст перед і після кожного символу ВТ вставляють додаткові маскуючі елементи, причому маскуючі елементи на кожному кроці вставлення визначаються найменшою частотою вживання цього елементу (із врахуванням вставлених маскуючих елементів) у відкритому тексті з маскуючими елементами, а при дешифруванні вилучають маскуючі елементи в такому порядку, як вони вставлялися перед множенням на матрицю-ключ шифрування.

Рисунок 1 - Загальна схема алгоритму вставлення маскуючих елементів

Процедурі шифрування передує вставлення перед і після кожного символу ВТ додаткових маскуючих елементів, причому при довжині блоку шифрування ц необхідно вставляти таку кількість маскуючих елементів (перед і після символу ВТ), щоб в кожний блок шифрування потрапив хоча б один символ ВТ. Хоча ця вимога не є критичною для виконання, занадто

багато маскуючих елементів вставляти недоцільно, оскільки досягнення позитивного результату можливе бути при незначному збільшенні кількості символів шифрованого тексту (ШТ).

Одним із елементів криптосистеми є генератор псевдовипадкових символів. Додаткові маскуючі елементи вибираються керованим генератором псевдовипадкових чисел таким чином, щоб статистичний аналіз ВТ до і після вставлення маскуючих елементів змінювався в сторону рівномірної частоти вживання символів. Генератор псевдовипадкових чисел на кожному кроці вставлення елементу вибирає такий символ, який має найменшу частоту вживання. Частота використання символів визначається на кожному кроці, і з кожним кроком частотна характеристика ВТ з маскуючими елементами стає все більш рівномірною, що унеможливлює отримання однозначного результату при обробці статистичних параметрів тексту.

Маскуючі елементи у кожному конкретному випадку встановлюються відповідно до обраного методу. Способів встановлення маскуючих елементів може бути багато, а тому алгоритм їх встановлення є доповнювальним захистом до вибору матриці-ключа, оскільки без знання алгоритму встановлення, і, відповідно, вилучення маскуючих елементів, не можливо отримати ВТ. Виключно сукупність відомостей про метод вставлення маскуючих елементів і ключ дає можливість дешифрувати ШТ.

Перелік посилань

1. Granzer W. Secure Communication in Home and Building Automation Systems: dissertation / Granzer W. - Wien, 2010. - 210 p.

2.Stallings W.. Computer security: principles and practice / William Stallings, Lawrie Brown.—2nd ed. - Pearson. - 2012. - 817 p.

3. Олійник Г.В. Дослідження використання інтелектуального програмного комплексу для захисту комп'ютерних мереж / Г.В. Олійник, С.В.Грибков // Вісник Національного університету "Львівська політехніка" "Комп'ютерні системи та мережі". - 2014. -№806. - с.208-213.

4. Якименко І. З. Аналіз ефективності захисту інформації на основі криптографічних перетворень з використанням маскованого представлення даних / Якименко І.З., Божик С.В. // АСІТ'5. "Сучасні комп'ютерні інформаційні технології". ТНЕУ. - Тернопіль. 22-23 травня 2015. - С. 182-184.

Ожиганов А.А. Криптографические системы с секретным и открытым ключом / Ожиганов А.А. - Санкт-Петербург, Университет ИТМО, 2015. - 318с.

Специфікація відбитків пальців (fingerprinting) TCP/IP

Лісовський О.С.
Науковий керівник – к.т.н.,доц. Муляр І. В.
Хмельницький національний університет

Важливим процесом в рамках реалізації функцій мережного моніторингу є збір відомостей про стан вузлів мережі. Для забезпечення ефективного вирішення завдань аналізу необхідно отримати характеристики широкого спектру показників, що описують роботу як всієї комп'ютерної мережі, так і окремих її компонентів.

Зняття відбитків пальців стека TCP/IP (TCP/IP stack fingerprinting) - пасивна колекція ознак конфігурації від віддаленого пристрою під час стандартного шару 4 мережевих комунікації [1]. Комбінація параметрів може тоді використовуватися, щоб вивести операційну систему віддаленої машини (інакше, зняття відбитків пальців OS), або включатися у відбиток пальця пристрою.

OS fingerprinting (OSF) - метод отримання інформації про ОС. OSF актуальне на початковому етапі реалізації атаки на хост. Так як маючи інформацію про тип ОС атакуючий може планувати на які відомі уразливості він буде впливати. При цьому, чим точніше атакуючий визначить тип і версію ОС віддаленого хоста, тим ефективніше буде виконано його "злом". Адміністратори йдуть на всілякі хитрощі, щоб виключити точне визначення своєї ОС. І для того, щоб точно визначити ОС доводиться використовувати комплексний підхід, що власне і описано в цьому документі. Сам процес визначення ОС не можна уявити не описавши методи сканування. Після застосування яких складається відбиток системи (fingerprint) по якому вже з бази заздалегідь відомих відбитків вибирається відповідність. OS fingerprinting буває двох видів - активний і пасивний. Активний OSF - це визначення типу ОС шляхом відсилання пакетів на досліджуваний хост.

Певні параметри в межах визначення протоколу TCP залишаються до реалізації. Різні операційні системи та різні версії однієї операційної системи встановлюють різні значення за замовчуванням для цих значень. Збираючи та вивчаючи ці значення, можна диференціюватися між різними операційними системами та реалізаціями TCP/IP. Поля TCP/IP, які можуть відрізнятися, включають:

- Початковий розмір пакета (16 біт)
- Початковий TTL (8 біт)
- Розмір вікна (16 біт)
- Максимальний розмір сегмента (16 біт)
- Значення масштабування вікон (8 біт)
- Прапор "не фрагментувати" (1 біт)
- Прапор "sackOK" (1 біт)
- прапор "nop" (1 біт)

Ці значення можуть бути об'єднані для формування 67-розрядного підпису або відбитка пальця для цільової машини. [1] Досить просто перевірити початкові поля TTL та розмір вікон, щоб успішно визначити операційну систему, що полегшує завдання виконання відбитків пальців вручну на ОС [2].

Інструменти для відбитків пальців мережі:

– Ettercap - пасивний відбиток стека TCP/IP.

– NetworkMiner - пасивний відбиток пальців стека DHCP та TCP/IP (поєднує бази даних p0f, Ettercap та Satori)

– Nmap - всебічний відбиток пальців активного стека.

– p0f - всебічний пасивний відбиток пальців TCP/IP.

– NetSleuth - безкоштовний пасивний інструмент для відбитків пальців та аналізу

– PacketFence [38] - відкритий код NAC з пасивним відбитком пальців DHCP.

– PRADS - Пасивний комплексний відбиток пальців TCP/IP та виявлення послуги

– Satori - пасивні CDP, DHCP, ICMP, HPSP, HTTP, TCP/IP та інші відбитки стека.

– SinFP - однопортовий активний / пасивний відбиток пальців.

– XProbe2 - активний відбиток стека TCP/IP.

– Веб-сайт пристрою для відбитків пальців[3] - відображає пасивний відбиток TCP SYN комп'ютера вашого браузера (або проміжного проксі)

– Queso - відомий інструмент з кінця 1990-х, який більше не оновлюється для сучасних операційних систем

Витяг значень RTO реалізується в такий спосіб. Клієнт відправляє запит SYN на встановлення з'єднання, після чого фіксує моменти часу, що відповідають отриманню пакетів SYN-ACK. Пакет з установленим прапором ACK при цьому не відправляється, тим самим клієнт імітує ситуацію втрати пакетів, що запускає механізм повторних передач на віддаленому вузлі. Послідовність значень часових інтервалів між посилками повторних пакетів SYN-ACK, а також кількість повторних передач є характеристиками, відмінними для різних реалізацій стека протоколів TCP/IP. Таким чином, аналіз даних значень дозволяє ідентифікувати версію ОС віддаленого вузла.

Аналіз значень RTO для ситуації розриву TCP-з'єднання. Процес завершення TCP-з'єднання передбачає процедуру обміну чотирма пакетами між учасниками з'єднання. Для реалізації функцій IOC виконується аналіз послідовності значень часових інтервалів між повторними передачами пакету FIN, що відправляється віддаленим вузлом. З метою вилучення значень RTO клієнт імітує ситуацію втрати пакетів, які не відправляючи заключний пакет ACK [4].

Сканування з встановленням наполовину відкритого з'єднання (SYN)/ Ідея сканування з встановленням наполовину відкритого з'єднання (також відомого як SYN-сканування) дуже проста. Без завершення трьохетапного

встановлення TCP з'єднання, надсилається пакет SYN і очікується відповідь. Якщо відповідь одержано SYN ACK, це означає, що віддалений порт відкритий, в іншому випадку, якщо отриманий пакет з прапором RST, порт закритий [5].

Однак, в деяких випадках, міжмережевий екран може просто заблокувати доступ до деяких відкритим портів. У цих випадках кажуть, що порт фільтровано. У таких ситуаціях ми не отримаємо відповідь на наш пакет SYN. Також багато ME блокують RST пакети, які є відповіддю на закритий порт. У таких ситуаціях складно зрозуміти, які порти закриті, а які фільтруватися. Нижче наводяться результати сканування з допомогою nmap хоста без ME.

root@life# nmap -P0 -p 1,2,21,80 202.83.174.99

Interesting ports on (202.83.174.99):
PORT STATE SERVICE
1/tcp closed tcpmux
2/tcp closed compressnet
21/tcp open ftp
80/tcp open http
Nmap finished: 1 IP address (1 host up) scanned in 1.140 seconds

Як можна помітити, даний хост не схожий на що знаходиться за ME. Ми просканували порти 1, 2, 21, 80, і встановили, що порти 1 і 2 закриті, а що залишилися два відкриті.

При активному зняття відбитка виробляється відправка довільних пакетів на цільової хост і робиться спроба визначення ОС на підставі таких значень полів заголовка відповідних TCP/IP пакетів, як тимчасові характеристики або IPID, TOS, TCP ISN, прапор фрагментації і т.д. інший старий метод визначення віддаленої ОС полягає в аналізі значення TTL ICMP echo-пакета. Це простий спосіб, однак він не може виявити відмінності різних варіантів однієї і тієї ж ОС, наприклад win98, XP і win2k. Зазвичай в кожній ОС встановлено фіксований, заздалегідь певне, значення TTL. В операційних системах Microsoft це значення за замовчуванням дорівнює 128, тоді як в Linux - 256. Нижче показаний приклад визначення віддаленої ОС за значенням TTL відповідного ICMP echo-пакета. Я просто пінгую цільову машину і перевіряю значення TTL отриманого у відповідь пакету. В даному випадку воно дорівнює 113, що дозволяє припустити, що віддалена ОС належить до сімейства Windows, так як стартове значення TTL цих систем дорівнює 128, а маршрут від моєї машини до цільової становить приблизно 15 проміжних хостів (113 + 15 = 128), що може бути перевірено за допомогою traceroute.

E:\>ping 209.41.165.180

Pinging 209.41.165.180 with 32 bytes of data:
Reply from 209.41.165.180: bytes=32 time 38ms TTL=113

Reply from 209.41.165.180: bytes=32 time 51ms TTL=113
Ping statistics for 209.41.165.180:
Packets: Sent = 2, Received = 2, Lost = 0 (0% loss),
Approximate round trip times in milliseconds:
Minimum = 38ms, Maximum = 51ms, Average = 44ms

Тепер спробуємо застосувати описану вище SYN ACK методику. Для початку ми повинні налаштувати локальний MCE на тихе блокування всіх SYN ACK пакетів, що приходять з віддаленого хоста.

life1# iptables -A INPUT -p tcp -j DROP -s 202.83.162.27

Тепер відправимо SYN пакет на відкритий 80 порт і почнемо стеження за виводом tcpdump

root@life# hping -S -p 80 -c 1 202.83.162.27

17:22:51.079596 202.134.134.230.1816 > 202.83.162.27.http: S win 512
17:22:51.208938 202.83.162.27.http > 202.134.134.230.1816: S ack win 5840
17:22:53.218939 202.83.162.27.http > 202.134.134.230.1816: S ack win 5840
17:23:57.218939 202.83.162.27.http > 202.134.134.230.1816: S ack win 5840
17:23:03.218939 202.83.162.27.http > 202.134.134.230.1816: S ack win 5840
17:23:11.468939 202.83.162.27.http > 202.134.134.230.1816: S ack win 5840
17:24:21.618938 202.83.162.27.http > 202.134.134.230.1816: S ack win 5840

Різниця в часі, між отриманням SYN ACK пакетів становить приблизно 2, 4, 5, 7 і 10 секунд.

Експерименти з іншими хостами показали, що затримки повторної передачі SYN ACK пакетів системи FreeBSD складають приблизно 3, 6, 12, 24 секунди, а Windows-хостів відповідно 2, 4, 6, 8, 10 секунд. Це інформація може бути корисною для правильного впізнання операційної системи в тих випадках, коли інші методи зазнають невдачі і дають невірні результати. Зверніть увагу, представлені вище значення не дуже точні і були отримані в результаті дослідів з двома хостами, на одному з яких встановлена indows 2010, а на іншому FreeBSD 4.6.

Автоматизовані способи зняття відбитків віддаленої системи можуть давати непогані результати, проте в деяких середовищах вони не завжди ефективні. У цих випадках для отримання найбільш точних результатів потрібно комбінувати кілька різних методик.

Більш досконалий метод IOC, заснований на аналізі значень RTO стека протоколів TCP/IP для ситуації втрати пакетів при передачі даних, є продовженням ідей Франка Вейсета і вперше був згаданий в роботі Грега Талек (Greg Taleck) [28] в 2004 р. Це метод Synscan.

Суть методу Synscan полягає в вимірі і аналізі значень RTO, характерних для відпрацювання механізму повторних передач втрачених пакетів з даними, переданими по TCP-з'єднання, наприклад, в процесі взаємодії по протоколу HTTP.

У прикладі на рисунку 2.1 числа, розділені двокрапкою, вказують на початок і кінець переданої в даному пакеті послідовності байт даних (тобто відповідають відносному номеру послідовності і довжині поля даних). Числа після «АСК» відповідають відносним номерами підтвердження, які встановлюються в переданих пакетах.

Обмін даними в представленому прикладі відбувається наступним чином:

1. Виконується успішне встановлення TCP-з'єднання з веб-сервером, і клієнт по протоколу HTTP запитує з сервера індексний файл.

2. Сервер починає відправку пакетів даних.

3. Відправкою АСК-пакета клієнт підтверджує успішне отримання кількох перших пакетів даних від сервера, після чого припиняє відправляти АСК-пакети.

4. Після закінчення часу очікування повторної передачі першого недоставленого пакета (в розглянутому прикладі пакет даних з номером послідовності 1025) сервер починає цикл його повторних передач.

Результати дослідження свідчить про наявність відмінностей між сигнатурами Synscan для різних ОС, в тому числі для ОС одного і того ж сімейства. Важливою відмінністю сигнатур Synscan від сигнатур RING є більший обсяг доступних для аналізу даних, що дозволяє визначати версію ОС цільового вузла з більш високою вірогідністю. Проте, сигнатури Synscan деяких ОС, в тому числі що належать різних сімейств, ідентичні, що в ряді випадків призведе до неоднозначності результатів аналізу і зниження вірогідності припущення про версії ОС цільового вузла.

Перелік посилань

1. Murphy, S. An Application of Deception in Cyberspace: Operation System Ob-fuscation / S. Murphy, T. McDonald, R. Mills // Proceedings of the 5th International Conference on Information Warfare and Security. - Ohio, 2010. - pp. 241-250.

2. Лавров, А. А. Алгоритмы классификации в задаче идентификации версии ОС удаленного сетевого узла / А. А. Лавров // Сб. тр. 65-й науч.-техн. конф. проф.-преп. состава СПбГЭТУ «ЛЭТИ». - СПб., 2012. - 102-106 с.

3. Chang, C.-C. LIBSVM: A library for support vector machines / C.-C. Chang, C-J. Lin // ACM Transactions on Intelligent Systems and Technology. - 2011. -Vol. 2, Issue 3. - Article No. 27.

4. Кореньков В. В. Архитектура и пути реализации системы локального мониторинга ресурсного центра / В. В. Кореньков, П. В. Дмитриенко // Системный анализ в науке и образовании. - Дубна, 2011. – 201-204 с.

5. Максимов, Н. В. Компьютерные сети / Н. В. Максимов, И. И. Попов. - М.: Форум, 2010. - 464 с.

Мережева модель представлення предметної області

Маковей А.А.
Науковий керівник – к.т.н., доц. Джулій В.М.
Хмельницький національний університет

З розвитком комп'ютерних і web-технологій все більш широке застосування знаходить електронний (віддалений) режим навчання. Однак, в більшості випадків, реалізація такого режиму навчання заснована на забезпеченні того, хто навчається великим об'ємом статичних електронних ресурсів без урахування здібностей учня, і без підтримки активного навчального процесу. В роботі [2] робиться припущення, що навчаючий, активно зацікавлений в навчальному процесі, ймовірніше досягне успіху.

Для управління навчальними ресурсами може застосовуватися семантична модель мережі знань, заснована на онтологіях і тематичних картах (рис. 1). Тематичні карти зазвичай використовуються для представлення та організації знань у такий спосіб, який може бути оптимізований для навігації. Модель мережі знань включає в себе наступні елементи: теми представляють елементи знань. Кожна тема має додаткові атрибути, такі як приналежність до категорії і т.д.; зв'язки визначають відношення між темами, включаючи генералізацію, агрегацію та посилання (наприклад, причинно-наслідкові зв'язки, аналогії і т.д.); навчальні матеріали. Кожна тема може бути пов'язана з одним або більше навчальними матеріалами. Атрибутами можуть бути ключові слова,тип елемента, педагогічне призначення, рівень складності і т.д.

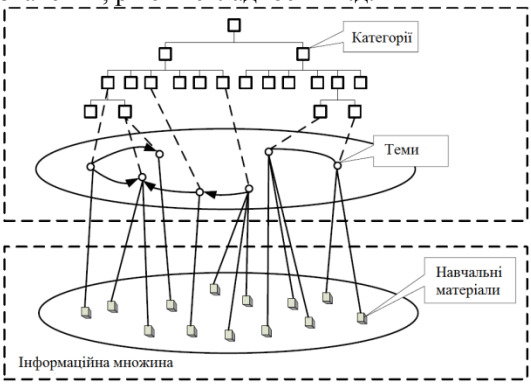

Рисунок 1 - Модель мережі знань

В рамках ІНС адаптивні навчальні курси (індивідуальні траєкторії навчання) можуть бути згенеровані на основі онтології, зв'язків між темами і профілю, хто навчається. У відповідь на запит, сформований навчаючим і відображаючим мету навчання, інформаційно-навчальною системою генерується індивідуальна траєкторія, що направляє навчання відповідно до

його цілі. Для цього пропонується вибрати теми, відповідні цілі навчання, виконати зворотний навігаційний алгоритм для отримання необхідних умов зв'язування тем. На цьому етапі можливо розширення списку тем, обраних на попередньому етапі. Потім здійснюється вибір інформаційних елементів (матеріалів), в недостатній мірі вивчених навчаючим, і формується послідовність вивчення матеріалів з гнучкою структурою навігації.

Подання траєкторії навчання за допомогою мережевих графіків.

Для наочного представлення траєкторій застосування інформаційних елементів (ІЕ) можна застосовувати мережеві графіки. Мережевий графік представляє собою динамічну модель виробничого процесу, що відображає технологічну залежність і послідовність виконання комплексу робіт з урахуванням витрат ресурсів і вартості робіт. Існують два типи мережевих графіків: вершини графа відображають стан об'єкта (події), а дуги - роботи, що ведуться на цьому об'єкті. Кожній дузі зіставляється час, за який здійснюється робота і / або витрачаються ресурси; вершини графа відображають роботи, а зв'язки між ними – залежності між роботами. В такому графі кожен вузол, як і робота, характеризується рядом атрибутів, як тривалість роботи, ранній час початку, пізній час початку.

Стосовно до задачі представлення траєкторій навчання, вершини представляють інформаційні елементи (або інформаційні фрагменти - ІФ), що характеризуються трудомісткістю, а дидактичні зв'язки між ІЕ відображаються дугами. Таким чином, для розглянутої задачі підходять мережеві графіки другого типу. В цьому випадку інформаційні елементи розташовуються у вузлах, а дидактичні зв'язки відображаються дугами і відповідають переходам між ними (рис. 2). Якщо розглядати як приклад траєкторії навчання навчальний план, інформаційними фрагментами будуть дисципліни, для яких вказані трудомісткості (в годинах або залікових одиницях трудомісткості - ЗОТ), дидактичні зв'язки між дисциплінами.

Для побудови мережевого графіка вводиться початкова фіктивна робота, наявність якої обумовлено паралельним вивченням кількох дисциплін в першому семестрі, для яких в мережевому графіку немає попередніх. Також в даному випадку, цю роботи можна розглядати як процес зарахування абітурієнта до ВНЗ. Кінцевою роботою мережного графіка є захист ВКР, а при наявності підсумкового державного іспиту графік буде закінчуватися цими двома послідовно з’єднаними роботами.

При такому розгляді завдання для її вирішення може бути застосований метод критичного шляху. При цьому необхідно відразу враховувати, що довжина критичного шляху не може бути більше заданої кількості семестрів, і якщо ця умова не виконується, то потрібна експертна або людино-машинна процедура для перестроювання мережевого графіка, тобто внесення змін до початкової умови задачі, що стосуються тих дисциплін, які лежать на критичному шляху.

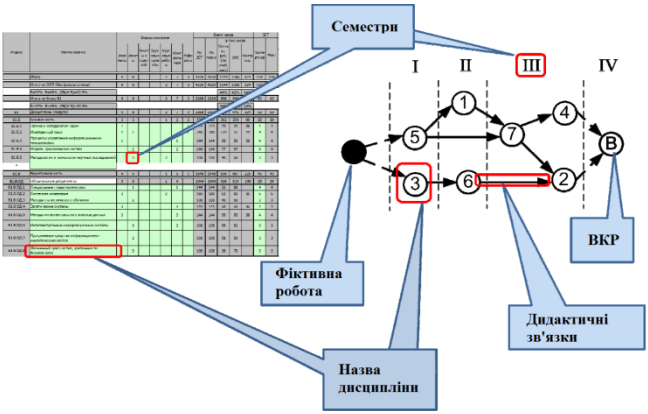

Рисунок 2 - Представлення навчального плану мережевим графіком

Цінність інформації, яка визначається в рамках прагматичного підходу, безпосередньо залежить від переслідуваної мети. Чим більшою мірою інформація допомагає досягненню мети, тим ціннішою вона вважається. На відміну від Шенноновського визначення кількості інформації, переданої по каналах зв'язку, цінність проявляється в результаті рецепції і, отже, безпосередньо з нею пов'язана. Іншими словами, цінність інформації залежить також і від рівня підготовки, попереднього запасу інформації - тезауруса. ця залежність наочно проілюстрована на прикладі цінності підручника з теорії ймовірностей. Якщо перед навчаючими ставиться завдання освоєння того чи іншого нового матеріалу, то буває важко самостійно визначити якими базовими знаннями, наприклад, термінологічним запасом, потрібно володіти, щоб успішно сприйняти нову інформацію. Подібні завдання виникають у студентів при вивченні нового для них матеріалу в рамках самостійної роботи або дистанційної освіти, особливо, якщо для цього використовується електронний навчальний контент, представлений на тому чи іншому освітньому ресурсі. Ту ж задачу вирішують викладачі ВНЗ, що формують тематичні плани дисциплін. Так само, потреба у вивченні великого обсягу слабо структурованої документації можуть відчувати розробники, приходячи в нові проекти. З метою допомогти їм швидше освоїти основні поняття і положення проекту може бути розроблена мережа документів для розробників. Часткова автоматизація перерахованих завдань досягається за рахунок застосування систем аналізу текстів, які здійснюють витяг індексу тексту, автоматичне формування множини рубрик, кластеризації множини текстів, віднесення тексту до рубрики (класифікація), порівняння текстів і т.д..

Підвищення ефективності роботи в тій чи іншій предметній області, наприклад вивчення навчально-довідкових матеріалів (таких, як ресурси wiki,

словники BaseGroup, а також глоссаріїв різних програмних пакетів - Statistica, Mathcad, Matlab і ін.) можливо за рахунок визначення найбільш логічного порядку розгляду, при якому отриманих раніше знань буде достатньо для роботи з кожним наступним об'єктом предметної області. Для цього потрібно структура відповідної предметної області. Прикладами моделей таких структур є класифікатори УДК, ДРНТІ і т.п. Однак вони охоплюють широку предметну область, за рахунок чого складні для пошуку окремого розділу. Іншим мінусом є недостатня деталізація таких класифікаторів - вони закінчуються практично на тому рівні, з якого починаються класифікації понять предметних областей прикладних задач. Довідкові інформаційні ресурси мають свою внутрішню структуру, але, як правило, впорядковані лише за алфавітом. Побудувати семантичну модель на рівні прикладної задачі вручну, не маючи достатніх знань у відповідній предметній області, досить важко. В рамках освітньої діяльності завдання впорядкованості інформаційних елементів/фрагментів може виникати при вирішенні задач складання навчальних планів у навчальних закладах, формування лекційних курсів, упорядкування навчально-довідкових матеріалів або статей, складання змісту навчального посібника, складання змісту випуску наукового журналу тощо.

Таким чином, між вершинами в мережевий моделі предметної області можуть бути встановлені дидактичні зв'язки і відношення впорядкованості, крім того, кожна вершина має деякий набір характеристик, що описують її. Тому доцільним представляється розгляд кожного об'єкта мережі як фрейми, слоти якого містять поля, що відповідають за описані вище характеристики об'єкта. Так само з допомогою окремих слотів можуть бути реалізовані деякі з зв'язків, наприклад, зв'язок «частина-ціле» за допомогою явної вказівки батьківської вершини. Подібним же чином можуть бути реалізовані і дидактичні зв'язки, за допомогою вказівки списку ключових слів, які необхідно знати для успішного освоєння розглянутого інформаційного об'єкта, а також списку понять, на формування яких спрямований даний інформаційний об'єкт.

Перелік посилань

1. Лапшин, В.А. Онтологии в компьютерных системах /В.А. Лапшин – М.: Научный мир, 2010. – 222 с.

2. Левитин, А. В. Алгоритмы. Введение в разработку и анализ / А. В. Левитин – М. : Вильямс, 2006. – 576 с.

3. Мариничева, М.К. Управление знаниями на 100%: путеводитель для практиков / М.К. Мариничева. – М. : Альпина Бизнес Букс, 2008. – 320 с.

Моделювання загроз для хмарного середовища

Мордовин О.С.
Науковий керівник – к.т.н.,доц. Чорненький В.І.
Хмельницький національний університет

Нині до хмарних технологій та реалізації на їх основі середовища хмарних обчислень (ХС) проявляється велика зацікавленість, а технологічно розвинені держави їх уже реалізували та широко застосовують. Використання технологій хмарних обчислень дозволяє досягти ряд переваг, основними з них є такі, як [1]: гнучкість, обчислювальна потужність, великий обсяг файлового сховища, різноманітність програмного забезпечення; повсякчасна можливість доступу до ресурсів в хмарі та швидке розгортання сервісів, можливість збільшення навантаження в хмарі; простота масштабування, резервування та самовідновлення; можливість управління навантаженнями та здійснення моніторингу в реальному часі тощо.

З точки зору здійснення захисту інформації також маються переваги, основними з яких є такі, як [2]:

- практична можливість централізованого керування конфігурацією, рівнем безпеки та здійснення аудиту;
- можливість динамічного масштабування ресурсів системи, резервування та аварійного відновлення при збоях;
- як правило, наявність штатних підрозділів, які повинні забезпечувати безпеку інформації при хмарних обчисленнях;
- централізоване розміщення програмного та програмно-апаратного забезпечень захисту інформації та захисту даних відповідно прийнятих політик безпеки тощо.

Політика безпеки інформації є частиною загальної політики безпеки організації і повинна успадковувати основні її принципи. Головною причиною запровадження політики безпеки зазвичай є вимога наявності такого документа від регулятора — організації, що визначає правила роботи підприємств даної галузі. У цьому випадку відсутність політики може спричинити репресивні дії щодо підприємства або навіть повне припинення його діяльності.

Крім того, певні вимоги (рекомендації) пред'являють галузеві або загальні, місцеві чи міжнародні стандарти. Зазвичай це виражається у вигляді зауважень зовнішніх аудиторів, які проводять перевірки діяльності підприємства. Відсутність політики викликає негативну оцінку, яка в свою чергу впливає на публічні показники підприємства — позиції в рейтингу, рівень надійності і т. Д.

Цікаво, що, згідно з дослідженням з безпеки, проведеного компанією Deloitte в 2006 році, підприємства, які мають формалізовані політики інформаційної безпеки, значно рідше піддаються злому. Це свідчить про те,

що наявність політики є ознакою зрілості підприємства в питаннях інформаційної безпеки.

У середовищі хмарних обчислень користувачі створюють багато динамічних віртуальних організацій, які насамперед ґрунтуються на довірі між цими організаціями.

Концепцію моделі хмарних обчислень часто розглядають двояко, деякі в ній бачать ризики для безпеки і нові «вектори загрози», але разом з тим дана система має новими можливостями для підвищення безпеки. Покращена спостережність інфраструктури, автоматизація та стандартизація - всі ці можливості підвищують рівень захищеності інформації. Наприклад, якщо використовувати заздалегідь заданий набір Qoud-інтерфейсів паралельно з централізованим управлінням ідентифікаційної інформацією, поряд з політикою управління доступом, то ми на порядок зменшуємо ризик доступу клієнтів до небажаних ресурсів. Такі заходи безпеки, як виконання обчислювальних сервісів в ізольованих доменах, використання шифрування до даних, значно підвищують збереження інформації, зменшуючи її втрати. Варто додати, що використання автоматичної ініціалізації і відновлення виконуваних образів скоротять простір для атак, дозволивши вирішувати ряд правових аспектів.

До основних недоліків хмарних технологій можна віднести [4]:

1. Залежність від підключення до мережі (необхідно мати копію вашого документа в хмарі і в локальних папках);
2. Захист персональних даних (не варто зберігати в хмарі конфіденційну інформацію);
3. Не кожне додаток дозволяє зберегти, наприклад, на флешку проміжні етапи обробки інформації;
4. Є ризик, що провайдер онлайнових сервісів одного разу не зробить резервну копію даних, і вони будуть загублені в результаті краху сервера;
5. Довіряючи свої дані онлайн-сервісу, втрачається над ними контроль.

Постійне підключення до мережі - для отримання доступу до послуг «хмари» необхідно постійне з'єднання з мережею Інтернет. Однак у наш час це не такий і великий недолік особливо з приходом технологій стільникового зв'язку 3G і 4G.

Програмне забезпечення та його кастомізація - є обмеження по ПО яке можна розгортати на «хмарах» і надавати його користувачеві. Користувач має обмеження в використовуваному ПО і іноді не має можливості налаштувати його під свої власні цілі.

Ризики часто виникають на інтерактивних вузлах між віртуальними машинами і є динамічним, непередбачуваним процесом. Уся процедура захисту даних побудована на конфіденційності, цілісності та доступності. Конфіденційність належить до так званої прихованої функції фактичних

даних або інформації і є однією із найжорсткіших вимог інформаційної безпеки. У випадку хмарних обчислень дані накопичуються в центрах обробки даних, де безпека та конфіденційність даних ще важливіші.

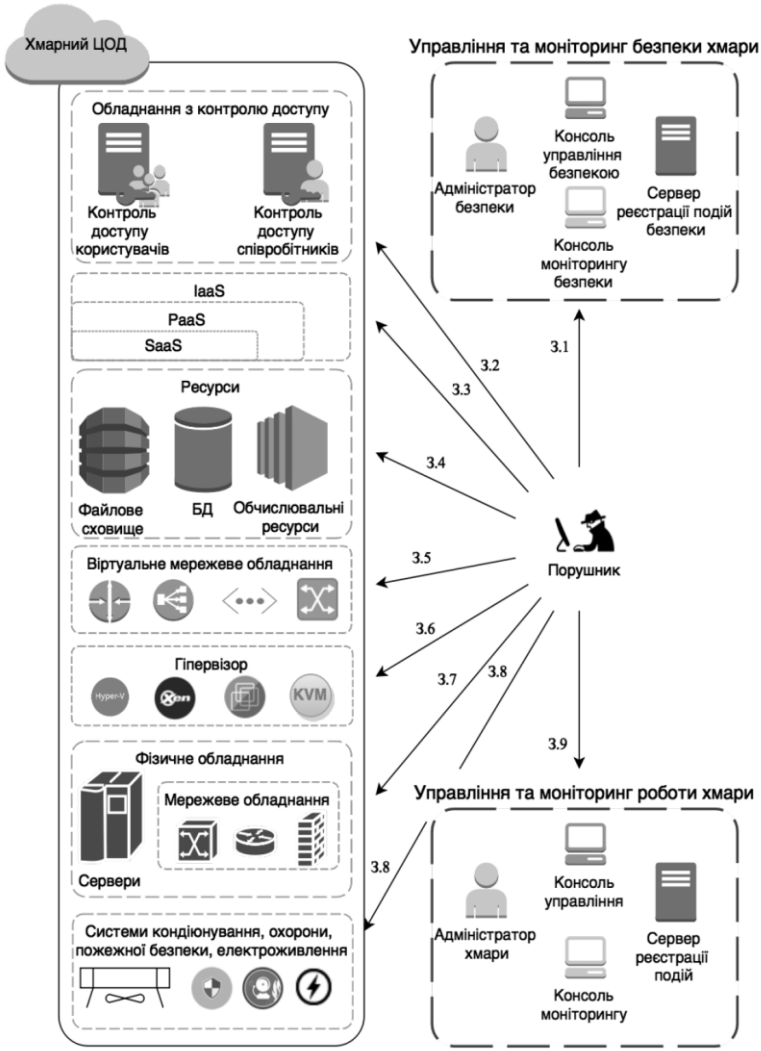

Рисунок 1– Модель загроз для хмарного середовища

Класифікація загроз за ймовірністю була проведена з урахуванням рекомендацій [5]. Згідно цих рекомендацій, найбільшу ймовірність мають загрози, що здійснюються на компоненти хмарної інфраструктури, які мають інтерфейси доступу з зовні та/або знаходяться в віртуалізованому середовищі.

Аналіз моделі загроз, зображеної на рис. 2 , показав, що найбільшу небезпеку становлять загрози управління хмарою (З.9) та її безпекою (З.1), а також загрози гіпервізору (З.6).

Модель ґрунтується на використанні поняття інформаційного віртуального з'єднання як способу опису мережевої взаємодії в ХС. Для опису привілеїв суб'єктів використовується рольова модель, в якій привілеї ролей виражені у формі правил фільтрації інформаційних сервісів для користувачів ХС. Адекватність моделі підтверджується тим, що будь-яке мережеве взаємодія в мережах TCP/IP можна представити у вигляді віртуального з'єднання, і модель включає в себе необхідні параметри для того щоб здійснити контроль мережевих з'єднань на відповідність політиці доступу.

Запропоновані на основі аналізу сучасного стану стандартизації та застосування хмарних сервісів моделі хмарних обчислень, порушника та загроз ІТС хмарних сервісів дозволили встановити, що найбільш проблемними та такими, що вимагають вирішення в частині надання послуг конфіденційності, цілісності, справжності та доступності тощо, є задачі захисту ключів та ключової інформації. Для цього на основі аналізу стану встановлено, що в середовищі хмари відносно ключових даних існують та можуть бути реалізованими такі загрози як компрометація, несанкціоноване знищення, перехоплення та запам’ятовування, нав’язування слабких та несанкціоноване використання тощо ключів. При цьому встановлено, що найбільшу небезпеку в середовищі хмарних обчислень для ключових даних користувача представляють адміністратори хмарних сервісів, які мають доступ до середовища, в якому розгорнуто хмарні додатки користувача.

Також на основі детального аналізу стану та вимог відносно безпечності управління ключами зі сторони нормативно-правових документі та стандартів, включаючи проекти, обґрунтовані механізми захисту конфіденційних, особистих та відкритих ключів користувача від виявленої множини загроз. Вони зводяться до використання для забезпечення високого рівня безпеки, тобто високого рівня ймовірностей реалізації загроз в середовищі хмарних обчислень, комплексу технічних, організаційних та організаційно-технічних заходів та засобів, в тому числі до використання:

- на рівні користувача захищених з необхідним рівнем безпеки ключових носіїв;
- на рівні каналів зв’язку між користувачем та хмарою захищених каналів зв’язку з взаємною автентифі- кацією сторін та стійкістю вищою за

стійкість ключів, що передаються;

- на рівні сервісів ідентифікації, автентифікації, авторизації та керування правами доступом надійних протоколів автентифікації з стійкими криптографічними алгоритмами, а також методів багатофакторної автентифікації;
- для здійснення криптографічних операцій на рівні сервісів додатків та інфраструктури захищених відповідним чином модулів криптографічного захисту - HSM.

Визначений в результаті аналізу перелік загроз та розроблена модель загроз ключовим даним дозволили зробити висновки про те, що порушник з високою ймовірністю може реалізовувати ряд наведених в підрозділі загроз, але найбільша небезпеку в середовищі хмарних обчислень для ключових даних користувача виникає при використанні їх в середині розгорнутою інфраструктури, без застосування криптографічних сервісів.

Перелік посилань

1. Simple Object Access Protocol (SOAP) 1.1 / World Wide Web Consortium [Електронний ресурс]. Режим доступу: http://www.w3.org /TR/2000/NOTE-SOAP-20000508/ (дата звернення 10.10.2019)

2. About Cloud Security Alliance [Електронний ресурс] / Cloud Security Alliance. Режим доступу: https://cloudsecurityalliance.org/about/ (дата звернення 10.10.2019)

3. Заборовский В.С., Сетецентрическая модель и методы контроля доступа к информационным ресурсам в среде облачных вычислений. / В.С. Заборовский, А.А. Лукашин / Научно-технические ведомости СПбГПУ. Информатика, Телекоммуникации, Управление. №2 (145) 2012. - СПб.: Изд-во Политехи. Ун-та, 2012. 183 с.

4. Зикратов, И. А. Оценка информационной безопасности в облачных вычислениях на основе байесовского подхода [Текст] / И. А. Зикратов, С. В. Одегов // Научно-технический вестник информационных технологий, механики и оптики. - 2012. - № 4 (80). - С. 121-126.

5. Hashizume, K. An analysis of security issues for cloud computing [Text] / K. Hashizume, D. Rosado, E. Fernandez-Medina, E. Fernandez // Journal of Internet Services and Application - 2013. - Vol. 4, Issue 5. - P. 15-28. doi: 10.1186/1869-0238-4-5

Оптимізація структури, підвищення доступності в коміркових мережах

Наконечний С.Л.
Науковий керівник – к.т.н.,доц. Хмельницький Ю.В.

На сьогоднішній день проведений аналіз показав, що на процеси надання послуг зв'язку та забезпечення їх неперервності визначальний вплив мають механізми керування мобільністю та балансування абонентського навантаження. Більшість методів балансування абонентського навантаження можна поділити на два основних типи: балансування навантаження за критерієм імовірності блокування запитів та балансування на основі показника завантаженості. Для першого типу характерним є менший об'єм службових даних, оскільки процес балансування починається тільки в момент перевищення порогові значення коефіцієнта блокування запитів. Наприклад, розподіл навантаження між комірками проводиться шляхом зменшення радіусу перевантаженої та збільшення радіусу сусідніх до неї комірок за допомогою регулювання потужності випромінювання базових станцій. Другий тип балансування навантаження є кращим з практичної точки зору, оскільки він враховує пропускну здатність і ступінь балансування навантаження як при виборі комірки. Наприклад, процес балансування навантаження починається з найбільш завантаженої комірки з метою досягнення рівномірного завантаження у мережі. Проте, усі розглянуті механізми в процесі балансування навантаження використовують ресурси сусідніх комірок. Тому такі механізми не підходять для ситуацій, коли в умовах пікових навантажень перебувають цілі групи сусідніх комірок. Звідси випливає формулювання наукового завдання роботи.

У комірковій топології з'єднання існують між усіма точками мережі. Це найбільш надійна та відмово стійка топологія. На жаль, вона також і найбільш дорога. Крім того, із збільшенням числа вузлів кількість з'єднань росте великими темпами. У повній комірковій топології кожен вузол мережі повинен бути з'єднаний з кожним іншим вузлом. У частковій комірковій топології ця вимога не обов'язкова. Надійність мережі з частковою комірковою топологією майже така ж, як і з повною, причому її вартість значно нижче. Як і в зіркоподібної, у багаторівневій глобальній мережі використовуються маршрутизатори, однак ця мережа значно надійніша завдяки тому, що в ній комутатори з'єднані з іншими вузлами каскадною схемою. Багаторівнева глобальна мережа легко розширюється, тому що до неї легко додавати нові вузли і навіть рівні. Ця топологія використовується у великих швидкоростучих мережах. У великих багаторівневих глобальних мережах серйозною проблемою може стати надмірне завантаження окремих ліній. Щоб уникнути цього, потрібно ретельно аналізувати завантаження ліній і розміщати устаткування оптимальним чином [1].

Для підвищення ефективності балансування навантаження

запропоновано модель коміркової структури мережі безпровідного доступу у вигляді графа (без урахування або з урахуванням перекриття несуміжних секторів), де кожен вузол відповідає сектору комірки. З'єднання вузлів позначають існування спільної зони обслуговування для відповідних секторів, що є базовою умовою для можливості балансування навантаження у досліджуваній мережі. Запропоновано метод балансування абонентського навантаження у комірковій мережі доступу, який відрізняється від відомих використанням в процесі балансування ресурсів не тільки сусідніх комірок, а й більш віддалених стосовно цільової, та враховує завантаженість комірок, тип абонентського навантаження, швидкість переміщення абонентського терміналу та ефективність використання каналу зв'язку, та забезпечує перенесення величини абонентського навантаження від перевантаженої до будь-якої доступної недовантаженої комірки без зростання рівня втрат запитів та без зміни параметрів радіомережі. Запропоновано метод оцінювання ефективності використання каналу зв'язку, який, на відміну від відомих, використовує узагальнений критерій частотної та енергетичної ефективності системи «термінал-базова мережа», враховує віддаль між межею ШЕНОНА та точкою, що позначає частотну та енергетичну ефективності системи та дає змогу визначити оптимальну стратегію планування безпровідного доступу. Застосування цього методу забезпечує вибір найбільш оптимального каналу зв'язку для перенесення навантаження в процесі його балансування. На сьогодні розроблено імітаційну модель обслуговування викликів у комірковій мережі безпровідного доступу, яка використовує запропонований метод балансування абонентського навантаження, метод оцінювання ефективності використання каналу зв'язку, а також модель коміркової структури мережі.

Імітаційна модель враховує розподіли швидкостей та напрямів руху абонентів, їх мережну активність в різний час протягом доби, реалізовуючи, таким чином, різні сценарії їх переміщення та генерації навантаження, що забезпечує гнучкість та загальність результатів моделювання. Урахування траєкторії руху абонентів дає змогу прогнозувати завантаження окремих зон коміркової мережі, що забезпечить підвищення ступеня балансування абонентського навантаження. Генерація активності абонентських терміналів відбувається з одночасним урахуванням тривалості комунікаційних сеансів, інтенсивності надходження запитів на їх початок, інтенсивносте початку та завершення їх обслуговування, та максимальної одночасної кількості сеансів у системі, що дає змогу сформувати абонентське навантаження на радіомережу з локальними перевантаженнями та підвищити адекватність моделювання процесів функціонування коміркової мережі безпровідного зв'язку.

Для перевірки ефективності запропонованих моделей та методів проведено моделювання процесу функціонування коміркової мережі без

провідного доступу та встановлено, що удосконалена модель балансування абонентського навантаження у комірковій мережі доступу дає змогу зменшити втрати запитів на послуги до 15% в умовах пікових навантажень на окремі сегменти мережі. У четвертому розділі для підвищення ефективності процесу надання послуг у коміркових мережах доступу запропоновано систему збору та обробки інформації, яка дає змогу зменшити тривалість виконання ітерації запропонованого алгоритму балансування навантаження [2]. Для того, щоб в реальних умовах обмежити розмірність завдання пошуку навантаження, яке підлягає балансуванню, запропоновано класифікацію активних терміналів за швидкістю переміщення. Також запропоновано алгоритм розміщення базових станцій у мережах 5G, який дає змогу зменшити кількість активних базових станцій у мережі шляхом послідовного виключення базових станцій та призначення обслуговуючих БС до окремих користувачів на основі розрахунку зваженої відстані між вибраним користувачем і усіма БС в мережі. Розроблений алгоритм дає змогу зменшити від 30% до 40 % кількість активних БС із збереженням достатньої якості обслуговування для більшості сценаріїв абонентського навантаження.

.У результаті проведеного дослідження відмічено, що для розподілених обчислювальних систем, до яких можна віднести системи відео зв'язку, функціональна надійність і відмово стійкість може забезпечуватися перерозподілом запитів між вузлами кластера. Незважаючи на те, що перерозподіл вносить додаткову затримку, ступінь адаптації системи до зміни потоку запитів збільшується, що призводить до зменшення відмов.

Сьогодні відсутня ймовірнісна модель доступу до систем відео зв'язку з гарантованою доставкою для авторизованих користувачів. Моделі доступу інших авторів в основному орієнтовані на підтримку конфіденційності, а не цілісності та доступності. Існуючі ймовірнісні моделі традиційно використовують в якості критерію надійності коефіцієнт готовності, що визначає працездатність системи, а не цілісність і доступність її ресурсів. В якості критерію надійності систем відео зв'язку раніше авторами не розглядалася ймовірність отримання доступу до ресурсів систем відео зв'язку. Існуючі ймовірнісні моделі доступу зазвичай не враховують характерні особливості систем відео зв'язку або прив'язані до конкретних технологій, наприклад, бездротовий відео зв'язок, що звужує їх область застосування.

Для вирішення можливої проблеми, пов'язаної з блокуванням сервера при втраті пакету з міткою останнього спеціального пакета необхідно припиняти з'єднання з клієнтом по досягненні певного часу з моменту передачі останнього пакету. Окремий випадок роботи алгоритму - єдиний сервер представлений на рисунку 1. Представлений метод підвищення надійності відео зв'язку для авторизованих користувачів з гарантованою доставкою повідомлень. Побудовано ймовірнісні моделі доступу верхнього і нижнього рівня до інформаційних ресурсів систем відео зв'язку. Ймовірнісні

значення доступу дозволяють визначити надійність системи, що раніше було трудомістким завданням (в інших реалізаціях ймовірнісної моделі доступу) або не розглядалося (в логічних моделях доступу). Отримане значення ймовірності отримання кожним суб'єктом повного доступу до кожного об'єкту порівнюється з необхідним значенням ймовірності отримання кожним суб'єктом повного доступу до кожного об'єкту, при необхідності в подальшому застосовується алгоритм управління навантаженням мережі.

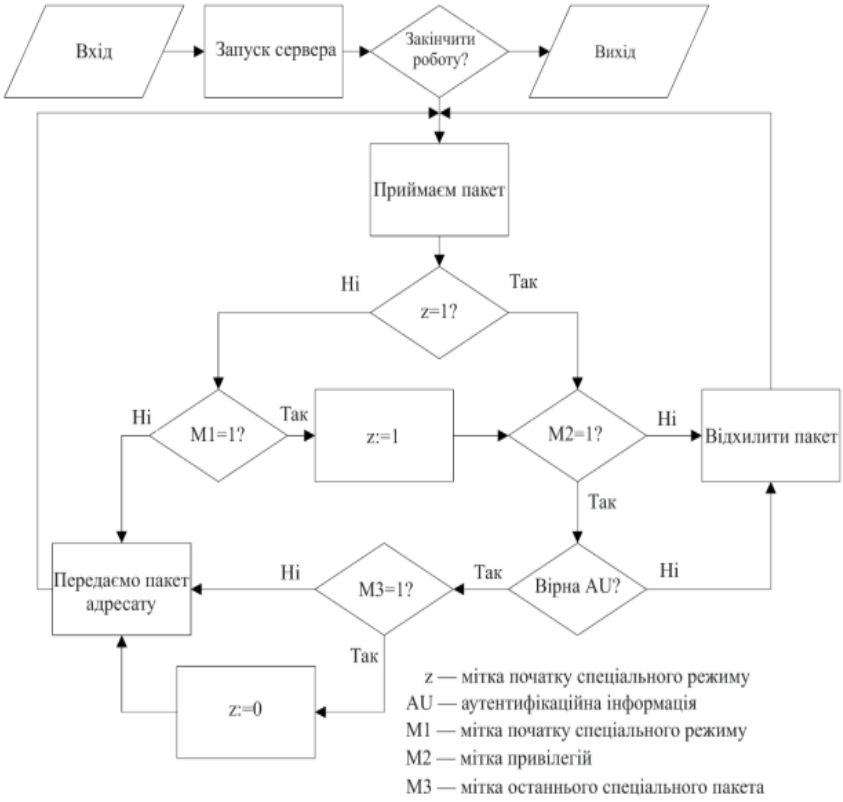

Рисунок 1 – Алгоритм роботи сервера послуг коміркової мережі

Розроблено алгоритм керування навантаженням коміркової мережі, представлено опис алгоритму, докладно розглянуто спеціальний режим, що складається з трьох етапів: підготовка, безпосередньо спеціальний режим і завершення роботи. Більш детально розглянуто окремий випадок роботи алгоритму - мережа з єдиним сервером. Досліджено ефективність запропонованого методу, дослідження показали, що найбільшу ефективність

комп'ютерний метод підвищення надійності відео зв'язку показує при кількості серверів.

Для збільшення пропускної здатності у комірковій мережі на межі дії базової станції передбачено використання технології координованого обслуговування, під якою розуміється обслуговування абонента декількома базовими станціями. Для функціонування координованого обслуговування необхідно, щоб всі станції, які здійснюють прийом чи передачу даних, були синхронізовані по часу і частоті. Дана взаємодія між базовими станціями досягається внаслідок нової архітектури ядра коміркової мережі для стандартів, серед особливостей якої виділимо наявність прямої взаємодії між станціями за допомогою інтерфейсу X2. Технологія координованого обслуговування має два основні варіанти:

- Режим координованої обробки. В даному режимі дані для передачі є на декількох базових станцій.
- Режим координованого планування і режим координованого формування діаграми спрямованості. При даній формі координації прийом чи передача здійснюється від одної базової станції, але при цьому здійснюється координація роботи всіх станцій.

Обмежена кількість частотних ресурсів окремих операторів для впровадження нових технологій в сучасних умовах не дозволяє реалізувати весь потенціал технології коміркових мереж. Доцільнішим буде прийняття рішення про об'єднання частотних ресурсів різних операторів, що дасть можливість побудувати мережу із високим рівнем якості надання послуг. NETWORK SHARING – колективне володіння та експлуатація спільної мережевої інфраструктури або її частин двома чи більше телекомунікаційними операторами. Переваги NETWORK SHARING:

- Зменшення витрат на розгортання мережі.
- Забезпечення оптимального покриття у відносно короткі терміни.
- Можливість надання абонентам послуг із вищою якістю.
- Зведення конкурентної боротьби між операторами до способів ведення їх маркетингової і тарифної політики.
- Можливість збереження існуючої абонентської бази для операторів, які співпрацюють між собою.

Таким чином, можна зробити висновок про те, що запропоновано комплексне застосування методу м'якого повторного використання частот сумісно із режимом координованої обробки для підвищення пропускної здатності коміркової мережі. Здійснено модифікацію методу із метою уникнення конфліктних смуг з технологію GSM. Передбачено необхідність перерозподілу частотних ресурсів GSM між операторами з метою отримання широкої смуги спектру для впровадження технології коміркових мереж. Проаналізовано способи спільної побудови коміркової мережі, по скільки спільна побудова надає змогу виділити більшу смугу для впровадження

технології 4-го покоління та реалізації її максимального потенціалу. На основі порівняльного аналізу способів спільної побудови та економії затрат для операторів вибрано оптимізовану архітектуру коміркових мереж. Проведені дослідження дозволяють операторам зв'язку здійснити максимально ефективне використання та провести оптимізацію структури і підвищити доступність в коміркових мережах, а також провести впровадження технології коміркових мереж по всій території.

Перелік посилань

1. Бак, Р. І. Імітаційна макромодель поведінки абонентів у мережі коміркового зв'язку / Р. І. Бак, П. О. Гуськов, О. А. Лаврів // Вісник Національного університету "Львівська політехніка", серія "Радіоелектроніка та телекомунікації," № 849, 2016. - С.274–284,

2. Хмельницький Ю.В. Забезпечення достовірності передачі інформації та сервісних послуг для високошвидкісних мереж при завадах / Ю.В Хмельницький, Д.П. Яковлєв // Збірник наукових праць Військового інституту Київського національного університету імені Тараса Шевченка. – К.: ВІКНУ, 2017. – Вип. № 57. – С. 111-119

Архітектура програмного комплексу забезпечення безпеки виявлення і протидії DDoS-атакам

Савіцька О.О.
Науковий керівник – к.т.н.,доц. Джулій В.М.
Хмельницький національний університет

Програмний комплекс виявлення початку атаки в режимі реального часу розраховує середньоквадратичне відхилення з урахуванням актуальних сезонних періодів за кількістю запитів до мережного ресурсу в кожному періоді. Програмний комплекс дає можливість задати розмірність розглянутих періодів: 1 хвилина, 15 хвилин, 1 годину і т.д. А також вести моніторинг відразу по декількох періодів. На підставі розрахованого середньоквадратичного відхилення задається верхня межа кількості запитів до мережного ресурсу відповідного періоду.

Програмний комплекс виявлення початку атаки має гнучкі налаштування, що дозволяють задати чутливість до можливої атаки (лістинг 1). Конфігураційні дані виділені в окремий php-файл, що дає додаткові можливості як з точки зору зручності, так і з точки зору безпеки. Чутливість варіюється за допомогою корекції границі, а також порушенням границі

відразу в декількох періодах різного розміру. Наприклад, при порушенні границі на хвилинних інтервалах засіб може тільки сповістити зацікавлених осіб про збільшення активності. У разі порушення границі також на п'ятихвилинному інтервалі відбувається повне включення механізму захисту.

Лістинг 1 - Фрагмент конфігураційного файлу

```
//час періоду мережевої активності в секундах, 86400 добу, 604800 тиждень
    $Loop = 40400;
//період для дослідження в секундах
    $User_per = 300;
//кількість періодів для аналізу
    $Count_user_per = 100;
//період, який необхідно відступити від початку атаки для позначки
//легітивного трафіку
    $Safe_per = 600;
//число в процентах, на яке благополучний трафік повинен відрізнятися від
//шкідливого
    $Pogresnost = 10.
```

У разі виявлення початку атаки виконуються наступні дії:

1. Розсилка повідомлень. В автоматичному режимі відбувається розсилка повідомлень електронною поштою.

2. Виконання скриптів. Запускаються скрипти або сторонні програми, підготовлені для виконання системним адміністратором. Це можуть бути як скрипти, що включають додаткові рівні кешування або ж відключають модулі web-ресурсу, що генерують підвищене навантаження, так і системні скрипти та програми.

3. Активація засобів фільтрації трафіка.

Засіб фільтрації трафіка. На підставі розробленого алгоритму засіб фільтрації трафіку проводить первинну кластеризацію. В результаті первинної кластеризації в базі даних створюються дві таблиці, що характеризують шкідливий і легітивний трафік. Отримані таблиці використовуються в якості навчальних вибірок при класифікації запитів, що надходять. В процесі роботи таблиці уточнюються і доповнюються.

Дані, які містяться в таблиці, що характеризує шкідливий трафік, використовуються для блокування трафіку. У розробляємому програмному засобі передбачена можливість вилучення з таблиці, відповідно шкідливому трафіку, клієнтських IP адрес і створення на їх основі заборонних правил. Крім цього, на підставі даних про шкідливий трафік можливо реалізувати

додаткові механізми захисту. Наприклад, при надходженні шкідливих запитів до конкретної сторінки можна в автоматичному режимі тимчасово заблокувати цю сторінку або ж підмінити її статичної або кеш-версією. В цьому випадку шкідливий трафік, який був некоректно класифікований і не був заблокований на попередньому рівні, завдасть меншої шкоди.

Блокування шкідливих запитів. Для блокування шкідливих запитів передбачено два варіанти. В першому варіанті блокування здійснюється за допомогою створення відповідних забороняючих правил для iptables. Другий варіант буде актуальний якщо засіб функціонує у вузьких рамках віртуального хостингу, в цьому випадку блокування шкідливого трафіку здійснюється за допомогою заборонних правил, зазначених у файлі .htaccess (Лістинг 2).

В обох випадках блокування трафіку здійснюється повністю в автоматичному режимі. Також передбачений механізм експорту даних про шкідливий трафік для блокування його в різних програмних файерволах або ж на вищих мережевих вузлах.

Лістинг 2 - Приклад блокування IP-адрес у файлі .htaccess

```
order allow, deny
deny from 192.168.0.1
deny from 192.168.0.2
allow from all
```

Архітектура програмного комплексу представлена на рис. 1. Взаємодія модулів програмного комплексу, один з одним і з WEB-сервером, відбувається за наступною схемою:

1. В результаті обробки запитів, що приходять до WEB-сервера з мережі інтернет, в журнал WEB-сервера додаються відповідні події.

2. Модуль завантаження даних з заданим інтервалом часу зчитує нові дані з журналу і завантажує їх в базу даних.

3. Модуль виявлення початку атаки, аналізує дані про запити, що містяться в базі даних. У разі виявлення початку атаки, цей модуль створює в базі даних дві порожні таблиці. Одну для легітимних запитів, другу для шкідливих.

4. Модуль виявлення шкідливого трафіку відстежує появу і стан зазначених вище таблиць БД. Якщо таблиці незаповнені, модуль проводить кластеризацию і первинне заповнення таблиць. Якщо в таблицях вже є дані, модуль аналізує запити, шо надійшли на предмет приналежності до груп легітивних або шкідливих запитів, і додає дані про запит в відповідну

таблицю.

5. Модуль блокування запиту отримує список IP-адрес з таблиці, що містить шкідливі запити і вносить їх в «чорні списки» брандмауера або передає для блокування на вищестоящий мережевий сегмент.

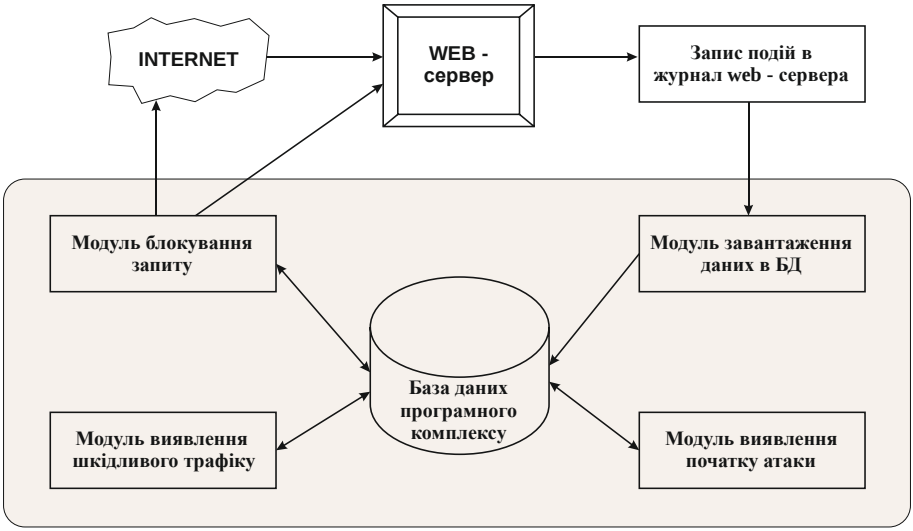

Рисунок 1 - Архітектура програмного комплексу

Розроблений програмний засіб повністю відповідає поставленим завданням. Основні риси створеного програмного комплексу для виявлення і протидії DDoS-атакам це кроссплатформенність, універсальність і масштабованість.

Програмний комплекс може застосовуватися в якості засобу забезпечення безпеки так званої «останньої милі». Основною спеціалізацією комплексу є забезпечення безпеки web-серверів від DDoS-атак типу http-flood. Програмний комплекс підтримує різні операційні системи, він може бути використаний з більшістю сучасних web-серверів. При цьому інсталяція комплексу може здійснюватися як в рамках фізичного сервера, так і в рамках віртуального хостингу.

Універсальність програмного комплексу виявлення і протидії DDoS-атакам полягає в можливості його використання не тільки для виявлення http-flood'a, а й інших DDoS - атак різних типів. При незначних змінах, що не

відносяться до основного модуля, програмний засіб може аналізувати різні дані, що містяться в log-файлах різних мережевих сервісах, або ж використовувати дані, отримані від мережевих локаторів.

У даній реалізації весь програмний комплекс, що складається з трьох модулів, розміщується на кінцевому мережевому ресурсі. В разі необхідності, модулі програми можуть бути розміщені в різних місцях мережі. Так, наприклад, на кінцевому сервері може знаходитися тільки засіб завантаження даних. Засоби виявлення атаки і фільтрації трафіку можуть бути встановлені на окремому сервері, недоступному для атаки з зовнішньої мережі. При такій установці програмний засіб зможе нормально функціонувати і проводити класифікацію трафіку навіть в випадку відмови атакуємого сервера.

Можливий варіант інсталяції, коли на вузлі, безпеку якого потрібно підтримувати, взагалі не встановлено ніяких модулів програмного засобу. У цьому випадку дані для аналізу можуть бути отримані від мережевих локаторів або вищестоящих маршрутизаторів. Блокування трафіку може бути здійснена на вищому вузлі.

Також програмний комплекс підтримує мультиінсталляцію при одночасному запуску декількох однойменних модулів. Так наприклад, дані для аналізу можуть надходити в базу даних з декількох джерел. Дані про шкідливий трафік можуть бути передані для блокування на різні рівні.

Перелік посилань

1. Бабаш, А.В. Криптографические методы защиты информации: учебник для студ. вузов / А. В. Бабаш, Е. К. Баранова. - М. : КНОРУС, 2016. - 190 с.

2. Батурин, Ю.М. Компьютерная преступность и компьютерная безопасность / Ю.М. Батурин, А.М. Жодзинский. – М.: Юридическая литература, 2006. – 160 с.

3. Борисов, М.А. Основы программно-аппаратной защиты информации: учеб. пособие для вузов / М. А. Борисов, И. В. Заводцев, И. В. Чижов. - 4-е изд., перераб. и доп. - М. : ЛЕНАНД, 2016. - 416 с.

4. Васильева, И.Н. Криптографические методы защиты информации : учебник и практикум для академ. бакалавриата / И. Н. Васильева. - Санкт-Петерб. гос. эконом. ун-т . - М. : Юрайт, 2017. - 349 с.

Огляд моделей захисту інформації в інформаційних системах

Савчук С.О.
Науковий керівник – к.т.н.,доц. Тітова В.Ю.
Хмельницький національний університет

В сучасному суспільстві комп'ютерні системи активно впроваджуються у фінансові, юридичні, промислові, торгові та соціальні галузі. У зв'язку з цим швидко зростає інтерес до проблем збереження та захисту інформації.

Тривалий час методи захисту інформації розроблялися тільки державними органами, а їхнє впровадження розглядалося як виняткове право певної держави [1]. Проте в останні роки збільшилися спроби несанкціонованого доступу до конфіденційної інформації, а проблеми захисту інформації виявилися в центрі уваги багатьох вчених і спеціалістів різних країн.

Згідно з оглядами міжнародних агентств з інформаційної безпеки, можна констатувати таке [2-3]:

- метою створення шкідливих програм і проведення атак стає, крім отримання грошового прибутку, крадіжка і подальше використання будь-якої можливої інформації
- з'являється новий клас шкідливих програм, націлений як на крадіжку персональної інформації користувачів, так і на тотальну крадіжку всіх інших даних.

А тому однією з актуальних на сьогоднішній день задач є вирішення питань ефективного захисту інформації, як від зовнішніх, так і від внутрішніх загроз, за рахунок створення та впровадження систем захисту інформації в автоматизованих системах підприємств, установ та організацій [2-3], що, серед іншого, потребує формалізації задачі захисту інформації для її наступної реалізації програмними та іншими засобами.

Відносно інформаційних систем застосовують наступні категорії безпеки [2-3]:

- надійність - гарантія того, що система працює в нормальному та позаштатному режимах так, як заплановано;
- точність - гарантія точного та повного виконання всіх команд;
- контроль доступу - гарантія того, що різні групи осіб мають різний доступ до інформаційних об'єктів, і ці обмеження доступу постійно виконуються;
- контрольованість - гарантія того, що в будь-який момент може бути зроблена повноцінна перевірка будь-якого компонента програмного комплексу;
- контроль ідентифікації - гарантія того, що клієнт, підключений у цей момент до системи, є саме тим, за кого себе видає;
– стійкість до навмисних збоїв - гарантія того, що при навмисному внесенні помилок у межах заздалегідь обговорених норм система буде

працювати так, як обговорено заздалегідь.

На основі зазначених категорій було розроблено кілька моделей інформаційної безпеки інформаційних систем.

Однією з перших моделей була модель Біба (рис.1) [2-3]. Відповідно до неї всі суб'єкти та об'єкти попередньо поділяються на декілька рівнів доступу, а потім на їх взаємодії накладаються наступні обмеження: 1) суб'єкт не може викликати на виконання суб'єкти з більш низьким рівнем доступу; 2) суб'єкт не може модифікувати об'єкти з більш високим рівнем доступу. Фактично, ця модель дуже нагадує обмеження, введені в захищеному режимі мікропроцесорів Intel 80386+ щодо рівнів привілеїв.

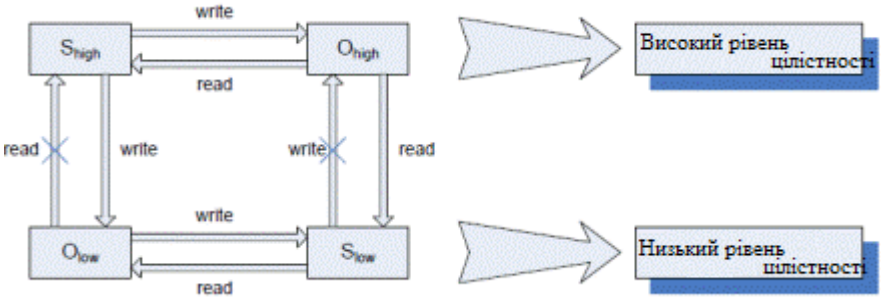

Рисунок 1 – Модель інформаційної безпеки Біба

Модель Гогена-Мезигера заснована на теорії автоматів [2-3]. Відповідно до неї система може при кожній дії переходити тільки з одного дозволеного стану в декілька інших. Суб'єкти та об'єкти в даній моделі захисту розбиваються на групи - домени і перехід системи з одного стану в інший виконується тільки відповідно до так званої таблиці дозволів, у якій зазначено, які операції може виконувати суб'єкт, скажімо, з домена C над об'єктом з домена D. У даній моделі при переході системи з одного дозволеного стану в інший використовуються транзакції, що забезпечує загальну цілісність системи.

Сазерлендська модель захисту наголошує на взаємодії суб'єктів та потоків інформації [2-3]. Так само як і у попередній моделі, тут використовується машина станів з множиною дозволених комбінацій станів і деяким набором початкових позицій. У даній моделі досліджується поведінка множинних композицій функцій переходу з одного стану в інший.

Важливу роль у теорії захисту інформації відіграє модель захисту Кларка-Уілсона (рис.2) [2-3]. Засновано дану модель на використанні транзакцій і ретельному оформленні прав доступу суб'єктів до об'єктів. В даній моделі вперше досліджена захищеність третьої сторони в даній проблемі - сторони, що підтримує всю систему безпеки. Цю роль в інформаційних системах відіграє програма-супервізор. Крім того, у моделі Кларка-Уілсона транзакції вперше були побудовані за методом верифікації,

тобто ідентифікація суб'єкта здійснюється не тільки перед виконанням команди від нього, але й повторно після виконання. Це дозволило зняти проблему підміни автора в момент між його ідентифікацією й самою командою.

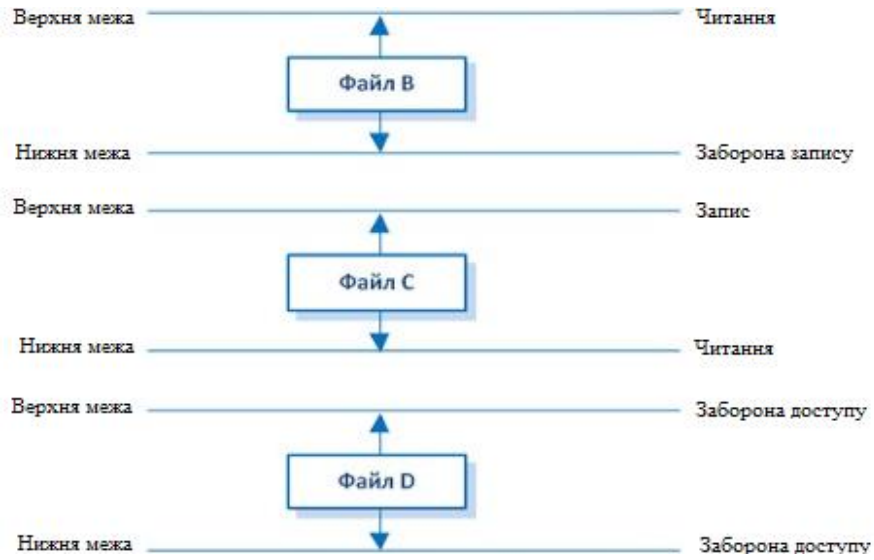

Рисунок 2 – Модель інформаційної безпеки Кларка-Уілсона.

Враховуючи переваги та недоліки наведених вище моделей, можна зробити висновки, що модель Кларка-Уілсона є однією із найкращих відносно підтримки цілісності інформаційних систем, а тому саме її доцільно взяти за основу для системи захисту інформації, описаній у [4].

Перелік посилань

1. Про державну таємницю [Електрон. ресурс] : закон України// Відомості Верховної Ради (ВВР). – 1994. – №16, ст. 93. – Режим доступу: http://zakon2.rada.gov.ua/laws/show/3855-12

2. Баранова Е.К. Информационная безопасность и защита информации: учеб. пособие / Е.К. Баранова, А.В. Бабаш. – М.: РИОР: ИНФРА-М, 2017. – 322 с. – ISBN: 978-5-369-01450-9

3. Бирюков А.А. Информационная безопасность. Защита и нападение/ А.А. Бирюков. – М. : ДМК-Пресс, 2017. – 434 с. – ISBN: 978-5-97060-435-9

4. Тітова В.Ю. Концептуальна модель системи захисту інформації в сучасних компютерних системах/ В.Ю. Тітова, С.О. Савчук, В.Ю. Черниш// Вісник ХНУ. - Хмельницький: ХНУ, 2019. - №3. - с. 164-168.

Формування і розмітка навчальної вибірки для проектування технічних систем за допомогою тематичної сегментації текстів

Сокальський В.Р.
Науковий керівник: ктн. доц. Огнєвий О.В.
Хмельницький національний університет

При побудові експериментального датасета для текстів наукової прози має місце принципове обмеження на доступний його обсяг. Для подолання цього обмеження було обрано базуватися на принципах відбору зразків текстів, і використовували прийом стратифікації масиву текстів, що належать до жанру наукової прози, досить типовий для дослідження корпусів текстів. Специфіка даної роботи полягала в тому, що стратифікація проводилася відповідно до категорій ознак, які були визначені в розділі 2: мова оригіналу, текстові ознаки, структура тексту. Конкретні набори і діапазони значень цих ознак вибиралися таким чином, щоб покривають потреби вітчизняних проектувальників при аналізі науково-технічної літератури [1].

Тексти відбиралися таким чином, щоб кожна страта в датасете була представлена кількома зразками. Зауважимо, що в магістерскій роботі не ставилося завдання повномасштабного статистичного дослідження, тому з точки зору репрезентативності такий підхід видається цілком правомірним.

Таким чином, для проведення магістерского дослідження були відібрані тексти з 15 джерел на різних мовах (українська, англійська, французька) зі складною граматикою і різним ступенем мінливості. Тематика відібраних текстів відображає аспекти технології в широкому діапазоні предметних областей.

Крім того, для порівняння з існуючими методами ТС і для виявлення можливої специфіки ТС набір даних включає в себе штучно створені медичні тексти, що представляють собою конкатенацію реальних медичних висновків. Тут під конкатенацією розуміється» склеювання " фрагментів зв'язкових текстів з різних джерел в єдиний текст. Зауважимо, що такий прийом широко використовується при формуванні експериментальних датастів у різних завданнях сегментації текстів.

Для дослідження відібрано безперервні текстові фрагменти, в яких представлені всі типи зв'язності на підставі класифікації. Такий підхід дозволяє в чистому вигляді досліджувати мовні особливості параметрів сегментації.

Всього було відібрано 15 текстів із загальним обсягом 146 000 слів, у тому числі 10 текстів технічної проблематики, 3 Тексти медичної проблематики, 2 тексти з IT-проблематики. 2 Тексти були французькою, 6-англійською, 7-українскою мовами. Для трьох текстів (2 – французькою мовою, технічної проблематики, 1 – англійською мовою, медичної проблематики) в датасет включений також їх професійний переклад на українську мову [2].

Лінгвістичні характеристики деяких відібраних текстів представлені в таблиці 1 відповідно.

Завдання ТС наукової прози, можна розглядати як задачу багатокласової класифікації без учителя – віднесення конкретного абзацу тексту до того чи іншого топіку (компоненту структури тексту, закладеної автором). Потенційно досяжна помилка сегментації визначається поєднанням обсягу класів і розміру навчальної вибірки, що містить ці класи.

Зокрема, в роботі експериментально показано, що найменша помилка класифікації досягається на класах, які займають не більше 10% вибірки [3].

Таблиця 1 – Лінгвістичні характеристики деяких відібраних текстів

Означення	Мова	Тематика	Розмір тексту			Розмір абзацу (слів)	
			слів	абзаців	термінів	min	max
T1	Франц.	Техн.	20887	108	5163	13	764
T2	Франц.	Техн.	15160	78	3922	13	607
T3	Англ.	IT	13252	102	4825	30	290
T4	Англ.	IT	10710	123	4178	35	173
T5	Англ.	Техн.	10243	86	2693	18	198
T6	Англ.	Мед.	7338	131	5451	17	233
T7	Укр.	Мед.	7435	131	5153	12	161
T8	Укр.	Мед.	4096	100	3249	8	173

Як показує практика, характерний розмір топіка $topic$, на який потрібно проводити сегментацію в задачах ТС для проектувальників технічних систем, не перевищує $topic = 10$ абзаців. Для оцінки необхідного обсягу вибірки використано вираз, справедливий для малої кількості подій $(n \le 30)$:

$$\varepsilon = \sqrt{\frac{p^*(1-p^*)}{n}}\sqrt{1-\frac{n}{N}}$$

де p^* – відносна частота появ події А в серії з n випробувань (точкова оцінка ймовірності p появи події А в окремому випробуванні). Характерні значення для ТС наукової прози представлені в таблиці 1. Як випливає з таблиці 2, для розміру топіка наукової прози в 10 абзаців необхідний обсяг вибірки становить 100 абзаців, що цілком відповідає відібраних текстів. Крім того, в наступних експериментах було вияснено, що значення $topic = 10$ абзаців є завищеними, і реальне значення становить $topic = 3-5$ абзаців,

тобто використовуваний в експериментах обсяг вибірок можна вважати достатнім.

Розмітка навчальної вибірки. Поділ тексту за формальним складовим (абзаців або розділів) не дозволяє однозначно виділити підтеми документа: межі абзацу та тематичних блоків можуть не співпадати, поділ на абзаци залежить від типу і призначення документа (наприклад, текст новин і художній), великі абзаци можуть містити в собі декілька підтем. Межі абзаців часто суб'єктивні, проте позиції зміни топіків в тексті в переважній більшості випадків корелюють з кордонами абзаців [4].

Таблиця 2 – Розрахунок необхідного обсягу вибірки

Величина	Змістовна інтерпретація для ТС наукової прози	Значення
N - генеральна сукупність	Середнє число абзаців в тексті	100
n - кількість подій A	Подія A - зміна топіка після поточного абзацу	10
$p*$	відносна частота появ подій A	0,1
Рівень значущості		0,1
довірчий інтервал		4,95%

Для з'ясування цього питання стосовно задачі ТС наукової прози були проведені дослідження відібраних текстів. Всі вони мають авторську розмітку, тобто заголовки і підзаголовки різного рівня вкладеності (таблиці 3).

Таблиця 3 – Кількість рівнів авторського структурування

Позначення тексту	Кількість рівнів авторського структурування
T1	1
T2	1
T3	3
T4	3
T5	2
T6	3
T7	3
T8	1

Найбільш поширеним способом встановлення достовірності ТС являється зіставлення отриманих результатів з діленням експерта. У цьому

зв'язку в роботі використана експертна процедура для формування опорної розмітки текстів, згідно з якою еталонна межа теми фіксується у тому випадку, якщо було вказано не менше 2/3 анотаторів. У дослідженні взяли участь в цілому 16 анотаторів з вищою професійною освітою, з яких випадковим чином були обрані підгрупи для конкретних текстів. Тексти Т1 і Т2 і їх професійні переклади на українську мову, а також текст Т5 і його професійний переклад на українську мову оцінювалися анотаторами окремо, але, так як ці тексти ідентичні за змістом з точністю до параграфа, результати оцінок об'єднувалися. Для оцінки узгодженості результатів анотації використаний показник Fleiss' kappa, який являє собою статистичну міру згоди серед фіксованої групи експертів, які аранжують деяку сукупність предметів.

У таблиці 4 представлені приклади анотування деяких текстів наукової прози, що використовуються в роботі.

Таблиця 4 – Результати анотування деяких текстових джерел

Текст	Число анотаторів	Межа топіка (№ абзацу)та її абсолютне відхилення	Fleiss' kappa	Авторська розмітка	
				Всього меж	Співпало з експертною оцінкою
Т1, частина 1	4	21 ± 0, 27 ± 1, 49 ± 0, 51 ± 1, 57 ± 2, 67 ± 1,	0.666	5	4
Т1, частина 2	3	71 ± 0, 73 ± 0, 78 ± 0, 84 ± 1, 97 ± 0, 108 ± 0,	0.555	5	5
Т2, частина 1	5	4 ± 1, 13 ± 2, 19 ± 2, 23 ± 1, 37 ± 2, 43 ± 1,	0.500	3	2
Т2, частина 2	3	54 ± 1, 62 ± 1, 66 ± 1, 79 ± 1, 86 ± 1, 91 ± 1,	0.700	5	5
Т5 і його професійний переклад	6	10 ± 2, 21 ± 2, 25 ± 1, 31 ± 1, 54 ± 0, 57 ± 1, 65 ± 0, 71 ± 1, 75 ± 1, 90 ± 1, 95 ± 1, 109 ± 1, 119 ± 1, 130 ± 0, 133 ± 0,	0.588	35	14

Як видно з таблиці 4, для використовуваних в дослідженні текстів середнє значення показника Fleiss' kappa = 0.68, при цьому максимальне відхилення граничного положення становило 2 абзацу (для текстів Т1 і Т2 з 1

рівнем заголовков). Таким чином, показано, що, незважаючи на високий рівень суб'єктивності завдання, узгодженість думок експертів досить висока і не перевищує ±1 абзац, а для текстів з рівнями заголовків 2 і 3 експертні та авторські розмітки збігаються.

Таким чином, у дослідженні як термінальну одиницю для поділу тексту на топіки обрано абзац, а не окрему пропозицію. З іншого боку, деякі автори навіть у науковій прозі формують надмірно короткі абзаци, обсяг яких принципово недостатній для встановлення належності до топіку. Такі абзаци при аналізі приєднувалися до наступних за ним абзаців [5].

Крім того, як встановлено в дослідженнях в наукових текстах між темами може існувати перехідна зона величиною 1-2 параграфа, які з великою ймовірністю є або заголовками, або авторськими відходами і не несуть самостійної думки рівня топіка. Такі абзаци при аналізі приєднувалися до наступних за ним абзаців.

Перелік посилань

1. Автоматическая обработка текстов на естественном языке и анализ данных: учеб. пособие/ Большакова Е.И., Воронцов К.В., Ефремова Н.Э., Клышин-ский Э.С., Лукашевич Н.В., Сапин А.С. — М.: Изд-во НИУ ВШЭ, 2017. — 269 с. ISBN 978–5–9909752-1-7

2. Агаркова Н.В.(Добренко), Артемова Г.О., Гусарова Н.Ф. Система поддержки принятия проектных решений для документирования научно-технической информации // Научно-технический вестник инфор-мационных технологий, механики и оптики - 2012. - № 1(77). - С. 128-134

3. Айсина Р.М. Обзор средств визуализации тематических моделей коллекций текстовых документов // Машинное обучение и анализ данных (http://jmlda.org). –2015. –Т. 1, № 11. – С.1584–1618

4. Алиев Т.И. Основы проектирования систем. – СПб: Университет ИТМО, 2015. – 120 с.

5. Большакова Е.И. и др. Автоматическая обработка текстов на естественном языке и компьютерная лингвистика. М.: МИЭМ. 2011.

Шифрування текстової інформації у зображення методом стеганографії

Федюра О.С.
Науковий керівник – викл. Дмитрієва М.В.
Ізмаїльський державний гуманітарний університет

Напевно у кожної людини в житті трапляється ситуація, коли необхідно за допомогою телефону написати певну конфіденційну інформацію так, щоб ніхто з оточуючих не зміг її прочитати і використати у власних цілях.

Базові принципи і алгоритми захисту інформації при передачі по мережі були описані ще в 70-ті роки і до цих пір актуальні. Для боротьби з перехопленням інформації користувачів створюються і удосконалюються

протоколи шифрування - набори криптографічних алгоритмів, що описують перетворення даних [1]. Ці протоколи охороняють дві базових властивості інформації - конфіденційність і цілісність. Розробкою проблемно-орієнтованих систем шифрування займається сучасна прикладна наука - криптографія.

Метою нашого дослідження стала організація безпечної передачі інформації шляхом шифрування її у зображення.

В історичній та математичній літературі дана тема досить добре опрацьована, оскільки вивчення історії шифрів та процесу шифрування необхідно для черпання ідей для її розвитку сьогодні. Матеріалом для написання роботи послужила, в першу чергу, книга А.П. Алфьорова, А.Ю. Зубова А.С. Кузьміна і А.В. Черьомушкіна «Основи криптографії» [2], в якій подано докладний історичний нарис раннього розвитку криптографії.

Криптографія - це наука про математичні методи забезпечення конфіденційності, цілісності й автентичності інформації. Вона розвинулась з практичної потреби передавати важливі відомості найнадійнішим чином, зміст яких, як люди сподівались, не зможе ніхто прочитати, окрім адресатів [3]. Існують такі способи прихованої передачі даних як:

- створити абсолютно надійний, недоступний для інших канал зв'язку між абонентами;
- використовувати загальнодоступний канал зв'язку, але приховати сам факт передачі інформації;
- використовувати загальнодоступний канал зв'язку, але передавати по ньому інформацію в перетвореному вигляді, щоб відновити її міг тільки адресат.

Один з найпростіших способів приховати інформацію - замінити кожну літеру алфавіту на інші літери того ж алфавіту. Зрозуміло, що так званий шифр Цезаря, де кожна літера листа замінювалась на літеру, що відстає за алфавітом на три позиції далі, був не дуже надійною криптосистемою. Так що щоб розшифрувати такий текст, досить знати алгоритм заміни.

Сьогодні дані шифруються за допомогою криптографічних ключів. Інформація шифрується до відправлення і розшифровується одержувачем. Таким чином, при передачі дані знаходяться в безпеці. Залежно від природи ключів шифрування може ділитися на 2 категорії: симетричне й асиметричне.

Симетричне шифрування: дані шифруються і розшифровуються за допомогою одного криптографічного ключа. Це означає, що ключ, який використовується для шифрування, використовується і для розшифровки.

Асиметричне шифрування: це досить новий метод. У ньому використовуються два різних ключі - один для шифрування, другий для розшифровки. Один ключ називається публічним, другий секретним [4].

Криптографічні методи стали широко використовуватися приватними особами в електронних комерційних операціях, телекомунікації та багатьох інших середовищах.

Займаючись вивченням мов програмування, нами була створена власна програма яка шифрує дані стеганографічним методом.

Стеганографія - тайнопис, при якому повідомлення, закодоване таким чином, що не виглядає як повідомлення - на відміну від криптографії. Таким чином непосвячена людина принципово не може розшифрувати повідомлення - бо не знає про факт його існування.

На сьогоднішній день існує лише один спосіб шифрування тексту у зображення суть якого полягає у наступному: обирається будь-яка готова картинка. Будемо розмислювати у форматі RGB. Кожен піксель складається з трьох каналів: R - red (червоний), G - green (зелений), B - blue (синій). Кожен канал містить значення від 0 до 255, тобто 1 байт (3 байта на піксел). 1 байт = 8 біт. Наприклад 01101101 - 8 біт або 1 байт. Змінюючи останній (молодший) біт (рис. 1), ми змінюємо значення усього байту (а також канал кольору) лише на 1/256.

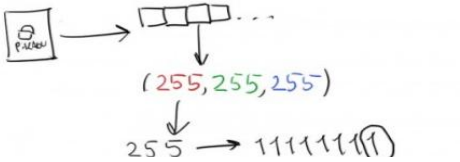

Рисунок 1- Зображення молодшого біту каналу кольору.

Маючи на увазі, що текст це послідовність біт (рис. 2), ми можемо записати кожен біт тексту у молодший біт значення каналу кольору (рис. 3).

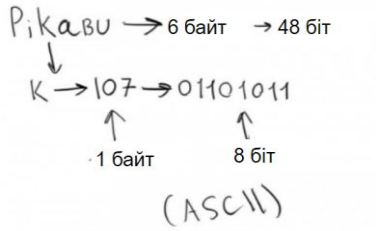

Рисунок 2- Текст - послідовність біт.

Так як це значення змінюється лише на 1/256, то людське око не зможе помітити різницю. Таким чином ми зможемо зберігати “Ширина * Висота (зображення) * 3 каналу кольору / 8 біт” символів у зображенні.

Повертаючись до розглядання нашого проекту слід звернути увагу на його варіативність. Тобто даний метод шифрування, з певними змінами, можна використовувати в односторонньому напрямку, а саме без можливості розшифрування. Такі типи криптографічних систем використовуються для збереження паролю користувачів у базах даних, на випадок отримання несанкціонованого доступу злочинцями, вони отримають лише зашифрований пароль, можливість відновлення оригіналу якого не є

можливим. А реалізується даний підхід у нашому проекті наступним чином: достатньо лише використовувати символи не юнікод таблиці, для створення словника, а по одному екземпляру кожного символу тексту. Тим самим для кожного тексту словник буде унікальним, і для розшифрування котрого необхідно знати безпосередньо оригінал тексту, для створення словнику для розшифрування, що не є можливим маючи лише зображення.

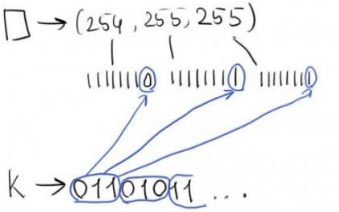

Рисунок 3- Заміна молодшого біта каналу кольору на біт тексту.

Нами було проведено апробацію в рамках нашого факультету. Вдало пройшла передача зашифрованого документу з деканату на кафедру. Без ключа не вдалося отримати початковий вигляд текстової інформації, але після введення правильного ключа вдалося розшифрувати отриманий файл.

Огляд науково-публіцистичної літератури з зазначеного питання, аналіз досліджень в області криптографії дають нам впевненості сказати, що на даний момент спосіб шифрування, описаний вище, є єдиним прикладом шифрування тексту у зображення, і тим самим ми можемо переконатися, що розроблений нами проект є унікальним.

Поставлені нами завдання на початку проекту були виконані в повному обсязі, а саме:

- розглянуто основні напрямки розвитку шифрування даних;
- наведені приклади використання методів та принципів шифрування даних;
- розробивши власний алгоритм, нам вдалося провести апробацію написаної нами програми для передачі даних у рамках факультету.

Вважаємо доцільним використання даного методу для безпечної передачі текстової інформації мережею Інтернет.

Перелік посилань

1. Музагафаров А. Шированный миф: азы криптографии. Просто, понятно и увлекательно. / Артем Музагафаров; 2008.- 34с.

2.Алферов А.П., Зубов А.Ю., Кузьмин А.С., Черемушкин А.В. Основы криптографии. –М.: Гелиос АРВ, 2001. –480 с.

3.Ємець В., Мельник А., Попович Р. Сучасна криптографія. Основні поняття. Львів. БаК., 2003. - 144с.

4.Венбо М. Современная криптография. Теория и практика / Мао Венбо. – Киев: Издательский дом "Вильяс", 2005. – 755 с..

Метод виявлення мережевих атак в комп'ютеризованих системах управління

Холявка Є.П.
Науковий керівник – професор Лавров Е.А.
Сумський державний університет, Суми, Україна

Під мережевий атакою розуміємо вплив на програмні компоненти цільової системи за допомогою програмних засобів. Метою атаки є отримання даних або здійснення проникнення. За характером впливу мережеві атаки поділяють на: пасивні(не відбувається прямого впливу на систему), активні(відбувається безпосередній вплив на функціонування системи). За цілями впливу мережеві атаки виділяють три типи: атаки розвідки, атаки отримання доступу, атаки отримання доступу(Denial of Service, DoS).

Методи виявлення зловживань базуються на порівняння поточного стану системи з образом, званим сигнатурою. Основний недолік методів виявлення зловживань – неможливість опису всіх можливих атак, до того ж, навіть невелика зміна в структурі атаки призводить до неможливості виявлення даними методами.

Самоорганізуючі карти Кохонена (Self-Organizing Map, SOM) – це різновид нейронних мереж, які навчаються на основі методу навчання без учителя. При навчанні без учителя результат навчання залежить тільки від структури вхідних даних, навчальна множина складається лише з вхідних векторів і перевірки з будь-якими еталонними значеннями не проводиться.

Самоорганізуючі карти вирішують задачу кластеризації і візуалізації вхідних даних, що дозволяє визначити наявність або ж відсутність взаємозв'язку в даних.

Самоорганізуючі карти представляють собою безліч нейронів, кількість яких збігається з кількістю кластерів. Нейрони є деякий вектор-стовпець виду:

$$w_j = [w_{j1}, w_{j2}, \dots w_{jN}]^T$$

Зручним інструментом вирішення завдання задачі кластеризації проблемних ситуацій наявності кібератак може бути програмне забезпечення WEKA. Досліджено 4 алгоритими формування навчальної вибірки. Приведені комп'ютерні експеременти, дозволяють визначити ефективність алгоритмів визначення множини необхідних атрибутів та виконання ранжування у порядку зростання ефективності:

1. OneR;
2. Correlation;
3. Information Gain;
4. Gain Ratio.

Система GPS моніторингу вантажного транспорту та снігоприбиральної техніки

Хоменко І.С. Сидорчук В.О.
Науковий керівник: к.т.н., доцент кафедри ІПЗ Сугоняк І.І.
Житомирський державний технологічний університет

Актуальність теми. Технології стрімко розвиваються, і сьогодні ні для кого не є проблемою відслідкувати потрібний тролейбус, трамвай або маршрутне таксі за допомогою свого смартфону. Технології GPS трекінгу широко використовуються в соціальних сферах з метою покращення сервісу, також однією з переваг слідкування є моніторинг ресурсів. Автоматизована система слідкування за снігоприбиральною технікою надасть можливість оптимізувати процес прибирання снігу з наших вулиць, а також допоможе відслідковувати ресурси які були затрачені.

Однією з основних задач системи є прокладання оптимального маршруту, що зменшить не цільове використання палива, а також дає можливість швидше і ефективніше прибрати найбільш завантажені вулиці нашого міста.

Другою, але не менш важливою проблемою є не цільове використання ресурсів. Система вирішила проблему автоматичним розрахунком потрібної кількості палива та хімікатів, що зробить не можливим використання ресурсів недобросовісними працівниками в свою користь.

Важливим фактором системи є трьохсторонній зв'язок, який дозволяє диспетчеру, системі та водію транспортного засобу завжди залишатися на зв’язку. Якщо виникне не передбачувана ситуація, водій зможе подати сигнал, система автоматично призначить на цей маршрут іншу техніку, а також сповістить диспетчера який зможе вислати на місце ремонтну бригаду.

До цього часу всі три задачі не були вирішені, вирішені частково або окремо одна від одної, що не давало повної автоматизації та погіршувало процес прибирання доріг. Це в свою чергу створювало проблеми пересування вулицями міста.

Головним об’єктом дослідження був процес побудови оптимальних маршрутів для снігоприбиральної техніки. Прокладені маршрути оптимізують процес прибирання. При прокладанні маршруту враховується декілька факторів:

- засніженість вулиць,
- не пересікання з іншою технікою,
- мінімізація повторного проїзду по тому самому маршруті.

Це допоможе скоротити витрати на пальне, та зменшити час за який всі вулиці будуть прибрані.

В досліджені було реалізоване слідкування за снігоприбиральною технікою за допомогою технологій GPS та ГЛОНАС. Данні надходять з GPS трекеру встановленого на транспорті, або смартфону водія.

На смартфон водія було встановлено розроблений нами android додаток, який буде збирати та відправляти на сервер інформацію про техніку.

Також за допомоги додатку водій має прямий зв'язок с диспетчером та зможе при необхідності викликати допомогу.

Данні прийняті від додатку зберігаються та обробляются на сервері системи. Оператор, за допомогою веб додатку, може:

- переглянути місце знаходження всіх одиниць техніки
- переглянути всі прокладені маршрути
- переглядати яка частина вулиць вже прибрана, а яка потребує призначення більшої кількості техніки.
- призначити додаткову техніку на маршрут
- моніторити ресурси техніки (пальне, хімікати для посипання, та вільне місце в кузові(в тому випадку якщо це саме збиральна техніка)
- комунікувати з водіями(попередити про затори, аварію, а також отримати повідомлення про несправність та відрядити ремонтну бригаду на місце пригоди).

Система складається з 3х основних елементів:

1.Android додатку, який збирає данні з датчиків техніки.

2.Бази даних, в нашому випадку MySQL. В базі зберігаються данні які ми отримаємо від техніки.

3.Веб додаток, який обробляє, структурує та виводить потрібну інформацію оператору.

Щоб зробити систему універсальною та не прив’язуватися до одного або декількох виробників датчиків та систем супутниково стеження, наш Android додаток обробляє та структурує отримані дані і відправляти на сервер в потрібному для системи вигляді.

Практичне значення одержаних результатів. Система дозволяє слідкувати за вантажною технікою, спец технікою а також легковими автомобілями які належать комунальним підприємствам. На основі отриманих результатів, комунальне підприємство може систематизувати процес прибирання снігу, та витрату ресурсів.

Перелік посилань

1. Д. Марка, К. МакГоуэн, Методология структурного анализа и проектирования. М.: МетаТехнология, 1993, 240 с.

2. GoogleMapsPlatform [Електроний ресурс]. Доступ за посиланням: https://developers.google.com/maps/documentation/?hl=ru [Дата звернення 20.04.2018]

Розробка алгоритму перетворення ЦВЗ для впровадження в цифрове зображення на основі використання математичного апарату модулярної арифметики для забезпечення цілісності ЦВЗ

Шепель А.В.
Науковий керівник: ктн. доц. Джулій В.М.
Хмельницький національний університет

Цифровий водяний знак (ЦВЗ) може виступати як послідовність чисел, символів або тексту, що містить знак охорони авторських прав, бінарного зображення з логотипом організації (якщо автором є юридична особа), QR – коду, який також може містити знак охорони авторських прав або посилання на сайт автора. Незалежно від того в якому вигляді виступає ЦВЗ, він може бути представлений модулярним кодом, що дозволяє при наявності деструктивного впливу на систему ЦВЗ можливість виявлення і корекції помилок [1].

Для побудови системи корекції спотворень в числовому представленні ЦВЗ був обраний спосіб, що грунтується на методі проекцій як найбільш обчислювально простий і ефективний.

Обчислимо A_{ij}, яке може бути отримано з числового значення A шляхом виключення з його подання цифр по модулях $p_i \; i \; p_j$ проекцією числового значення A по модулям $p_i \; i \; p_j$, причому має бути дотримана умова $i \neq j$. Далі необхідно провести обчислення всіх проекцій числового значення A за всіма основами, наявними в заданій числовій системі: $A_{12},...,A_{ij}$. Серед отриманих проекцій необхідно знайти таке числове значення, яке задовольняло б умові:

$$A_{ij} < \frac{P'}{p_{k+1} \cdots p_{k+n}}$$

значить, спотвореними є цифри α_i, α_j. Після того, як обчислені спотворені цифри в числовому поданні ЦВЗ, необхідно здійснити процедуру їх корекції відповідно до формули:

$$\alpha_i = \tilde{\alpha}_i + \left[\frac{p_i(1 + np_{k+1})}{p_{k+1} m_i} \right]$$

На рисунку 1 у вигляді блок схеми, представлений алгоритм перетворення ЦВЗ для вставки зображення-контейнер, який є об'єктом інтелектуальної власності [2].

Рисунок 1 – Алгоритм перетворення ЦВЗ для вбудовування в зображення

Вхідними даними для роботи алгоритму, представленого на рисунку 1, є ЦВЗ, незалежно від форми його подання. Завдання, що стоїть на початковому етапі виконання даного алгоритму, полягає у визначенні типу ЦВЗ.

Як було зазначено раніше, ЦВЗ може бути представлено послідовністю чисел або символів, містити текст або зображення. Після визначення типу ЦВЗ, з яким буде здійснюватися подальша обробка, необхідно привести початкове уявлення до виду, придатного для функціонування алгоритму [3].

Формується двійковий масив, групи елементів якого, в залежності від робочого діапазону обраної системи залишкових класів, перетворюються в блоки по t-біт. Після чого дані блоки піддаються перетворенню з двійкової системи в десяткову. На наступному етапі здійснюється процедура перетворення матриці, елементами якої є числа в десятковій системі числення в матрицю, елементами якої є цифри, представлені надмірною модулярним кодом (далі – НМК) [4].

У разі якщо вихідною формою ЦВЗ є текстові дані, символи цього вектора можуть бути представлені десятковими цифрами відповідно до заздалегідь визначеним кодуванням. У процесі перетворення обов'язковим критерієм входження символу в перетворену матрицю є його приналежність до робочого діапазону обраної системи підстав, тобто належність відповідного числового значення в десятковій системі числення діапазону дозволених значень системи підстав. Як і у випадку із зображенням, на наступному етапі проводиться процедура перетворення матриці, елементами якої є числа в десятковій системі числення в матрицю, елементами якої будуть вже цифри, які представлені НМК [5].

Відповідно до ідеї алгоритму, залишки по кожному з обраних модулів системи повинні зберігається в окремих областях зображення-контейнера, звідки випливає необхідність в поділі матриці із залишками, отриману на попередньому етапі, на n-матриць містять значення по кожному модулю. Виконання даної вимоги необхідно для забезпечення цілісності ЦВЗ.

У випадку, якщо зображення з вкладеним ЦВЗ піддається деструктивному впливу, інформація по кожному з підстав системи є відносно ізольованою один від одного, що дозволяє з більшою ймовірністю відновити втрачені дані.

Перетворення з позиційної системи числення (далі – ПСЧ) в надлишковому модулярному коді (НМК) буде проводитися по шести модулях, два з яких будуть надлишковими. Для оцінки коригувальних здібностей коду в розділі 1 даної роботи було використано поняття кодової відстані, що забезпечує відповідність між надмірністю кодування і здатністю до виявлення і виправлення помилки.

Робочий діапазон обраної системи дорівнює

$P = p_1 p_2 ... p_k = \prod_{i=1}^{k} p_i = 210,$ повний діапазон системи

$P' = p_1 p_2 ... p_k p_{k+1} ... p_{k+n} = \prod_{i=1}^{k+n} p_i = 30030,$ причому $d_{\min} = 3$, звідки випливає, що здатність коду дорівнює

$$\frac{M^T - M}{M} * 100\% = \frac{30030 - 210}{30030} * 100\% \approx 99.3\%$$

Таким чином, при наявності двох надлишкових основ можливо виявити будь-які одиночні і подвійні помилки, виявити потрійні помилки з ймовірністю 99,3% і гарантовано виправити всі поодинокі помилки.

При необхідності кількість надлишкових основ коду може бути збільшено, а, отже, і поліпшені коригувальні здатності коду.

Згідно з розробленим алгоритмом, матриця з елементами ЦВЗ, представленими НМК, поділяється на n – матриць (у розглянутому випадку на 6 матриць) для того, щоб залишки по кожному окремому модулю (основи) заносилися в окремий блок зображення, який відповідає певній основі системи. При такій організації, навіть при повному видаленні частини зображення, яка включає повністю один з блоків переднього плану, ЦВЗ буде гарантовано повністю відновлено.

Перелік посилань

1. Гонсалес, Р. Цифровая обработка изображений / Гонсалес Р., Вудс Р., – М: Техносфера, 2005. – 1072 с. ISBN 5-94836-028-8.

2. Шипилов, А. Авторские права в цифровую эпоху/ А. Шипилов // Ephoto. –2002. –№ 3 (13). – С.34-37.

3. Глумов, Н.И. Алгоритм поблочного встраивания стойких ЦВЗ в крупноформатные изображения / Н.И. Глумов, В.А. Митекин // Компьютерная оптика. – 2011. – Том 35, № 3. – С.368-372.

4. Глумов, Н.И. Алгоритм встраивания полухрупких цифровых водяных знаков для задач аутентификации изображений и скрытой передачи информации / Н.И. Глумов, В.А. Митекин // Компьютерная оптика. – 2011. – Том 35, № 2. –С.262-267.

5. Горбачев, В.Н. Методы цифровой стеганографии для защиты изобразительной информации / В.Н. Горбачев, Е.М. Кайнарова, А.Н. Кулик, И.К. Метелев // М.: Проблемы полиграфии и издательского дела. – 2011. – № 2. – С.32-49.

Наукове видання

«Інтелектуальний потенціал – 2019» - збірник наукових праць молодих науковців і студентів з нагоди 30-річчя кафедри кібербезпеки та комп'ютерних систем і мереж ХНУ/Колектив авторів – Хмельницький: ПВНЗ УЕП, 2019. – Ч.1: Комп'ютерні системи та кібербезпека. – 100 с.

Відповідальність за зміст текстів і якість редагування матеріалів покладена на авторів і наукових керівників.

Комп'ютерна верстка: Чешун В.М.
Дизайн: Муляр І.В.

Здано до складання 11.11.19. Підписано до друку 14.11.19. Формат 60x84/16. Папір друкарський. Тираж 50 прим. Умовних друкованих аркушів – 7,5.

Редакційний відділ ПВНЗ УЕП 29016, м. Хмельницький, вул. Львівське шосе, 51/2.

ББК 74.480.278
С.88