НПК МНІС ІП–2020

ЗБІРНИК НАУКОВИХ ПРАЦЬ МОЛОДИХ НАУКОВЦІВ І СТУДЕНТІВ

ЧАСТИНА 2

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

Західноукраїнський національний університет

Інтелектуальний потенціал - 2020

збірник наукових праць молодих науковців і студентів

сформовано за матеріалами
Всеукраїнської науково-практичної конференції
молодих науковців і студентів
«Інтелектуальний потенціал – 2020»

9-10 листопада 2020 р.

Частина 2

Хмельницький
2020

ББК 74.480.278
С.88

«Інтелектуальний потенціал – 2020» – збірник наукових праць молодих науковців і студентів / Колектив авторів – Хмельницький: ПВНЗ УЕП, 2020. – Частина 2. – 100 с.

Відповідальний редактор: *Желавська Н.В.*

Відповідальний за випуск: *Чешун В.М.*

Редакційна колегія:
Лєнков С.В. (ВІКНУ)
Яцків В.В. (ЗУНУ)
Якименко І.З. (ЗУНУ)
Желавський О.Б. (ПВНЗ УЕП)
Желавська Н.В. (ПВНЗ УЕП)
Кльоц Ю.П. (ХНУ)
Чешун В.М. (ХНУ)
Тімофєєва Л.В. (ХНУ)

© Університет економіки і підприємництва

ЗМІСТ

Оцінка ефективності методу прогнозування взаємоблокувань процесів в комп'ютерній системі

Андрощук О.С., Горчилов І.І., Нагребецький О.В.
Хмельницький національний університет

При експлуатації комп'ютерних систем (КС) досить часто виникає ситуація блокування процесів, що виконуються у них. Частковим випадком блокування задач є можливість їх взаємного блокування [1]. Взаємне блокування – ситуація в багатозадачному середовищі або системах керування базами даних (СКБД), при якій кілька процесів перебувають у стані нескінченного очікування ресурсів, зайнятих самими цими процесами. Виникнення взаємних блокувань задач призводить до збільшення часу їхнього виконання (може зростати до нескінченості), до не ефективного використання ресурсів КС (порожні цикли очікування).

Тому однією із проблем виконання процесів в КС є уникнення входження їх в стан взаємоблокування. Дані питання досліджувались Дейкстрою, Хоаром, Брінч-Хансеном, Деккером, Петерсеном, Коффманом, Хольтом [2]. На сьогодні розроблена велика кількість методів та алгоритмів для уникнення взаємоблокувань процесів. Проте частина з них носить лише теоретичний характер, оскільки не може бути реалізованою в сучасних КС. Інша частина при реалізації стає достатньо громіздкою і ресурсоємною. Тому розробники сучасних операційних систем (ОС) сімейств Windows та Linux, а також розробники сучасних СКБД не включають відомі алгоритми уникнення взаємоблокувань процесів.

За умови завантаження системи процесами до 20% відсутність таких засобів була допустима. Проте стрімкий розвиток апаратних засобів, зростання об’єму та складності ПЗ, яке займається вирішенням масштабних та відповідальних задач, не повинно дозволяти виникнення взаємоблокування, що в свою чергу вимагає розробки нових підходів до вирішення цієї задачі.

Для усунення суттєвих недоліків відомих методів та алгоритмів вирішення проблеми взаємоблокування було розроблено метод прогнозування взаємоблокування процесів, що враховує життєвий шлях процесу [3].

Постановка задачі. З метою визначення доцільності використання та порівняння запропонованого методу з існуючими засобами вирішення задачі взаємоблокування необхідно провести оцінку його часової складності та ефективності.

Оцінка часової складності методу. Для оцінки часової складності методу прогнозування взаємоблокувань процесів було використано web-сервер Apache2 з PHP 5 та sql-сервер MySQL. Для перевірки роботи методу на сервері запускався на виконання PHP-скрипт, представлений в лістинзі 1,

який при паралельному виконанні призводить до виникнення взаємоблокувань.

Лістинг 1

```
$sql = "SELECT COUNT(*) FROM t "; $x = mysql_query($sql);
$r = mysql_fetch_row($x);$max_id = $r[0];
$id1 = rand(0,$max_id); do { $id2 = rand(0,$max_id); } while ($id1==$id2);
mysql_query("START TRANSACTION;");
$sql1 = "select a from t where id = $id1 for update"; mysql_query($sql1);
//Вибір кортежу для обрахунку
usleep(100); //моделювання обробки даних
$sql2 = "select b from t where id = $id2 for update"; mysql_query($sql2);
mysql_query("COMMIT;"))
```

В представленому коді видалено бізнес-логіку, зате повністю збережено послідовність запитів, що при паралельному виконанні призводять до появи взаємних блокувань.

Рівні взаємоблокувань, представлені на рис. 1, показують взаємоблокування в КС. Суцільною лінією показано рівень взаємоблокувань при використанні стандартних засобів MySQL для виявлення заблокованих транзакцій. Пунктирною лінією показано рівень взаємоблокувань, що виникали при використанні запропонованого методу прогнозування взаємоблокувань.

Час виконання процесів при використанні різних механізмів вирішення взаємоблокувань показаний на рис. 2. Суцільною лінією показано середній час виконання процесів при використання стандартного механізму вирішення взаємоблокувань. Пунктирною – середній час виконання процесів при використанні запропонованого методу прогнозування взаємоблокувань. Штрих-пунктирною – середній час виконання процесів, за умови ненастання взаємоблокувань.

У випадку використання стандартного механізму виявлення взаємоблокувань спостерігається значне зростання часу виконання навіть при незначній кількості паралельних процесів. В цей самий період не спостерігається значного завантаження процесора (не більше 20%), оскільки процеси більшу частину часу очікують доступу до заблокованих ресурсів.

У випадку, коли не виникає взаємоблокувань процесів, час їх виконання зростає незначно (на 3% при кількості паралельних процесів 25 і рівні завантаження процесора не більше 20%).

Використання запропонованого методу прогнозування взаємоблокувань показує помітно пологіше наростання середнього часу виконання процесів, за таких самих умов.

Зниження часу виконання процесів при виконанні єдиного потоку

пояснюється відсутністю затримок, пов'язаних з очікуванням звільнення ресурсу.

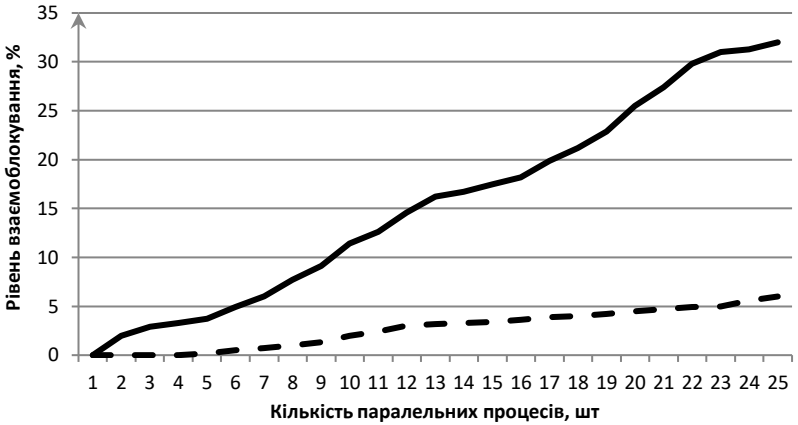


Рисунок 1 – Залежність рівня взаємоблокувань від кількості паралельних процесів

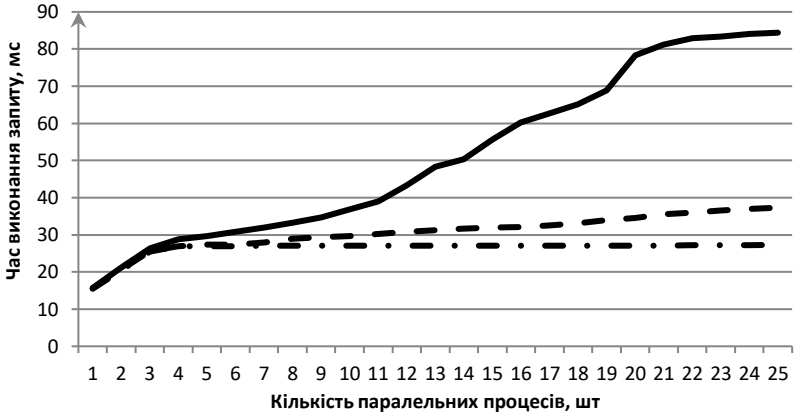


Рисунок 2 – Залежність часу обробки запитів від кількості паралельних процесів

В ході дослідження було проведено порівняння часу виконання фаз транзакцій при максимальному завантаженні системи і різних режимах роботи, що представлено на рис. 3.

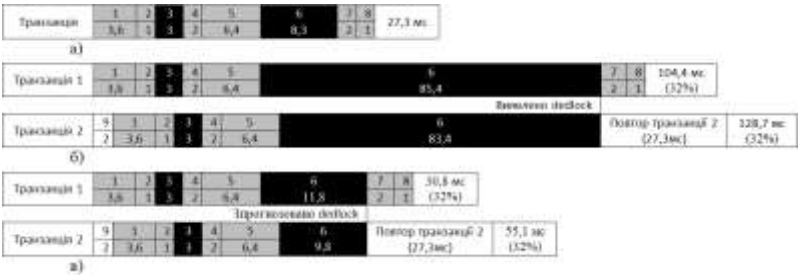


Рисунок 3 – Часові діаграми виконання фаз транзакції:

а) без взаємоблокування; б) взаємоблокування із стандартним виявленням; в) взаємоблокування із прогнозуванням. (1 – Підготовчі дії; 2 – Запуск транзакції; 3 – Очікування звільнення кортежу id1; 4 – Виконання 1-го SELECT; 5 – Опрацювання даних; 6 – Очікування звільнення кортежу id2; 7 – Виконання 2-го SELECT; 8 – Завершення транзакції; 9 – Затримка від початку запуску.)

На рис. 3.а представлена часова діаграма виконання фаз транзакції за умови, що при її виконанні не відбулось взаємоблокування. Середній час виконання транзакції склав 27,3мс, з них 11,3мс (41%) зайняло очікування заблокованих ресурсів. В ході дослідження таких транзакцій виявилось 36%.

На рис. 3.б представлено часову діаграму взаємоблокування двох транзакцій із стандартним механізмом їх вирішення. Транзакція 1 при першому звертанні (4) до таблиці БД вибирає та блокує кортеж з ключем id1. Транзакція 2 розпочинає виконуватись на 2мс пізніше, ніж Транзакція 1 і при першому звертанні (4) вибирає та блокує запис із ключем id2. Після опрацювання отриманих даних (5) транзакція 1 звертається до кортежу з id2. Оскільки він заблокований, то транзакція переходить до очікування звільнення ресурсу (6). В свою чергу транзакція 2 звертається до кортежу з id1. Він також виявляється заблокованим і транзакція 2 переходить в режим очікування звільнення ресурсу (6). Процеси потрапляють в ситуацію взаємного блокування і самостійно не можуть вийти з циклу нескінченного очікування. Взаємоблокування вирішує КС періодично аналізуючи граф процесів і ресурсів. Оскільки задача є алгоритмічно складною, то вона виконується через інтервали часу в 150-200мс. Після виявлення взаємоблокування один із процесів, що пізніше надійшов у систему, примусово завершується і повторно запускається на виконання. Інший з заблокованих процесів продовжує своє виконання. Великі часові проміжки між повторними аналізами графа призводять до значних затримок при виявленні взаємоблокувань. Середній час процесу, який був виконаний повторно складає 128,7мс, процесу, який продовжив роботу після взаємного блокування – 104,4мс. В ході дослідження таких транзакцій виявилось по

32%, оскільки процеси у взаємоблокування потрапляють парами.

На рис. 3.в представлено часову діаграму взаємоблокування двох транзакцій із прогнозуванням взаємоблокувань. До моменту запиту в транзакції 2 кортежу з ключем id2 часові діаграми ідентичні. В момент цього запиту (6) транзакція 2 потрапляє в граничний стан і для неї проводиться прогнозування. Час прогнозування складає близько 10мс. В результаті прогнозу визначається ймовірність взаємоблокування процесів, після чого обирається один із них. Цей процес знімається з виконання і запускається повторно. Середній час виконання процесу, що був виконаний повторно, складає 55,1мс; процесу, що продовжив роботу після взаємного блокування – 30,8мс. В ході дослідження таких транзакцій виявилось по 32%.

З отриманих досліджень середній час виконання транзакцій із стандартним механізмом виявлення взаємоблокувань склав: $27,3 \cdot 0,36 + 104,4 \cdot 0,32 + 128,7 \cdot 0,32 = 84,4$ мс, із прогнозуванням взаємоблокувань: $27,3 \cdot 0,36 + 30,8 \cdot 0,32 + 55,1 \cdot 0,32 = 37,3$ мс.

Застосування методу прогнозування взаємоблокувань забезпечило зменшення часу виконання процесів в $\frac{84,4}{37,3} = 2,3$ рази. Це дозволяє більш ефективно використовувати ресурси КС.

В роботі запропоновано оцінку часової складності та ефективності методу прогнозування взаємоблокувань поцесів в КС. Проведено його експериментальне дослідження на прикладі модифікованої СКБД MySQL. Середній час виконання процесу зменшився з 84,4мс при використанні стандартних засобів MySQL до 37,3 мс при використанні запропонованого методу. Отримані результати вказують на зменшення часових витрат на виконання процесів, що в свою чергу дозволяє опрацьовувати в 2,3 рази більшу кількість даних в одиницю часу для задач, в яких часто виникають взаємоблокування (відбувається конкурентна боротьба за ресурси).

Перелік посилань

1. Kaveh N. Deadlock detection in distribution object systems / Nima Kaveh, Wolfgang Emmerich // Software Engineering Notes. – September 2001. – Vol.26, No.5. – Pages 44 – 51.

2. Coffman E.G. System deadlocks / E.G. Coffman, M,J, Elphick, A. Shoshani. // Computing Surveys. – June 1971. – Vol.3, No.2. – Pages 67 – 78.

3. О.С. Савенко, С.В. Мостовий. Прогнозування потрапляння процесів у стан взаємоблокування в комп'ютерних системах. - Труды XIII Международной научно-практической конференции "Современные информационные и электронные технологии" (СИЭТ 2012). - Одесса: Одесский национальный политехнический университет,2012.- ст. 74-75

Дослідження проблем інформаційної безпеки в інформаційно-телекомунікаційних мережах

Атаманюк А.В., Джулій В.М., Кльоц Ю.П.
Хмельницький національний університет

Інформаційно-телекомунікаційна мережа надає різні сервіси для організації соціальних взаємовідносин між користувачами (абонентами). На сьогоднішній день найбільш популярними з них є соціальні мережі. З бурхливим ростом кількості користувачів інформаційно-телекомунікаційних мереж виникають і проблеми безпеки в них.

Розглянемо існуючі проблеми інформаційної безпеки в інформаційно-телекомунікаційних мережах, які актуальні для даного дослідження:

1. Використання глобальної мережі Інтернет як розподіленої інформаційно-телекомунікаційної системи. Найбільш вразливими компонентами системи, що часто атакуються, є: сервери; робочі станції; середовище передачі інформації; вузли комутації. Типові інформаційні впливи зловмисників:

- Прослуховування мережевого трафіку. Щоб прослухати трафік (sniffing) мережевий адаптер переводиться в «безладний» режим. У цьому режимі адаптер перехоплює всі мережеві пакети, що проходять через нього, а не тільки призначені даною адресою, як в нормальному режимі функціонування-технології – ARP Spoofing (ARP-poisoning), MAC Flooding і MAC Duplicating. Перехоплення здійснюється з використанням мережевих моніторів, з яких найбільш функціональними є Sniffer Pro від компанії Sniffer Technologies, IRIS Network Traffic Analyzer від компанії EYE і TCP Dump.

Наслідки. Сучасні мережеві протоколи (TCP / IP, ARP, HTTP, FTP, SMTP, POP3 тощо) не мають механізмів захисту (дані передаються у відкритому вигляді). Зловмисник, що перехоплює трафік між сервером і будь-яким вузлом мережі, може заволодіти аутентифікаційними даними користувача (отримати пароль).

Протидія. Відомо ряд методів визначення наявності запущеного сніфера в мережі, наприклад метод пінга, метод ARP, метод DNS і метод пастки.

- Сканування вразливостей. Результатом роботи сканера є інформація про систему, що містить список мережевого обладнання, комп'ютерів з запущеними на них службами, версіями мережевого ПЗ (а отже і вразливостей, властивих даному ПЗ), облікові записи користувачів. Сканування вразливостей зазвичай є етапом, що передує атаку. Саме результати сканування дозволяють точно підібрати експлойти для здійснення безпосереднього НСД.

Виявлення. Само по собі сканування не є незаконним. Однак, якщо сканування з боку зовнішньої, по відношенню до системи, мережі звичайне

явище, то сканування комп'ютерів з внутрішньої мережі – безумовно, інцидент безпеки, що вимагає негайної реакції з боку мережевого адміністратора. Виявити кроки сканування можна, вивчаючи журнали реєстрації міжмережевих екранів (МЕ). Однак такий підхід не дозволяє своєчасно реагувати на подібні інциденти. Тому сучасні МЕ і системи виявлення вторгнень СВВ мають модулі (plug-in), що дозволяють виявити сканування в режимі реального часу. Деякі сканери вразливостей використовують оригінальні методи, що дозволяють здійснювати сканування максимально приховано. Наприклад, в Nmap існують можливості, що дозволяють значно ускладнити виявлення сканування для СВВ.

Протидія. Використання мережевих СВВ, або періодичне вивчення журналів реєстрації МЕ.

- Мережеві атаки. Мережеві атаки можна розділити на: атаки, засновані на переповненні буфера (overflow based attacks). Вони використовують вразливість системи, яка полягає в некоректній програмної обробці даних. При цьому з'являється можливість виконання шкідливого коду з підвищеними привілеями; атаки, спрямовані на відмову в обслуговуванні (Denial Of Service attacks). Атаки не обов'язково використовують вразливості в ПЗ системи, що атакується. Порушення працездатності системи відбувається через те, що дані, що їй посилають, призводять до значної витрати ресурсів системи. Найпростішим прикладом атаки цього типу є атака «Ping Of Death». Суть її в наступному: на комп'ютер жертви надсилається сильно фрагментований ICMP-пакет великого розміру. Реакцією ОС Windows на отримання такого пакету є повне зависання.

- Атаки, засновані на використанні вразливостей в ПЗ мережевих додатків – експлойти (exploit). Даний клас атак заснований на експлуатації різних дефектів в ПЗ. Експлойти є шкідливими програмами, що реалізують відому вразливість в ОС або прикладному ПЗ, для отримання НСД до вразливого хосту або порушення його працездатності. Для експлойтів характерна наявність функцій подавлення антивірусних програм і МЕ. Наслідки застосування експлойтів можуть бути самими критичними. У випадку отримання зловмисником віддаленого доступу до системи, він має практично повний (системний) доступ до комп'ютера. Наступні дії і збиток від них можуть бути такими: впровадження троянської програми, впровадження набору утиліт для приховування факту компрометації системи, несанкціоноване копіювання зловмисником даних з жорстких та зовнішніх носіїв інформації, створення на віддаленому комп'ютері нових облікових записів з будь-якими правами в системі для подальшого доступу як віддалено, так і локально, крадіжка файлів з хешами паролів користувачів, знищення або модифікація інформації, здійснення дій від імені користувача системи.

Протидія. МЕ і СОВ, встановлені на системі, що атакується, в деяких випадків не в змозі відобразити дію експлойтів. Для успішного відображення атак експлойтів засоби захисту необхідно оновлювати, оскільки механізм виявлення вторгнень заснований на розпізнаванні сигнатур вже відомих атак. Хоча є розробки, здатні за завіреннями розробників відображати невідомі атаки, практика показує, що вони все ще не ефективні.

- Шкідливі програми. Шкідливі програми – це комп'ютерна програма або переносний код, призначений для реалізації загроз інформації, що зберігається в мережі, або для прихованого нецільового використання ресурсів або якого іншого впливу, що перешкоджає нормальному функціонуванню мережі. До шкідливих програми відносяться комп'ютерні віруси, троянські коні, мережеві черв'яки тощо.

Протидія. Типовим методом протидії є застосування антивірусних засобів, що працюють в режимі реального часу (моніторів). Для виявлення троянських програм існує спеціалізоване програмне забезпечення.

2. Проблема забороненого контенту. Залежно від законодавства країни різні матеріали можуть вважатися нелегальними. У більшості країн заборонені: матеріали сексуального характеру за участю дітей і підлітків, порнографічний контент, описи насильства, в тому числі сексуального, екстремізм і розпалювання расової ненависті. В українському законодавстві кілька законів регулюють питання надання інформації про фізичних та юридичних осіб, а саме: Закон України «Про інформацію» від 02.10.92, що регулює відносини щодо одержання і поширення інформації; Закон України «Про захист персональних даних» від 01.06.2010, що визначає захист і обробку персональних даних; Закон України «Про доступ до публічної інформації» від 13.01.2011, який надає право на отримання інформації, що знаходиться у володінні розпорядників.

Аналогічно з концепцією забезпечення комплексного захисту об'єкта інформатизації, можна сформувати повну множину функцій захисту від забороненої інформації. Під функцією захисту (ФЗ) розуміється сукупність однорідних в функціональному відношенні заходів, що регулярно здійснюються в автоматизованих системах різними засобами і методами з метою створення, підтримки і забезпечення умов, об'єктивно необхідних для надійного захисту інформації.

Перелік повної множини функцій захисту від забороненої інформації в соціальних мережах:

1. Попередження умов виникнення забороненої інформації. Функція реалізується за допомогою нормативно-правових актів. Вона не може повністю виключити загрозу поширення забороненої інформації в соціальних мережах, так як в цілому ситуація з дотриманням законів незадовільна, а в інтернет-просторі загострюється через технічні складнощі.

2. Попередження безпосередньої прояви забороненої інформації. Функція реалізується за рахунок механізмів прогнозування поширення забороненої інформації в соціальній мережі.

3. Виявлення забороненої інформації, яка проявилася. Функція пов'язана з моніторингом ІТКМ на предмет забороненої інформації на сторінках абонентів. Як правило, для реалізації даного захисту використовується різні СОРМ (система оперативно-розшукових заходів). Дана ФЗ пов'язана з проблемами контекстного пошуку, а також необхідністю контролю над всією системою.

4. Попередження впливу на абонентів забороненої інформації, яка проявилася. Функція може бути реалізована за допомогою автоматичного пересилання повідомлення з попередженням про відповідальність за розповсюдження забороненої інформації, аж до блокування абонента. Блокування може здійснюватися легітимними засобами за наявності доступу до керування системи та нелегітимними – при його відсутності (зламання акаунта). ФЗ ділиться на дві функції. Перша пов'язана з попередженням абонентів, на сторінках яких була знайдена заборонена інформація, а друга – з розсилкою попереджень потенційним одержувачам забороненої інформації.

5. Виявлення впливу забороненої інформації на абонентів. Функція пов'язана безпосередньо з фіксацією процесу поширення забороненої інформації, може бути реалізована через контекстний аналіз повідомлень.

6. Локалізація, обмеження впливу забороненої інформації на абонентів. Функція реалізується через блокування абонентів, що поширюють заборонену інформацію, або абонентів – потенційних розповсюджувачів. Дана ФЗ опирається на попередні функції і для її ефективної реалізації необхідний контроль над системою.

7. Ліквідація наслідків виявленого впливу забороненої інформації на абонентів. Функція пов'язана з видаленням забороненої інформації з системи. Для реалізації даної функції також необхідний контроль над системою.

На основі проведеного аналізу функцій захисту видно, що найбільш ефективні функції – це перші функції, оскільки вони забезпечують захист на початкових етапах. Наведені функції захисту мають свої недоліки. Найбільш перспективною ФЗ інженерно-технічного напряму є $ФЗ_2$. На даному етапі, маючи інформацію про топології ІТКМ і потенційних розповсюджувачів забороненої інформації, можливе прогнозування процесу її поширення.

Одним з підходів до прогнозування загрози поширення забороненої інформації (ЗПЗІ) є моделювання, наприклад, з використанням моделей впливу, моделей просочування і зараження. Дані моделі, як правило, не враховують топологічні особливості мережі (розподіл ступенів зв'язності, кластерний коефіцієнт, середня довжина шляху). Взаємодія між абонентами в межах цих математичних моделей описується переважно гомогенним

графом, що при моделюванні великомасштабних мереж (більше 10 млн. вузлів) може дати похибку прогнозування загрози поширення забороненої інформації більше 30%. Крім того, дані підходи мають в основному теоретичний характер, практика їх використання не виходить за межі експериментів. Таким чином, дослідження, спрямовані на створення моделей та алгоритмів загрози поширення забороненої інформації, актуальні і мають теоретичне і практичне значення у вирішенні проблеми забезпечення інформаційної безпеки в системах і мережах телекомунікацій.

Прведене дослідження проблем інформаційної безпеки виявило, що крім проблем, пов'язаних з використанням глобальної мережі Інтернет як розподіленої інформаційно-телекомунікаційної системи, які досить добре відомі і можна вирішити, існує маловивчена проблема забороненого контенту, існуючі рішення малоефективні.

Ще однією важливою проблемою є великомасштабність ІТКМ, яка заважає отримати дані з імітаційної моделі за прийнятний час. Розв'язання цієї задачі полягає у створенні аналітичної моделі загрози поширення забороненої інформації в ІТКМ.

Перелік посилань

1. Биячуев, Т.А. Безопасность корпоративных сетей: учеб. пособие / Т.А. Биячуев; под ред. Осовецкого Л.Г. – СПб.: СПбГУ ИТМО, 2016.– 161 с.

2. Лукацький, А.В. Предотвращение сетевых атак: технологии и решения / А.В. Лукацкий. – СПб. : Экспрес Электроника, 2014. – 268 с.

3. Тропіна М. Дослідження соціальних мереж як нового феномену сучасного світу / М. Тропіна // Наукові записки Малої академії наук України. Серія «Педагогічні науки» : [зб. наук. праць ; редкол. : С.О. Довгий (голова), О.Є. Стрижак, О.В. Лісовий, І.М. Савченко та ін.]. — Київ : Національний центр «Мала академія наук України», 2019. — Вип. 16. – С. 57-63

Модель тренувального процесу та метод обробки музичних даних програмної системи генерування музичних творів із використанням штучного інтелекту

Бабич І.Р.
Науковий керівник – к.т.н., доцент Яшина О. М.
Хмельницький національний університет

В основі моделі тренувального процесу системи генерування музичного контенту лежить удосконалене розуміння музичної структури та чітке передбачення нотних даних із наданням можливості генерування поліфонічної музики (коли одна нота на один крок у часі[1]), що узгоджується із музичними правилами.

Для відображення гармонійної та мелодійної структури між нотами, модель використовує двошарову архітектуру водночас з мережею довгої короткочасної пам'яті та з рекурентною мережею, в якій існують періодичні зв’язки вздовж осі ноти.

Маючи одну мережу довгої короткочасної пам'яті на осі часу, а іншу – на осі ноти, модель використовує вираз з роботи Даніеля Джонсона: «двоосьову» конфігурацію [2].

Дотримуючись наукової публікації Мун [3] як рекомендований орієнтир, до кожного шару LSTM було застосовано випадання 0,75. Обраним оптимізатором було ADADELTA [4]. Вибрана швидкість навчання - 1,0. Всі компоненти моделі тренувального процесу будуть оцінюватись у двох вимірах.

На основі моделі Джонсона буде розроблена модель тренувального процесу.

Дана модель має на меті вивчання гармонійних та мелодійних ритмічних ймовірностей з тренувальних поліфонічних MIDI файлів.

За допомогою цієї моделі можна здійснити тренування та валідацію моделі в чисельній формі. Використання потім натренованої моделі необхідне для створення згенерованої музичної композиції у вигляді файлу MIDI.

Загальний процес, що буде відбуватись в моделі, є заснованим на методі Джонсона і є вдосконаленим методом обробки музичних даних.

Даний метод має на меті використання вектору даних у нейронній мережі даного проекту, що називається "матрицею стану ноти", яку показано на рисунку 1. Вона представляє стан «відтворення» та «артикуляції» кожної ноти в діапазоні значень Midi та для кожного часового кроку через визначений проміжок часу (тобто 8 мір з 16 кроками часу на такт).

Алгоритм методу обробки музичних даних при тренуванні моделі є таким:

– генерування з тренувальних MIDI файлів партію векторів даних характеристик, єдиний тензор 4D, який називається «Note_State_Batch»;

– генерування відповідного тензору ймовірності відтворення ноти на кроці часу;

– зміщення тензору logP на 1 крок у часі і обчислення функції втрат;

– використання функції оптимізатора для оновлення параметрів.

Виходячи за рамки виразів Даніеля Джонсона, вдосконалений метод обробки музичних даних має більш розвинену гнучкість введення даних, а також більшу загальність у встановлені та зміні різних гіперпраметрів, таких як кількість шарів, розмір прихованої одиниці, довжина послідовності, проміжки часу, розмір пакетів, метод оптимізації та швидкість навчання. Також даний метод є параметризованим, тому користувачі, на відміну від оригінального методу Джонсона, можуть також встановити довжину

партитури нот, поданих у LSTM нотних осей, і тривалість кроків часу, поданих у LSTM часових осей. Розмір партитури та тривалість часових кроків є важливими особливостями, оскільки музика сильно варіюється залежно від жанру та виконавця. Загалом, даний метод дозволить адаптувати її до конкретних потреб користувача. Також код, який буде написаний для даного методу, має мінімізувати використання циклів «for», щоб збільшити швидкість за обчислювальний час, збільшуючи швидкодію тренування нейронної мережі.

Оригінальні необроблені музичні дані у вигляді файлів .MIDI спочатку обробляються для генерації кожного Note_State_Batch за допомогою пакету Python-Midi[5]. У роботі було використано цей пакет лише для імпортування сегментів файлів MIDI як Note_State_Batches, а також для створення файлів MIDI із згенерованих зразків.

$$Note\ State\ Matrix = \begin{bmatrix} [p,a]_N^{(1)} & \cdots & [p,a]_N^{(T)} \\ & \vdots & \\ [p,a]_1^{(1)} & \cdots & [p,a]_1^{(T)} \end{bmatrix}$$

Рисунок 1 – Матриця стану ноти в режимі обробки

Дана матриця стану ноти - це оброблений вектор характеристик нейронної мережі, де N - номер ноти Midi, виділеної з музичної композиції, T - номер часових кроків в пакеті, p - вказує двійкове значення відтворюваної ноти, а «a» вказує двійкове значення відповідної артикуляції.

Загальна структура програмної реалізації розділена на дві основні задачі: тренування або валідація моделі чисельно, а потім використання натренованої моделі для створення нових файлів .MIDI для якісного оцінювання. Обидві функції використовують однаковий граф моделі в різних контекстах: тренувальний процес, показано на розробленій моделі у рисунку 2, ітеративно вводить у модель Note_State_Batch, запускає модель через усі відповідні кроки часу та ноти, присутні в пакетах, і потім виводить тензор відповідного логіту або зворотну сигмоподібну ймовірність того, що дана нота може відтворюватись на даному проміжку часу.

Розглянемо функціональність всіх компонентів моделі тренувального процесу.

Функція логарифмічної правдоподібності вхідних даних інтерпретується як здатність моделі приймати в якості вхідних даних вектор нот на даному кроці часу і прогнозувати набір нот на наступному кроці часу. Функція втрат, псевдокод якої показаний на рисунку 3, обчислює перехресну ентропію між згенерованими Logits та Note_State_Batch (після вибудовування Logits до елементів Note_State_Batch, що відповідають одному кроку часу в майбутньому).

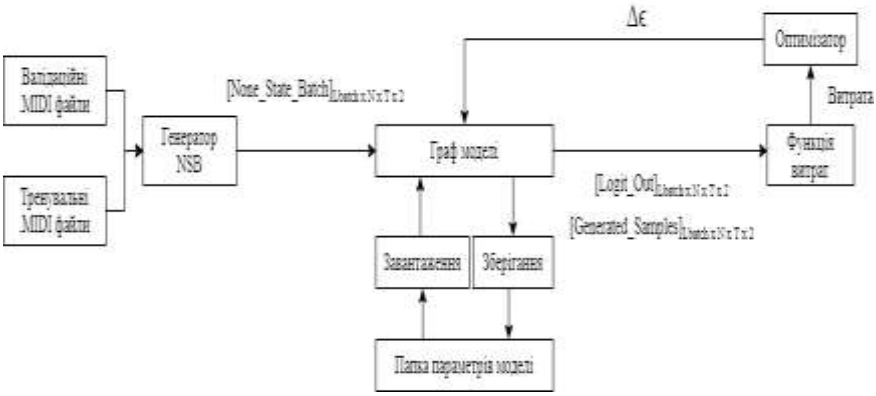


Рисунок 2 – Діаграма моделі тренувального процесу

Аргументи:

- Logits]$_{LbatchxNxTx2}$ (inverse sigmoid of Probability that play/articulate = 1;
- Labels(t-1) = [Note_State_Batch]$_{LbatchxNxTx2}$.

cross_entropy = sigmoid_cross_entropy_with_logits(logits=logits. labels=Labels) = - note_state ln(σ(logits)) + - (1 - note_state) In (1 - σ((logits)) = - note_state ln(Probability=1)+ -(1 - note_state) In (Probability=0)

$$Loss = \frac{1}{TNL\,L_{batch}} \sum_{b=1}^{L_{batch}} \sum_{t=1}^{T} \sum_{n=1}^{N} cross-entropy$$

$$\log-likelihood\ at\ 1\ time\ step = -\frac{1}{TL_{batch}} \sum_{b=1}^{L_{batch}} \sum_{t=1}^{T} \sum_{n=1}^{N} cross-entropy$$

$$\text{Log-likelihood at 1 time step} = -\frac{1}{TL_{batch}} \sum_{b=1}^{L_{batch}} \sum_{t=1}^{T} \sum_{n=1}^{N} cross-entropy$$

Повертає:

- Loss (scalar).

Рисунок 3 – Псевдокод для обчислення витрат

Під час задачі створення музики, представленої на рисунку 4, модель ітеративно виконується через один крок часу, кожен раз повертаючи генеровані вибірки(Generated_Samples) як вхідні дані Note_State_Batch для наступного кроку часу. Ці вибірки накопичуються, і це створює тензор генерованих вибірок у вигляді довільної довжини часу Note_State_Batchof. Потім згенеровані зразки перетворюються у файли .MIDI за допомогою функцій подальшої обробки з Python-Midi для якісної оцінки.

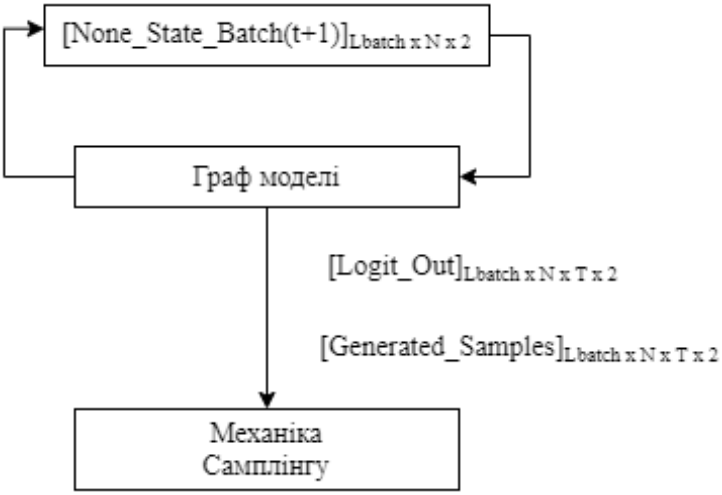


Рисунок 4 – Діаграма генерації музичної композиції

Функціональний розподіл самого графу моделі, показаного на рис. 5.

Було проведено декомпозицію вищого рівню графу моделі для того, щоб здійснити деталізацію кожного з компонентів генерації музичного контенту.

Компонент вхідного ядра приймає Note_State_Batch як свої вхідні дані і для кожної пари ноти й артикуляції генерує розширений вектор, який складається з:

- номера ноти Midi;
- одного гарячого вектора класу висоти ноти;
- вікна відтворення значень чи артикуляції відносно «n»-ої ноти (ефективний згорнутий аспект ядра моделі);
- вектор суми всіх відтворених нот у кожному класі висоти;
- бінарний вектор, що представляє позицію 16 ноти в межах міри.

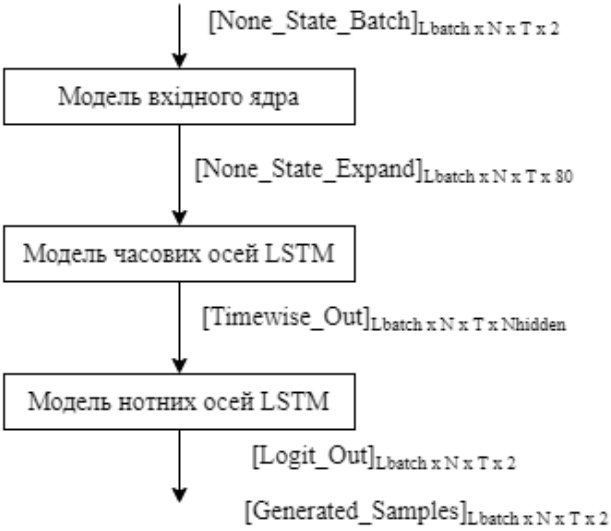


Рисунок 5 – Декомпозиція графу моделі

Псевдокод роботи вхідного ядра є показаним на рис. 6.

Наступний компонент декомпозиції графу ядра є компонентом обробки часових осей LSTM. У цьому компоненті комірка LSTM запускається вздовж часових осей на довжину виміру часового пакету. Ця операція виконується над вектором Note_State_Expand для кожної ноти паралельно з прив'язаними вагами. Ця частина графіку фіксує послідовні патерни музики і, у поєднанні з вхідним ядром, зберігає незмінність трансляції завдяки вікну введення відносних нот та прив'язаним вагам LSTM для всіх нот. Завдяки цим зв’язаним вагам, обчислення можуть виконуватися паралельно між нотами та між вибірками Матриці стану ноти як окремі ефективні пакети. Єдиний необхідний послідовний аспект - вздовж осі часу. Виконується довільна кількість каскадних комірок LSTM і після кожної комірки застосовується маска виключення.

Псевдокод для компоненту обробки часових осей LSTM зображений на рис. 7.

Заключним компонентом у декомпозиції, як описано у діаграмі, є компонент обробки нотних осей LSTM. Це потенційно один або багатошаровий компонент LSTM, такий як компонент часових осей LSTM, який включає в себе виключення після кожного шару. Однак, замість послідовного проходження по часової осі, цей етап проходить послідовно по нотній осі. Крім того, цей компонент включає в себе «локальний» зворотний зв'язок згенерованих вибірок до їхніх вхідних даних.

Аргументи:

– $[\text{None\_State\_Batch}]_{\text{LbatchxNxTx2}}$.

$$[p,a]_N^{(1)} = \text{'Kernel'} = \left[\begin{array}{c} [n]_{1x1} - \textit{MIDI note №} \\ \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}_{12x1} - \text{'nth' part} - \textit{pitchclass} \\ \begin{bmatrix} p_{n-12}^{(t)} \\ a_{n-12}^{(t)} \\ \vdots \\ p_{n}^{(t)} \\ a_{n}^{(t)} \\ \vdots \\ p_{n+12}^{(t)} \\ a_{n+12}^{(t)} \end{bmatrix}_{50x1} - \textit{part} - \textit{prev} - \textit{vicinity} \\ \begin{bmatrix} \textit{pitchclass} - \textit{count}_{(n)}^{(t)} \\ \vdots \\ \textit{pitchclass} - \textit{count}_{(n+11)}^{(t)} \end{bmatrix}_{12x1} - \textit{part} - \textit{context} \\ \begin{bmatrix} \textit{LSB} \\ \vdots \\ \textit{MSB} \end{bmatrix}_{4x1} - \textit{beat} \\ [0] - \textit{zero ?} \end{array}\right]_{80x1}$$

Повертає:

– $[\text{None\_State\_Expand}]_{\text{LbatchxNxTx80}}$.

Рисунок 6 – Псевдо-код для вхідного ядра

Аргументи:

– $[\text{None\_State\_Expand}]L_{batch}xNxTxL_{filt}$. Where $L_{filt} = 80$

$t = 1{:}T$

$h_{(1)n}^{(t)} = \text{LSTM}(h_{(1)n}^{(t-1)}, \text{None\_State\_Expand}_n^{(t)})$

$h_{(2)n}^{(t)} = \text{LSTM}(h_{(2)n}^{(t-1)}, h_{(1)n}^{(t)})$

$\vdots$

$\text{timewise\_out} = h_{(num\text{-}t\text{-}layer)n}^{(t)} = \text{LSTM}(h_{(num\text{-}t\text{-}layer)n}^{(t-1)}, h_{(num\text{-}t\text{-}layer)n}^{(t)})$

Повертає:

– $[\text{timewise\_out}]_{L_{batch}xNxTxN_{num\text{-}t\text{-}layer}}$.

Рисунок 7 – Псевдо-код для часових осей LSTM

Після кожного кроку ноти комірка LSTM виробляє пару логітів, що представляють зворотну сигмоїду ймовірності генерування відтворення або артикуляції для цієї ноти. Далі, з розподілу Бернуллі витягується зразок гри та артикуляції. Якщо вибірка відтворення має значення «0», що означає «не відтворено», артикуляційна вибірка також примусово доходить до «0», щоб уникнути генерації будь-яких значень, відсутніх у вхідних даних.

Згенерована дискретизована пара в ноті (n-1), об'єднана з вхідними даними часового LSTM на ноті (n), повертається назад на вхід нотатурного LSTM для кроку (n). Цей зворотний зв'язок створює умовну ймовірність для кожної ноти на основі фактичних значень, створених для нижчих нот. Це допомагає запобігти відтворенню дисонансних одночасних нот. Остаточними вихідними тензорами графу моделі є пакет Logits та відповідні згенеровані вибірки, які будуть використовуються для тренування та генерації музики, відповідно.

Псевдокод для компоненту обробки нотних осей LSTM зображений на рис. 8.

Аргументи:

- $[\text{timewise\_out}]_{L_{batch} \times N \times T \times N_{num\text{-}t\text{-}layer}}$.

For n = 1 to N:

$$\text{cell\_input}_n = \begin{bmatrix} timewise-out_n \\ note-gen_{n-1} \end{bmatrix}$$

$$h_{(1)n} = \text{LSTM}(h_{(1)(n-1)}, \text{cell\_input}_n)$$

$$h_{(2)n} = \text{LSTM}(h_{(2)(n-1)}, h_{(1)n})$$

$$\vdots$$

$$h_{(\text{num-t-layer})n} = \text{LSTM}(h_{(\text{num-t-layer})(n-1)}, h_{(\text{num-t-layer-1})n})$$

$$\text{Logits}_n = Wh_{(\text{num-t-layer})n} + b = \begin{bmatrix} Logit_n - play \\ Logit_n - artic \end{bmatrix}$$

$$\text{note\_gen}_n = \text{Sample}[\sigma(\text{Logits}_n) = \text{Prob}(\text{note\_n=1})] = \begin{bmatrix} play-gen_n \\ artic-gen_n \end{bmatrix}$$

if $(\text{play\_gen}_n = 0)$ then $\text{artic\_gen}_n = 0$

Повертає:

- $[\text{Logits}]_{L_{batch} \times N \times T \times 2}$.
- $[\text{note\_gen}]_{L_{batch} \times N \times T \times 2}$.

Рисунок 8 – Псевдо-код для нотних осей LSTM

Отже, на основі моделі Джонсона була розроблена модель

тренувального процесу та вдосконалений метод обробки музичних даних.

Дана модель має на меті вивчання гармонійних та мелодійних ритмічних ймовірностей з тренувальних поліфонічних MIDI файлів.

За допомогою цієї моделі можна здійснити тренування та валідацію моделі в чисельній формі. Використання потім натренованої моделі необхідне для створення згенерованої музичної композиції у вигляді файлу MIDI.

У вдосконаленого метода обробки музичних даних, на відміну від оригінального методу Деніела Джонсона, є розвинута гнучкість при введенні даних користувачем, загальність у встановлюванні різних гіперпараметрів, а також наявна параметризація, що дозволяє встановити довжину партитури нот та тривалість кроків часу необхідної для відповідної часової та нотної осі. Дані особливості збільшують можливість інтерактивної взаємодії живої людини з нейронною мережею, дозволяючи розробляти більш оригінальні мелодії.

Розроблюваний псевдокод з мінімальним використанням циклів дає також більшу швидкодію роботи програмного забезпечення для тренування нейронної мережі.

Перелік посилань

1. arXiv.org [Електронний ресурс] : [архів з відкритим доступом]. – Електронні дані (1 789 904 записів). – США : Корнельський університет, 2020. – Режим доступу: https://arxiv.org/pdf/1709.01620.pdf (дата звернення 07.08.2019). – Назва з екрана.

2. Джонсон Д. Д. Генерування поліфонічної музики за допомогою пов'язаних паралельних мереж / Д. Д. Джонсон // Обчислювальний інтелект у музиці, звуці, мистецтві та дизайні : зб. наук. Праць / 6-а Міжнародна конференція з еволюційних обчислень у комбінаторній оптимізації – Амстердам, 2017 – № 9 – С. 128 – 143

3. Мун Т, Чой Х, Лі Х, Сонг І. RnnDrop: новий випадок для RNN в ASR [Електронний ресурс] / Інститут інженерів електротехніки та електроніки – Електрон. дані. – Скоттсдейл, Арізона, США, 2015. – Режим доступу: https://ieeexplore.ieee.org/abstract/document/7404775 (дата звернення 11.02.2016). – Назва з екрана.

4. arXiv.org [Електронний ресурс] : [архів з відкритим доступом]. – Електронні дані (1 789 904 записів). – США : Корнельський університет, 2020. – Режим доступу: https://arxiv.org/pdf/1212.5701.pdf (дата звернення 22.12.2012). – Назва з екрана.

5. Github [Електронний ресурс] : [Інтернет-портал]. – Електронні дані. – [Сан Франциско, 2020]. – Режим доступу: https://github.com /vishnubob/python-midi (дата звернення 08.06.2015). – Назва з екрана.

Метод тестування на проникнення як засіб забезпечення безпеки корпоративної мережі

Гавронський В.Є.[1], Муляр І.В.[1], Яцків В.В.[2]
Хмельницький національний університет [1]
Західноукраїнський національний університет[2]

Необхідним елементом безпеки корпоративної мережі є тестування її програмного забезпечення як на етапах розробки, так і його використання. Проте не існує єдиної ефективної методики тестування на проникнення. Наприклад, існують криптографічні методи, що базуються на математичних алгоритмах та використовуються для шифрування даних з подальшою передачею їх відкритими каналами зв'язку. Додатковим ступенем захисту є приховання самого факту передачі інформації, наприклад, за допомогою методів цифрової стеганографії [1]. Іншим підходом до вирішення цієї задачі є використання в якості носіїв інформації хаотичних сигналів, які характеризуються широким неперервним спектром та високою інформаційною ємністю [2].

Таким чином, розроблення методу тестування безпеки програмного забезпечення для захисту інформації в корпоративній мережі є актуальним науковим завданням.

Проведені дослідження існуючих методик тестування безпеки програмного забезпечення, факторів, що впливають на цей процес, а також технологій математичної формалізації дозволили виявити ряд недоліків і обмежень щодо їх використання в умовах підвищеної уваги до програмного забезпечення у зловмисників.

На основі аналізу сучасних методів забезпечення безпеки ПЗ та систематизації сучасних підходів у предметній галузі захисту даних сформульовано актуальне наукове завдання, що полягає в розробленні методу тестування безпеки програмного забезпечення для захисту інформації в корпоративній мережі. Для цього розроблено математичну модель початкової генерації коду кібератаки несанкціонованого доступу до інформаційних ресурсів та GERT-модель початкової генерації коду кібератаки несанкціонованого доступу до ресурсів корпоративної мережі. Спрощену GERT-мережу представлено у вигляді рис. 1.

На рис. 1 перехід (1,2) характеризує операції вибору обладнання-жертви для злому. Переходи (2,3) (2,4) описують процес вибору методу атаки з урахуванням визначення операційної системи на вузлі-жертві (Windows або Linux, відповідно).

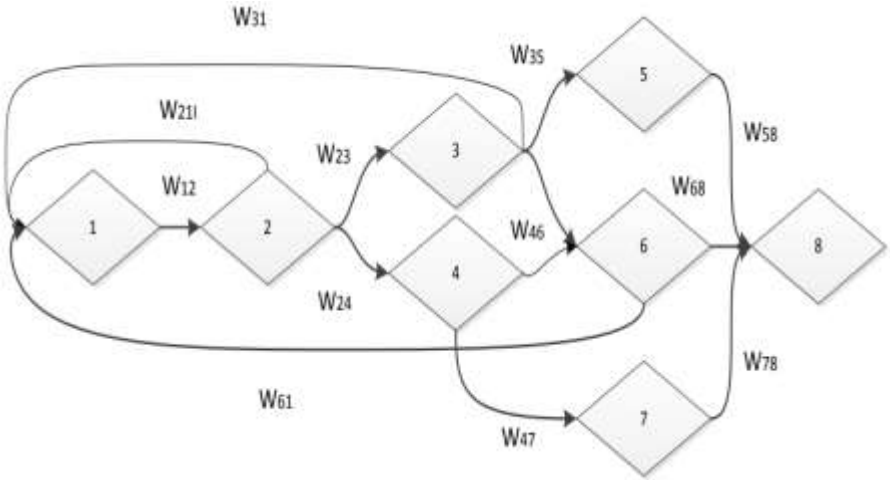


Рисунок 1 - GERT-мережа алгоритму генерації коду кібератаки несанкціонованого доступу

Переходи (2,1) і (3,1) наводять ситуацію, коли зловмисник, через певні причини, не зміг здійснити вибір методу атаки в межах заданого часу або характеристики знайденого шкідливого ПЗ, відповідно, не відповідають умовам і цілям кібератаки несанкціонованого доступу. Перехід (3,5) характеризує процеси пошуку ПЗ в глобальній мережі Інтернет, скачування та установлення ПЗ, що функціонує під OS Windows. Відповідно перехід (4,7) описує процес отримання вихідного коду в глобальній мережі Інтернет і компіляції ПЗ, яке функціонує під OS Linux.

Переходи (3,6) і (4,6) представляють процедури кодування та налагодження ПЗ під OS Windows і Linux, відповідно, за умови відсутності такого в глобальній мережі Інтернет. Перехід (6,1) характеризує ситуацію, коли зловмисник не зміг виконати операції кодування та налагодження шкідливого ПЗ в заданий для атаки час. Переходи (5,8), (6,8) і (7,8) описують процедури запуску шкідливого ПЗ і введення первинних параметрів IP-сервера вузла-жертви.

Проведені дослідження показали, що в складних GERT-мережах з можливими циклами відсутні прості методи знаходження особливих точок функції. Це пов'язано з тим, що для знаходження особливих точок необхідно вирішувати нелінійні рівняння, і чим складнішою є структура GERT-мережі, тим складнішим є і вихідне рівняння [3].

Синтез трас – це наступний етап розробленого методу. Вихідними даними для алгоритму синтезу трас «Синтез» виступає набір графів $\Gamma=\{G1,G2,...,GM\}$, таких, що всі вони споріднені до деякої траси *t*.

Алгоритм складається з двох кроків. На першому кроці виконується розбиття та синтез базових блоків (алгоритм «Базовий блок»). На другому кроці новостворені базові блоки з'єднуються ребрами (алгоритм «Синтез ребер»).

Спочатку враховуються внутрішні команди управління, обумовлені операторами BRANCH. Прямим проходом по операторах цього блоку будується орієнтований ациклічний граф, що описує вирази, які зустрічаються в цьому базовому блоці. Листові вершини в такому графі відповідають операторам INIT.

При побудові такого графа паралельно підтримується хеш-таблиця, ключі в якій відповідають виразам. Під час підрахунку хешу враховується код операції, а також хеші підвиразів. Тоді доповнення графа при перегляді чергового оператора зводиться до перевірки знаходження еквівалентного виразу в хеш-таблиці. Якщо такий вираз знайдено, то нові вершини не створюються. В іншому випадку необхідно створити нову вершину та додати вихідні ребра підвиразів, якщо такі в даному операторі є.

Після того, як граф побудовано, можна виключити повторне обчислення підвиразів: послідовно проглядаються оператори та виключаються ті, які обчислюють підвирази повторно. Слід зазначити, що додатково необхідно враховувати залежності, які проходять через біти слова стану: якщо будь-яка операція виставляє деякий біт, а інша операція, розташована далі, його читає, то першу з них виключати не можна.

У сукупності з уже наявними в середовищі можливостями реалізовані програмні компоненти дозволили повною мірою проводити запропоновану процедуру виділення алгоритму, у тому числі й ітеративно, з поповненням набору розглянутих трас в процесі аналізу.

Основною відмінністю розробленого методу є можливість його використання в ітеративному сценарії, коли в розгляд додаються нові траси за умови відсутності обмежень на природу аналізованого коду. Це дає можливість отримати уявлення про природу динамічної модифікації коду програми за допомогою побудови її еволюційного графа, розміри якого можуть розглядатися як одна з метрик складності програми.

Таким чином, розроблено математичну GERT-модель процесу генерації коду кібератаки несанкціонованого доступу. Запропонована математична модель відрізняється від відомих урахуванням у процесі математичної формалізації GERTмережі основних етапів генерації коду для операційних систем Windows або Linux з можливістю пошуку сучасних рішень у мережі Інтернет. Модель може бути використано для дослідження основних етапів генерації коду кібератаки з метою вироблення практичних рекомендацій протидії процесу несанкціонованого доступу до ресурсів корпоративної мережі.

Перелік посилань

1.Абазина Е.С. Цифровая стеганография: состояние и перспективы / Е.С. Абазина, А.А. Ерунов // Системы управления, связи и безопасности. – 2016. – № 2. – С. 182–201.

2.Колесов В.В. Применение дискретных хаотических алгоритмов в широкополосных телекоммуникационных системах / В.В. Колесов, А.И. Полубехин, Е.П. Чигин, А.Д. Юрин // Вестник СибГУТИ. – 2016. – № 3. – С. 77–92.

3.Барабаш О. В. Методи пошуку оптимальних маршрутів графа структури розгалуженої інформаційної мережі за заданим критерієм оптимальності при різних обмеженнях / О. В. Барабаш, І. П. Саланда, А. П. Мусієнко // Наукові записки Українського науково-дослідного інституту зв'язку. -К.: УНДІЗ, 2016. - №2 (42). - С 99-106.

Оптимальне кодування як засіб підвищення захищеності передачі шифрованих даних

Гончар Р. М., Нагребецький О.В., Орленко В.С., Чешун В.М.
Хмельницький національний університет

Постійне збільшення обсягів інформації в кіберпросторі і зростання її цінності зумовлює зацікавленість конкуруючих сторін і зловмисників у незаконному заволодінні нею, що створює постійну появу нових загроз щодо цілісності і конфіденційності інформації і актуальність заходів її захисту. Одним із основних способів захисту даних є шифрування, про що свідчить поява великої кількості методів та алгоритмів шифрування з різними функціональними можливостями і принципами дії (алгоритми DES-базовий, подвійний і потрійний DES, IDEA, ГОСТ 28147, Діффі-Хелмана, RSA тощо [1]) та тенденція до їх постійного вдосконалення.

Підвищення криптостійкості алгоритмів шифрування досягається як розробкою нових їх реалізацій, так і модернізацією-вдосконаленням існуючих або їх комбінуванням.

Проведені дослідження показують, що підвищення криптостійкості алгоритмів шифрування можна досягти попередньою підготовкою вхідних даних, в ході якого забезпечується порушення статистичних даних повторюваності символів вхідного тексту, тобто, збільшення характеристик його ентропії. Одним із варіантів такої підготовки вхідного тексту може бути застування методів оптимального нерівномірного кодування.

Для демонстрації можливості збільшення криптостійкості алгоритмів шифрування попередньою підготовкою вхідних даних оптимальним кодуванням обрано два класичних методи:

– кодування Хаффмена як класичний метод оптимального

ентропійного нерівномірного кодування даних, що дає стабільний оптимальний код на виході у відповідності до статистичних властивостей алфавіту вхідного тексту;

– шифри зсуву (заміни) як найпростіший варіант шифрування даних, що дозволяє наочно спостерігати вплив попередньої підготовки даних на криптостійкість алгоритму шифрування.

Недоліком шифрів зсуву є збереження статистичних характеристик появи символів первинного алфавіту (вхідного тексту) у вторинному алфавіті (зашифрованому тексті), що зумовлює їх низьку криптостійкість

Таким чином, криптостійкість шифрів зсуву може бути підвищена зміною властивостей алфавіту шифрування, для чого застосовується оптимальне кодування Хаффмена.

Стосовно кодування Хаффмена можна сказати наступне:

– оптимальне кодування Хаффмена є різновидом ефективного кодування;

– оптимальне кодування Хаффмена відноситься до класу ентропійних кодів;

– коди Хаффмена є нерівномірними кодами;

– коди Хаффмена є префіксними кодами;

– головною відмінністю коду Хаффмена від коду Шеннона-Фано є те, що він завжди дає оптимальний варіант ентропійного кодування за наявними статистичними даними;

– коди Хаффмена, як і інші методи ефективного кодування, безпосередньо не призначені для вирішення задач шифрування і захисту даних;

– стиснення даних оптимальним кодуванням Хаффмена дає позитивний ефект для захисту даних через зменшення розмірів повідомлень, що передаються (менша імовірність втрат інформації при передачі);

– відхилення варіанту кодування Хаффмена від примітивного кодування символів також ускладнює дешифрування тексту.

Для визначення перспектив застосування оптимального кодування Хаффмена в задачах криптографічного захисту дослідимо принципи цього кодування.

Побудова оптимального нерівномірного коду за методикою Хаффмена виконується за загальним алгоритмом, що включає наступні етапи:

– всі символи, що кодуються, упорядковуються в порядку зменшення ймовірностей;

– останні два символи впорядкованої множини (вони повинні мати найменші значення ймовірностей) замінюються допоміжним символом, значення ймовірності для якого визначається сумарною ймовірністю елементів, що замінюються;

– всі елементи нової множини знову упорядковуються на зменшення

ймовірностей;

– виконання попередніх двох операці повторюється до отримання єдиного допоміжного символу.

Для визначення кодових комбінацій символів виконується зворотний аналіз виконаних об'єднань.

Тобто, двом останнім символам, при об'єднанні яких було отримано символ з ймовірністю 1, присвоюються значення коду 0 і 1. Після цього розглядаються символи попереднього рівня, які прийняли участь в утворенні останніх допоміжних символів. Аналогічним чином їм ставляться у відповідність значення 0 та 1, які дописуються в молодший розряд кодових комбінацій.

Завершення кодування Хаффмена відбувається після досягнення етапу, на якому кодові комбінації будуть співставлені у відповідність всім символам вхідного алфавіту.

Отриманий за наведеним алгоритмом методикою нерівномірний код є оптимальним кодом Хаффмена для використаного в тексті алфавіту.

Для визначення перспектив застосування оптимального нерівномірного коду Хаффмена для збільшення криптостійкості алгоритмів шифрування розглянемо приклад практичного застосування методу кодування Хаффмена.

В якості прикладу дослідимо застосування методу Хаффмена для довільного набору з 100 латинських символів:

ADBCBADCSDCASCZDSMMMCCMMM
MAWSDSASWZDSMWUSCCMMMMEEE
WWBWBUDWSDWUSCCCCMMMMCWSD
SASCZDSMMMMMEWWZDSMMZDSMM

Потужність застосованого в наданому для кодування тексті алфавіту дорівнює 10, оскільки в ньому використано 10 символів: A, B, C, D, E, M, S, U, W, Z.

Спочатку визначимо рекомендовану розрядність двійкових кодів для заданого алфавіту при використанні примітивних (рівномірних) кодів:

$$n = \log_2 10 = 3.322.$$

Відповідно, для результату 3.322 отримуємо мінімально-достатню розрядність примітивного коду 4.

Для аналізу використаємо простий варіант послідовного кодування символів двійкового алфавіту двійковими числами (табл. 1.1).

Таблиця 1.1 – Примітивний код для кодування вхідного тексту

Символ алфавіту	A	B	C	D	E	M	S	U	W	Z
Код	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

Із застосуванням наведеного в таблиці 1.1 примітивного рівномірного коду початковий текст перетворюється в масив двійкових кодів загальною розрядністю 400 біт:

0000 0011 0001 0010 0001 0000 0011 0010 0110 0011
0010 0000 0110 0010 1001 0011 0110 0101 0101 0101
0010 0010 0101 0101 0101 0101 0000 1000 0110 0011
0110 0000 0110 1000 1001 0011 0110 0101 1000 0111
0110 0010 0010 0101 0101 0101 0101 0100 0100 0100
1000 1000 0001 1000 0001 0111 0011 1000 0110 0011
1000 0111 0110 0010 0010 0010 0010 0101 0101 0101
0101 0010 1000 0110 0011 0110 0000 0110 0010 1001
0011 0110 0101 0101 0101 0101 0101 0100 1000 1000
1001 0011 0110 0101 0101 1001 0011 0110 0101 0101

Дослідження статистичних властивостей вхідного тексту дозволяє визначити ймовірності появи в ньому символів алфавіту (табл. 1.2).

Таблиця 1.2 – Статистичні характеристики алфавіту вхідного тексту

Символ алфавіту	A	B	C	D	E	M	S	U	W	Z
Кількість повторів символу	6	4	14	12	4	25	16	3	11	5
Статистична імовірність	0.06	0.04	0.14	0.12	0.04	0.25	0.16	0.03	0.11	0.05

Наявність статистичних даних щодо імовірностей входження символів алфавіту до тексту, що подається на кодування, дозволяє застосувати алгоритм кодування Хаффмена. На рис. 1.1 зображене кодове дерево Хаффмена, побудоване за даними таблиці 1.2.

На основі кодового дерева Хаффмена формуємо кодові комбінації оптимального нерівномірного коду Хаффмена для кодування тексту (табл. 1.3).

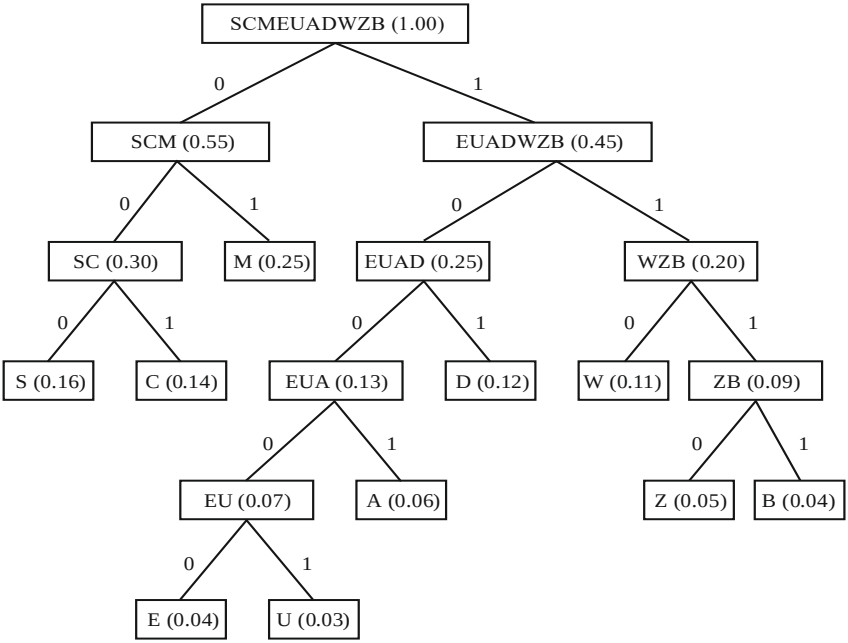


Рисунок 1.10 – Кодове дерево Хаффмена

Таблиця 1.3 – Оптимальний код Хаффмена для кодування алфавіту тексту

Символ алфавіту	A	B	C	D	E	M	S	U	W	Z
Код Хафмена	1001	1111	001	101	10000	01	000	10001	110	1110

Із застосуванням наведеного в таблиці 1.3 оптимального нерівномірного коду Хаффмена початковий текст перетворюється в масив двійкових кодів загальною розрядністю 304 біт:

1001 101 1111 001 1111 1001 101 001 000 101
001 1001 000 001 1110 101 000 01 01 01
001 001 01 01 01 01 1001 110 000 101
000 1001 000 110 1110 101 000 01 110 10001
000 001 001 01 01 01 01 10000 10000 10000
110 110 1111 110 1111 10001 101 110 000 101
110 10001 000 001 001 001 001 01 01 01
01 001 110 000 101 000 1001 000 001 1110
101 000 01 01 01 01 01 10000 110 110
1110 101 000 01 01 1110 101 000 01 01

Першочергово відзначимо позитивний ефект стиску даних із застосуванням оптимального нерівномірного кодування Хаффмена – замість 400 біт даних сам текст в оптимальному кодуванні займає лише 304 біти.

Визначимо коефіцієнт стиснення коду:

$$k = \frac{304}{400} * 100\% = 76\%.$$

Отриманий коефіцієнт є досить високим показником для несистематизованого тексту і дозволяє стверджувати, що ризик ушкодження оптимального коду тексту порівняно з початковим примітивним зменшується майже на чверть (24% пропорційно зменшенню розрядності коду). Це підтверджує ефективність використання оптимального кодування Хаффмена за призначенням для кодування тексту даного прикладу.

Для визначення перспектив застосування оптимального кодування Хаффмена для підвищення ефективності захисту даних з використанням шифрів зсуву перетворимо отриманий нерівномірний код початкового тексту в рівномірні кодові комбінації, потрібні нам для реалізації шифру зсуву.

Початково формуємо потокове (нерозривне послідовне) представлення тексту кодом Хаффмена. Для наочності і зручності подальшого аналізу коди Хаффена окремих символів з непарними номерами позицій в тексті виділено жирним шрифтом (застосовано виділення жирним символів через один для можливості наочного розрізнення кодів символів в бітовій послідовності):

1001101**1111**001**1111**1001**101**001**000**101**001**1001**000**001**1110**101**000**01**0**101**001**
001**01**01**01**1**001**110**000**101**000**1001**000**110**1110**101**000**01**110**10001**000**001**001**
01**01**01**1**10000**10000**10000**110**110**1111**110**1111**10001**101**110**000**101**110**10001**0**
00001**001**001**001**01**01**01**001**110**000**101**000**100**1000**001**1110**101**000**01**0**101**01**0**
110000**110**110**1110**101**000**01**01**1110**101**0000**01**.

Розіб'ємо отримане потокове представлення тексту кодом Хаффмена на чотирьохрозрядні комбінації (тетради) у відповідності із розмірністю початкового варіанту примітивного кодування:

1001 1011 **111**0 0111 **1**110 0110 **1**001 **000**1 0100 **1**100
1**000** 0011 **110**1 0100 **0**010 **1**010 **01**00 1**0**10 1**0**10 1**100**
1110 **000**1 0100 **0**100 1**000** 1101 **110**1 0100 **0**011 **10**10
0010 **00**00 1**001** 0101 0101 1000 0**100** **00**10 000**1** **10**11
0**111** 1110 **1111** 1000 1**101** 110**0** **00**10 1**110** 1000 1**000**
0010 **0**100 1**001** 0101 0101 0011 **1**000 0**101** 000**1** **001**0
00**00** **1**111 0**101** 000**0** **1**010 **1**010 **1**100 00**11** **0**110 **1110**
101**0** **00**01 **0**111 10**10** **1**000 **01**01

Аналіз отриманого розбиття дозволяє побачити певні особливості утворення рівномірного коду з нерівномірного:

– лише невелика кількість отриманих кодових комбінацій

отриманого рівномірного коду відповідає кодовим комбінаціям символів у реалізації коду Хаффмена (однократно зустрічаються коди 1001, 1110 і 1111);

– визначені в попередньому аналізі кодові комбінації коду Хаффмена 1001, 1110 і 1111 в рівномірному коді наявні не лише як коди символів A, B і Z відповідно – аналогічні кодові комбінації утворюються з фрагментів кодів інших символів;

– в більшості випадків кодові комбінації коду Хаффмена при переході від нерівномірного кодування до рівномірного дробленням бітової послідовності зазнають дроблення на частини між декількома кодовими комбінаціями рівномірного коду (кожна з комбінацій 101**1, 111**0, 01**11**, **11**10**,** 01**10**, як і більшість інших, утворені ділять між собою фрагменти двох кодових комбінацій коду Хаффмена);

– наслідком попередньої властивості є те, що більшість кодових комбінацій рівномірного коду, отриманого дробленням бітової послідовності коду Хаффмена, містять фрагменти декількох кодових комбінацій коду Хаффмена (кожна з комбінацій 101**1, 111**0, 01**11, 11**10, 01**10,** як і більшість інших, утворені з фрагментів двох кодових комбінацій коду Хаффмена, а комбінації **0**01**1** і **0**01**1,** як приклад, утворені з фрагментів трьох кодових комбінацій);

– серед кодових комбінацій рівномірного коду, отриманого дробленням бітової послідовності коду Хаффмена, зустрічаються кодові комбінації, які є фрагментами кодових комбінацій коду Хаффмена більшої розрядності (кодова комбінація 1000 в одному місці отримана урізанням з комбінації 10000, а в іншому - з комбінації 10001). При цьому можливе виокремлення частини розрядів як зі сторони молодших розрядів, так і зі сторони старших або з середньої частини довгої кодової комбінації коду Хаффмена.

Більш детальний аналіз рівномірних кодів, отриманих дробленням бітової послідовності оптимального нерівномірного коду Хаффмена на тетради, дозволяє визначити збільшення кількості наявних кодових комбінацій порівняно з початковим примітивним кодуванням.

В наведеному нижче представленні підкреслено різні види кодових комбінацій, отриманих дробленням бітової послідовності оптимального нерівномірного коду Хаффмена на тетради:

1001 1011 1110 0111 1110 **0110** 1001 **0001 0100 1100**
1000 0011 1101 0100 **0010 1010** 0100 1010 1010 1100
1110 0001 0100 0100 1000 1101 1101 0100 0011 1010
0010 **0000** 1001 **0101** 0101 1000 0100 0010 0001 1011
0111 1110 **1111** 1000 1101 1100 0010 1110 1000 1000
0010 0100 1001 0101 0101 0011 1000 0101 0001 0010
0000 1111 0101 0000 1010 1010 1100 0011 0110 1110
1010 0001 0111 1010 1000 0101

З прикладу чітко видно, що в отриманих дробленням бітової послідовності оптимального нерівномірного коду Хаффмена на тетради кодових комбінаціях наявні всі 16 можливих кодових комбінацій довжиною 4 біти, а не 10, як це було в початковому примітивному коді.

В таблиці 1.4 наведено статистичні дані щодо частоти появи різних кодових комбінацій рівномірного коду, отриманих дробленням бітової послідовності оптимального нерівномірного коду Хаффмена на тетради.

Таблиця 1.4 – Статистичні характеристики фінального рівномірного коду

№ з/п	Код	Кількість повторів коду	Статистична імовірність
1.	0000	3	0,039474
2.	0001	5	0,065789
3.	0010	6	0,078947
4.	0011	4	0,052632
5.	0100	8	0,105263
6.	0101	7	0,092105
7.	0110	2	0,026316
8.	0111	3	0,039474
9.	1000	8	0,105263
10.	1001	4	0,052632
11.	1010	8	0,105263
12.	1011	2	0,026316
13.	1100	4	0,052632
14.	1101	4	0,052632
15.	1110	6	0,078947
16.	1111	2	0,026316

Порівняння статистичних даних таблиць 1.2 і 1.4 свідчить про руйнування застосованою процедурою оптимального нерівномірного кодування Хаффмена статистичних залежностей між кодовими комбінаціями, що співставляються символам вхідного тексту при примітивному кодуванні, а також про здатність відповідної процедури змінювати кількість використовуваних комбінацій рівномірного коду і характер їх формування.

Невідповідність кількості кодових комбінацій застосовуваного коду кількості символів кодованого алфавіту та руйнування статистичних залежностей і ентропійних взаємозв'язків між символами є передумовою підвищення криптостійкості шифрування зсувом і може бути позитивно застосоване для криптографічних методів шифрування загалом.

Перелік посилань

1 Мережні інформаційні технології: навчальний посібник / О. А. Мясіщев, В. М. Джулій, С. Р. Красильников, В. М. Чешун. – Хмельницький : ХНУ, 2012. – 422 с.

2 Курко А. М. Введення в теорію інформації: посібник до вивчення дисципліни / А. М. Курко, В. Я. Решетник – Тернопіль : Тернопільський національний технічний університет імені Івана Пулюя, 2017. – 108 с.

3 Беркман Л.Н. Основні поняття та теореми теорії інформації: навчальний посібник для самостійної роботи студентів ВНЗ / Беркман Л.Н., Комарова Л.О., Чумак О.І. – Київ: ДУТ ННІТІ, 2015. – 91 с.

4 Класифікація основних методів кодування і кодів [Електронний ресурс] / Портал «stud.com.ua». – Режим доступу: https://stud.com.ua/171429/tehnika/klasifikatsiya_osnovnih_metodiv_koduvannya_kodiv (дата звернення 30.10.2020). – Назва з екрана.

5 Захист інформації в комп'ютерних системах та мережах: навчальний посібник / С.Г.Семенов, А.О.Подорожняк, О.І.Баленко, С.Ю.Гавриленко. – Х.: НТУ «ХПІ», 2014. – 251 с.

6 Тарнавський Ю. А. Технології захисту інформації: підручник / Ю. А. Тарнавський; КПІ ім. Ігоря Сікорського. – Київ : КПІ ім. Ігоря Сікорського, 2018. – 162 с.

7 Помехоустойчивое кодирование и декодирование в дискретных КПС [Електронний ресурс] / Портал «wiki». – Режим доступу: https://ru.bmstu.wiki/Помехоустойчивое_кодирование_и_декодирование_в_дискретных_КПС (дата звернення 30.10.2020). – Назва з екрана.

8 Остапов С. Е. Технології захисту інформації : навчальний посібник / С. Е. Остапов, С. П. Євсеєв, О. Г. Король. – Х. : Вид. ХНЕУ, 2013. – 476 с.

9 Майданюк В. П. Кодування та захист інформації. навчальний посібник / В. П. Майданюк. – Вінниця:ВНТУ, 2009. – 164 с.

10 Бойко Ю. М. Теоретичні аспекти підвищення завадостійкості й ефективності обробки сигналів в радіотехнічних пристроях та засобах телекомунікаційних систем за наявності завад : монографія / Ю. М. Бойко, В. А. Дружинінін, С. В. Толюпа. - Київ : Логос, 2018. - 227 с.

11 Прикладна криптологія : системи шифрування : підручник / О. Г. Корченко, В. П. Сіденко, Ю. О. Дрейс. – К. : ДУТ, 2014. – 448 с.

12 Кодування джерел інформації та каналів зв'язку: навчальний посібник / [Беркман Л.Н., Бондарчук А.П., Гайдур Г.І., Чумак Н.С.]. – Київ: ННІТІ ДУТ, 2018. – 91 с.

13 Alencar & Marcelo S Information theory / Alencar & Marcelo S. – NEWYORK : Momentum Press , LLC, 2015. – 178 p.

14 Galić I. Image compression with B-tree coding algorithm enhanced by data modelling with Burrows-Wheeler transformation / Irena Galić, Časlav Livada, Branka Zovko-Cihlar. //Journal for Control, Measurement, Electronics, Computing and Communications. – 2017. – Volume 57, Issue 1. – P. 76-88.

Аналіз поточного стану дій в області захищеної IP- телефонії

Гулечко М.С., Джулій В.М., Тітова В.Ю.
Хмельницький національний університет

Протоколи IP-телефонії поділяються на дві великі групи, а саме протоколи передачі медіа інформації по пакетним мережам, а також протоколи управління встановленням з'єднання. В першу групу входить протокол RTP (Real-time Transport Protocol), що працює поверх UDP (User Datagram Protocol) протоколу. Сукупність протоколів RTP / UDP / IP забезпечує транспортний механізм для мовного трафіку. Протоколи другої групи забезпечують управління при обслуговуванні виклику між абонентами. До цієї групи належать протоколи SIP (Session Initiation Protocol), H.323, MGCP (Media Gateway Control Protocol). Протоколи встановлення з'єднання можуть працювати як поверх UDP транспорту, так і по TCP (Transmission Control Protocol). Таким чином, сукупність протоколів (SIP / H.323 / MGCP) / (UDP / TCP) / IP формують сигнальний механізм для передачі мовного і медіа трафіку.

В силу загальнодоступності використовуваних каналів передачі голосової інформації в IP мережах особливої актуальності набуває забезпечення конфіденційності VoIP-сервісів. Для вирішення цього завдання можуть бути використані різні підходи: забезпечення прямого захищеного каналу між кореспондентами (наприклад, VPN-тунель); застосування спеціальних протоколів забезпечення безпеки для IP-сервісів.

Перший спосіб набув широкого поширення при побудові віртуальних корпоративних мереж, але для його реалізації кореспонденти повинні підтримувати VPN-протокол. Однак, багато VoIP-пристрої не підтримують VPN. Тому, для забезпечення безпеки досить часто застосовуються спеціальні протоколи забезпечення безпеки IP-телефонії.

До спеціальних протоколів забезпечення безпеки IP-телефонії відносяться протоколи Secured SIP, SRTP, MIKEY, SDES, ZRTP, DTLS, S-MIME. Ці протоколи можна розділити на 3 категорії: протоколи захисту сигналізації (Secured SIP); протоколи захисту медіаінформації (SRTP); протоколи генерації і розподілу ключів для протоколів захисту медіаінформації (MIKEY, SDES, ZRTP, DTLS).

Протоколи захисту сигналізації призначені для забезпечення безпеки інформації про телефонні номери, підтримуваних кодеків. Для вирішення цього завдання використовується Secured SIP (SSIP, SIP / TLS). Цей протокол працює за аналогією з протоколом HTTPS, організовуючи між кореспондентом і сервером SSL тунель з використанням сертифікатів і відкритого ключа. Всі SIP-повідомлення (сигналізація) передаються з цього тунелю. Недоліком протоколу є необхідність застосування інфраструктури відкритих ключів, що використовується для організації TLS.

Для забезпечення конфіденційності при передачі мови широко використовується захищений протокол реального часу - Secure Real-time Transport Protocol (SRTP), який реалізує функції криптографічного захисту - шифрування і аутентифікації мовних повідомлень на основі алгоритму шифрування AES.

Криптографічний захист пакетів голосової інформації виконується протоколом SRTP в режимі реального часу і не вносить змін в ймовірносно-часові характеристики протоколу RTP. Але для його роботи необхідно попереднє формування криптографічних ключів. Це завдання вирішує протокол розподілу ключів (ПРК).

Рекомендація RFC 3711 описує дві складових - власне протокол SRTP для перенесення і криптозахисту медіа даних, а також протокол SRTCP (Secure Real-time Transport Control Protocol) для управління медіа сесією.

Основними завданнями протоколу SRTP є виконання таких функцій: шифрування переданих голосових даних; аутентифікація переданих повідомлень; захист від передачі повторних пакетів; збереження смуги пропускання, стиснення RTP заголовків.

Основними завданнями протоколу SRTCP є виконання таких функцій: шифрування переданих даних; аутентифікація переданих повідомлень. Аутентифікація і шифрування можуть працювати незалежно один від одного. Таким чином, можливий варіант, коли шифрування вимкнено і SRTP здійснюється виключно з метою аутентифікації. Обмеженням протоколу є те, що аутентифікація повідомлення обов'язкова в SRTP і не може бути відключена.

Протоколи генерації і розподілу ключів для захисту медіаінформації.

Протоколи третьої групи, за аналогією з протоколами розподілу ключів в бездротових мережах, призначені для генерації і розподілу між кореспондентами ключів шифрування медіаінформації. Для вирішення цього завдання можна використовувати протоколи MIKEY, SDES, ZRTP, DTLS.

Протокол обміну ключами MIKEY описаний в рекомендаціях RFC3830 і RFC6309. MIKEY має кілька режимів роботи, що визначають спосіб формування секретного ключа сесії SRTP: режим встановленого ключа, режим відкритого ключа та режим Діффі-Хелмана. Причому другий і третій режими не захищають від атаки вторгнення в середину (MiTM, Man In the Middle) і вимагають реалізації механізму аутентифікації повідомлень. Транспортом для переносу повідомлень протоколу може виступати як SIP / SDP, так і протокол RTSP (Real Time Streaming Protocol). SDES (Session Description Protocol Security) описується в RFC4568. Суть протоколу полягає в тому, що один з кореспондентів передає ключ в SIP повідомленні по сигнальному каналу. Кореспондент отримує його і використовує для шифрування. Однак при цьому обмін сигнальними повідомленнями повинен бути захищений від зловмисника. З цієї причини - SDES може

використовуватися тільки при наявності SIP / TLS захищеного з'єднання з цифровим сертифікатом сервера. Також даний спосіб не забезпечує безпеки з кінця в кінець. Це означає, що якщо з'єднання буде виконуватися через IP АТС, SDES буде виконувати розподіл ключів між кореспондентом А і IP PBX, між кореспондентом Б і IP-телефонною станцією, але не між кореспондентами А і Б безпосередньо.

Протокол DTLS для SRTP описується в RFC 5764. Протокол описує формування медіа-сесій точка-точка з двома учасниками з жорстким фіксуванням портів UDP кореспондента і респондента. Повідомлення протоколу передаються спільно з RTP пакетами. Кожна сесія містить одну DTLS асоціацію і два SRTP контексту (для SRTP і SRTCP). Для організації сесії (DTLS-асоціації) кореспонденти виконують обмін повідомленнями, DTLS handshake. Так як в основі протоколу лежить TLS, що використовує інфраструктуру відкритих ключів (Public Key Infrastructure, PKI), то застосування TLS можливо теж тільки при наявності PKI.

Одним з найбільш перспективних протоколів генерації ключів є ZRTP. Протокол застосовується в додатку для Android CsipSimple, програмних телефонах Jitsi, Phoner, програмних АТС FreeSwitch і Asterisk, апаратних VoIP шлюзах компанії UM-Labs. Відмінною особливістю ZRTP протоколу є можливість забезпечення безпеки від точки до точки, від одного кореспондента до іншого. Завданнями протоколу ZRTP є: генерація ключових параметрів SRTP сесії; забезпечення конфіденційності повідомлень протоколу; забезпечення аутентифікації кореспондентів; захист від атаки вторгнення посередині, як з використанням, так і без використання інфраструктури відкритих ключів.

Протокол передбачає роботу кореспондентів по топології точка-точка, при цьому окремо виділяється можливість застосування протоколу при багатопотоковому режимі, коли необхідно організувати кілька захищених медіа потоків. Крім того, передбачений режим роботи з легітимним посередником, яким може бути, наприклад, корпоративна телефонна станція. Кожен з кореспондентів-учасників протоколу повинен мати встановлений ідентифікатор (ZID), який повинен бути унікальний.

В основі протоколу - обмін ключами по алгоритму Діффі-Хелмана. Особливістю протоколу є передача параметрів всередині RTP пакетів, залишаючи пакети сумісними з RTP / AVP профілем. В цьому випадку, ZRTP-несумісним пристроєм ZRTP-пакети просто відхиляються і не впливають на встановлене з'єднання.

Для аутентифікації кореспондентів, а також виключення атаки вторгнення в середину (MiTM, Man in The Middle), протокол ZRTP передбачає використання короткого аутентифікаційного рядка (SAS, Short Authentication String), а також частини ключового матеріалу від попередніх сесій між кореспондентами. Для контролю цілісності переданих повідомлень

кожне повідомлення ZRTP включає в себе код CRC, а також код аутентифікації повідомлення MAC (Message Authentication Code). MAC обчислюється, як ключова хеш-функція, яка узгоджується на першій фазі протоколу.

Виявлення помилки тільки в хеш-повідомленні, як правило, означає виявлення атаки MiTM, оскільки спотворення за рахунок канальних помилок виявляються і при перевірці CRC ZRTP пакета. Протокол виконується послідовно в чотири фази: виявлення; підтвердження; обчислення ключів; завершення. У загальному випадку, ZRTP працює на самому початку розмови кореспондентів, відразу після завершення роботи протоколу SIP, як починає працювати в сторони протокол RTP

Існуючі дослідження в області робіт із захисту голосових зв'язків можна розділити на кілька категорій, а саме: розробка безпечних систем IP-телефонії; аналіз безпеки, що забезпечується системами IP-телефонії; аналіз безпеки, що забезпечується окремими протоколами VoIP, а також аналіз самих протоколів.

При оцінці впливу протоколів забезпечення безпеки на якість потрібно враховувати особливості IP-телефонії в порівнянні з традиційною телефонією. Так, в традиційній телефонії час відгуку вузла зв'язку, тобто час з початку передачі інформації про заняття абонентської лінії до моменту отримання кінцевим обладнанням сигналу готовності до прийому номера, визначається готовністю станції обслужити виклик. У IP-телефонії цей час визначається кінцевим обладнанням і не залежить від поточного стану телефонної станції.

Необхідно оцінити, як протоколи безпеки IP-телефонії можуть впливати на нормовані показники функціонування мереж телефонної мережі зв'язку. Застосування SIP-S може впливати на норму "втрати викликів" в разі, якщо при сценарії абонент-абонент один з кореспондентів використовує політику безумовного використання SIP-S, а другий не підтримує SIP-S протокол. Деяка затримка додатково може виникати за рахунок часу, необхідного на організацію TLS каналу між кореспондентами, необхідного для роботи SIP-S протоколу.

Протоколи розподілу ключів впливають на час встановлення з'єднання або на час організації захищеного мовного каналу, в залежності від місця спрацювання протоколу в сценарії з'єднання. Так протокол ZRTP може працювати після встановлення з'єднання, починаючи з етапу, коли один з кореспондентів зняв трубку. В цьому випадку, протокол впливає на норму "час встановлення з'єднання". Інші протоколи також вимагають передачу додаткових повідомлень, що може збільшувати значення нормованих параметрів.

Перелік посилань

1. Борисов М. А. Основы программно-аппаратной защиты информации. / М. А. Борисов, И. В. Заводцев, И. В. Чижов. – М.: УРСС: Либроком, 2013. – 370 с.

2. Гольдштейн Б.С. IP-телефония. / Б.С.Гольдштейн, А.В.Пинчук, А.Л.Суховицкий. - М.: Радио и связь, 2015-336 с.

3. Тарнавський Ю. А. Технології захисту інформації: підручник / Ю. А. Тарнавський; КПІ ім. Ігоря Сікорського. – Київ : КПІ ім. Ігоря Сікорського, 2018. – 162 с.

4. Бойко Ю. М. Теоретичні аспекти підвищення завадостійкості й ефективності обробки сигналів в радіотехнічних пристроях та засобах телекомунікаційних систем за наявності завад : монографія / Ю. М. Бойко, В. А. Дружинінін, С. В. Толюпа. - Київ : Логос, 2018. - 227 с.

5. Прикладна криптологія : системи шифрування : підручник / О. Г. Корченко, В. П. Сіденко, Ю. О. Дрейс. – К. : ДУТ, 2014. – 448 с.

6. Шинкарук О.М. Основи функціонування багатоканальних систем передачі інформації: навч. посіб./ О.М. Шинкарук, Ю.М. Бойко, І.І. Чесановський. – Хмельницький : ХНУ, 2011. – 245с.

7. Кодування джерел інформації та каналів зв'язку: навчальний посібник / [Беркман Л.Н., Бондарчук А.П., Гайдур Г.І., Чумак Н.С.]. – Київ: НННТІ ДУТ, 2018. – 91 с.

Інформаційна модель захисту інформації.

Даценко В.С., Тітова В.Ю., Шевчук І.М.
Хмельницький національний університет

Розуміючи інформаційну безпеку як «стан захищеності інформаційного середовища суспільства, що забезпечує її формування, використання і розвиток в інтересах громадян та організацій», правомірно визначити загрози безпеки інформації, джерела цих загроз, способи їх реалізації та цілі, а також інші умови і дії, що порушують безпеку [1]. При цьому, природно, слід розглядати і заходи захисту інформації від неправомірних дій, що призводять до заподіяння шкоди.

Практика показала, що для аналізу такого значного набору джерел, об'єктів і дій доцільно використовувати методи моделювання. При цьому слід враховувати, що модель не копіює оригінал, а є простішою. При цьому, модель повинна бути досить загальною, щоб описувати реальні дії з урахуванням їх складності [2].

Можна запропонувати компоненти моделі захисту інформації на першому (інформаційному) рівні декомпозиції. На нашу думку, такими компонентами інформаційної моделі можуть бути:

- об'єкти загроз;
- загрози;
- джерела загроз;
- цілі загроз з боку зловмисників;
- джерела інформації;
- способи неправомірного оволодіння інформацією (способи доступу);
- напрямки захисту інформації;
- способи захисту інформації;
- засоби захисту інформації.

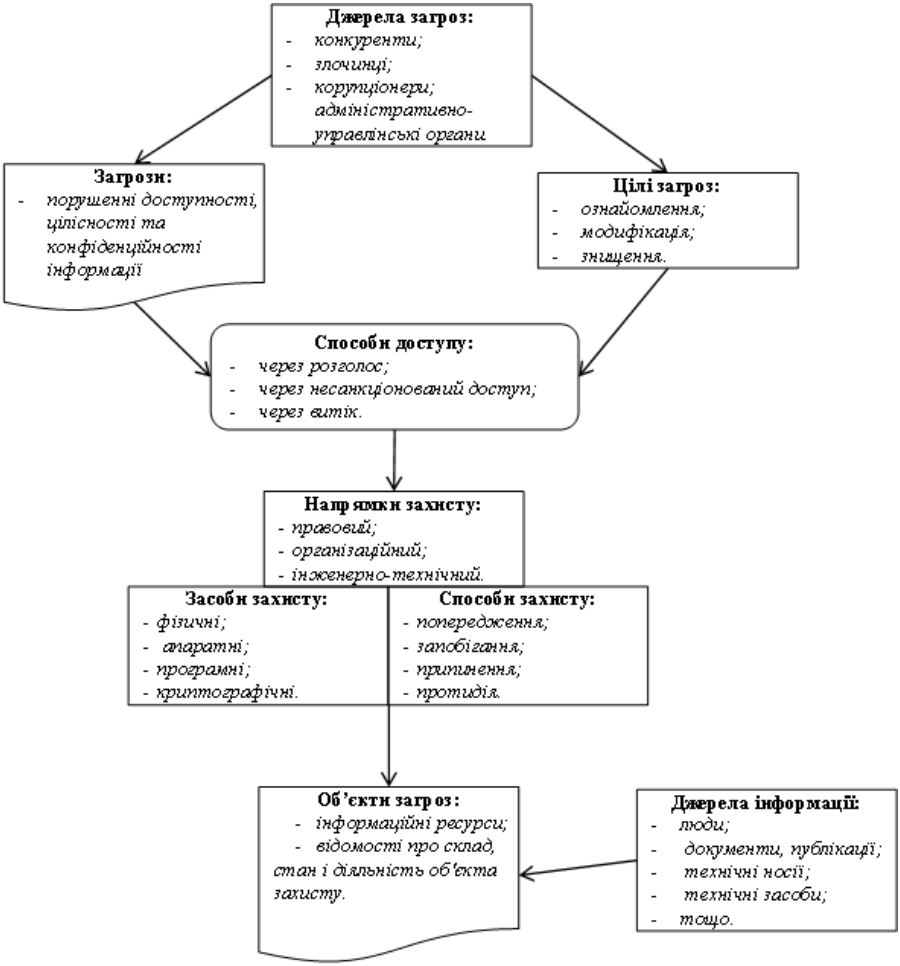


Рисунок 1 – Інформаційна модель захисту інформації

Об'єктами загроз інформаційної безпеки виступають відомості про склад, стан і діяльність об'єкта захисту (персоналу, матеріальних і фінансових цінностей, інформаційних ресурсів), тощо.

Загрози інформації виражаються в порушенні її доступності, цілісності і конфіденційності.

Джерелами загроз виступають конкуренти, злочинці, корупціонери, адміністративно-управлінські органи, тощо.

Джерела загроз переслідують при цьому наступні цілі: ознайомлення з відомостями, їх модифікація в корисливих цілях і знищення для нанесення прямих матеріальних збитків.

Неправомірне заволодіння відомостями можливо за рахунок їх розголошення джерелами інформації, за рахунок витоку через технічні засоби і за рахунок несанкціонованого доступу до відомостей. Джерелами інформації є люди, документи, публікації, технічні носії інформації, технічні засоби забезпечення виробничої та трудової діяльності, продукція і відходи виробництва.

Основними напрямками захисту інформації є правовий, організаційний та інженерно-технічний захист інформації, як показники комплексного підходу до забезпечення інформаційної безпеки.

Засобами захисту інформації є фізичні засоби, апаратні засоби, програмні засоби та криптографічні методи. Останні можуть бути реалізовані як апаратно, програмно, так і змішано-програмно-апаратними засобами. В якості засобів захисту виступають всілякі заходи, шляхи, способи і дії, що забезпечують попередження протиправних дій, їх запобігання, припинення та протидія несанкціонованому доступу.

В узагальненому вигляді розглянуті компоненти у вигляді інформаційної моделі безпеки інформації наведені на наступній схемі (рис. 1).

Співставлення об'єкта (фірма, організація) і суб'єкта (конкурент, зловмисник) в інформаційному процесі з протилежними інтересами можна розглядати з позиції активності, яка призводить до оволодіння інформацією. У цьому випадку можливі такі ситуації:

- власник (джерело) не приймає ніяких заходів до збереження інформації, що дозволяє зловмисникові легко отримати цікаві для нього відомості;
- джерело інформації суворо дотримується заходів інформаційної безпеки, тоді зловмисникові доводиться докладати значних зусиль до здійснення доступу до потрібних йому відомостей, використовуючи для цього всю сукупність способів несанкціонованого проникнення;
- проміжна ситуація - це витік інформації по технічним каналам, при якій джерело ще не знає про це (інакше він прийняв би заходи захисту), а

зловмисник легко, без особливих зусиль може їх використовувати в своїх інтересах.

Отже, на основі вищевикладеного можна зробити наступні висновки:

1. Інформація - це ресурс. Втрата інформації приносить моральні чи матеріальні збитки.

2. Умови, що сприяють неправомірному оволодінню інформацією, зводяться до її розголошенню, витоку і несанкціонованого доступу до її джерел.

3. У сучасних умовах безпека інформаційних ресурсів може бути забезпечена тільки системою захисту інформації, яка буде протидіяти загрозам через блокування неправомірних способів доступу та охоплювати усю множину існуючих способів за засобів захисту інформації

Перелік посилань

1. Теоретичні засади поняття інформаційної безпеки та класифікація загроз системі інформаційного захисту/ О. В. Черевко. // Ефективна економіка. – 2014. – №5. – Режим доступу: http://nbuv.gov.ua/UJRN/efek_2014_5_103

2. Інформаційна безпека. Навчальний посібник. Ч.1/С.В. Кавун, В.В. Носов, О.В. Мажай. – Харків: Вид. ХНЕУ, 2008. – 352 с.

3. Основні поняття. НД ТЗІ 1.1-003-99: Термінологія в галузі захисту інформації в комп'ютерних системах від несанкціонованого доступу. – Київ: Департамент спеціальних телекомунікаційних систем та захисту інформації Служби безпеки України. – 1999. – 30 с.

Метод приховування великого об'єму даних в файлах формату JPEG

Дацюк Р.М., Муляр І.В.
Хмельницький національний університет

Актуальність вивчення стеганографії постійно зростає, оскільки з поширенням персональних комп'ютерів, і особливо Інтернету, можливість конфіденційно передавати інформацію привертає увагу значної кількості людей. Переважна більшість теоретичних та практичних досліджень у галузі стеганографії присвячена розробці нових та вдосконаленню існуючих методів приховування даних. Кількість останніх постійно зростає з часом, але в сучасній науковій літературі [1] відсутня чітка класифікація таких методів, що ускладнює пошук і не дозволяє повною мірою оцінити рівень існуючих досягнень для їх подальшого ефективного використання.

Аналізуючи процес розвитку комп'ютерної стеганографії, можна сказати, що в найближчі роки інтерес до розробки її методів буде дедалі

більше зростати. Актуальність проблеми інформаційної безпеки постійно зростає і стимулює пошук нових методів захисту інформації. З іншого боку, швидкий розвиток інформаційних технологій дає можливість впроваджувати ці нові методи захисту.

Стеганографічні методи поряд із криптографічними займають важливе місце серед методів захисту інформації.

Але якщо в криптографії наявність зашифрованого повідомлення саме по собі привертає увагу зловмисника, то в стеганографії прихований зв'язок залишається невидимим, що робить організацію цього процесу досить актуальною.

Загальною особливістю стеганографічних методів є те, що приховане повідомлення або додаткова інформація вбудовується в якийсь нешкідливий, непомічений об'єкт або контейнер, в результаті чого з'являється повідомлення, яке потім відкрито транспортується до одержувача за каналом зв'язку. або зберігаються як такі

Але, більшість стеганографічних алгоритмів дозволяють приховувати невеликі об'єми інформації. Але на практиці часто виникає потреба в прихованій передачі значних масивів даних. Тому дослідження в напрямку розробки методу, що приховує великі об'єми інформації в відомих графічних форматах, для їх подальшої передачі є актуальним.

JPEG - можна сказати один з нових і досить потужних алгоритмів. Він працює на зонах 8x8, де яскравість і колір змінюються досить плавно. В результаті, коли матриця такої області розкладається на подвійний рядок уздовж косинусів, значущими є лише перші коефіцієнти. Таким чином, стиснення в JPEG відбувається за рахунок плавної зміни кольорів зображення.

Структуру JPEG-файлу зображено на рисунку 1.

Цей формат має таку універсальну структуру:

- Title (2 байти) : $ff, $d8 (SOI) (ідентифікується JPEG/JFIF файл);
- фрагмент APPO. Для JFIF файлів йде відразу за маркером SOI;
- довільна кількість "фрагментів" (подібні IFF (Image File Format) частинам);
- кінець (2 байти) : $ff, $d9 (EOI) [2].

Усі фрагменти мають таку структуру:

- Title (4 байти) :
- $ff ідентифікатор фрагменту
- n клас фрагменту (1 байт)
- sh, sl - розмір фрагменту.
- Вміст фрагменту, максимально 65533 байти.

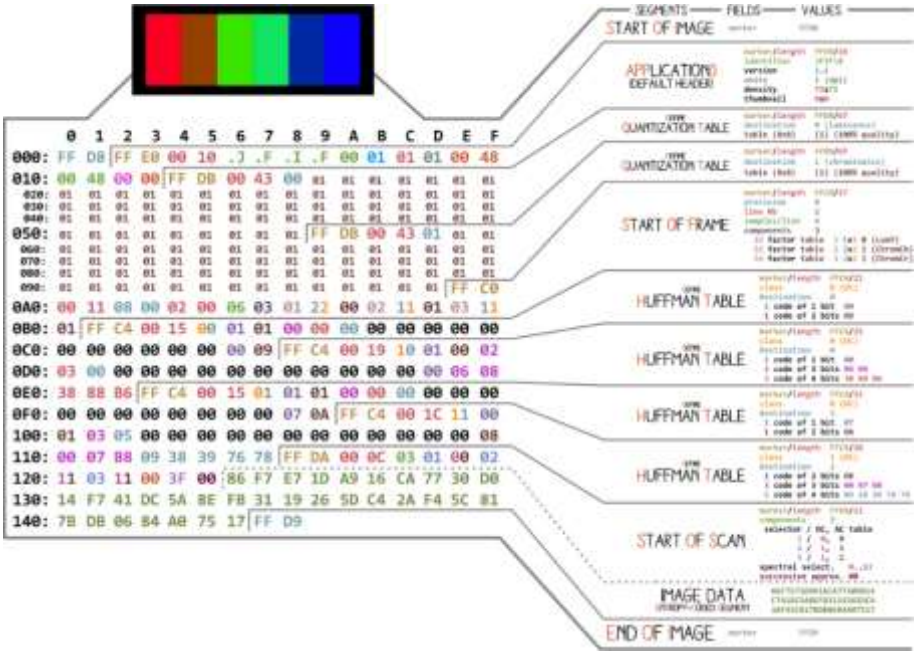


Рисунок 1 – Структура фомату JPEG

Майже кожен двійковий файл містить декілька маркерів (або заголовків). Можете вважати їх свого роду закладками. Вони вкрай важливі для роботи з файлом і використовуються такими програмами як file (на Mac і Linux), щоб ми могли дізнатися подробиці про фото. Маркери вказують, де саме в файлі зберігається певна інформація. Найчастіше маркери розміщуються відповідно до значення довжини (length) конкретного сегмента.

Стиснення даних JPEG (Joint Photographic Experts Group), що дозволяє стискати окремі (нерухомі, нерухомі зображення) зображення, можна розділити на три етапи [3]:

1 етап - трансформація та відбір кольорової інформації;

2 етап - блок дискретних косинусних перетворень (ДКП);

3 етап - квантування та кодування дискретних значень ДКП.

В результаті аналізу файлів JPEG було визначене, що структура JPEG- файлу нерегулярна. І хоча проаналізована вище структура маркерів відповідає в тому або іншому ступені будь-яким JPEG- файлам, але проте будь-якому узятому окремо JPEG- файлу властиві деякі відмінності. Наприклад, в якомусь JPEG- файлі визначена 1 таблиця квантування, і тому 1 блок даних сканування, в іншому файлі визначені декілька таблиць і декілька

блоків. Іноді з'являються JPEG- файли з маркерами перезапуску, що ускладнюють аналіз з поелементним розбором JPEG- структури. Крім того, JPEG- файл іноді зберігає зменшені копії зображень, призначених для попереднього перегляду. В цьому випадку розростається число сегментів даних зображення, що збільшує час при аналізі JPEG- файлу [4].

Отже, базуючись на розглянутій структурі формату JPEG, можна зробити висновки, що є маркери, що визначають сегменти, але що не беруть участі в JPEG- перетворенні. А тому вони не впливають на візуалізацію зображення. І природно вони ігноруються програмою перегляду. Перерахуємо їх:

1. COM;
2. APP15;
3. DAC;
4. DNL;
5. SOF2 - SOF10;
6. Неспецифіковані сегменти.

Розроблений алгоритм, використовуючи специфіку формату для приховування використовує перераховані маркери. Атже фрагменти, які позначаються перерахованими маркерами, дозволяють записувати певні дані. Але необхідн враховувати обмеженість об'єму сегменту, який задається двома байтами - OxFFFF.

Розглянемо основні етапи розробки даного методу:

Згідно до вимог, до впровадження приховані данні мають бути зашифровані і стиснуті. Потім необхідно врахувати параметри впровадження і об'єм стегоконтейнера.

Так як у форматі JPEG реалізовано стиснення з втратами, то потрібно реалізувати ряд попереджувальних заходів при впроваджені прихованих данних, щоб їх вберегти від спотворення [5].

Спочатку відбувається перетворення JPEG- файлу у BMP-файл (потік даних). В результаті відбувається збільшення розміру потоку даних через зміни кодування інформації про колір різних ділянок початкового зображення. Але з огляду на те, що в BMP- форматі піксель закодований 3 байтами, що відповідають за вклад основних кольорів (R - червоного, G - зеленого і B - синього) в підсумковий колір пікселя, розмір потоку збільшується досить помітно і тому дає можливість впровадження великого об'єму інформації.

Структура алгоритму запису прихованих даних зображено на рис. 2

З метою мінімальної зміни просторової області, реалізована стеганосистема за умовчанням використовує тільки молодший біт такого байта. Це дозволяє отримати мінімальну вірогідність виявлення детектування навіть на зображеннях з великою площею заливки синього кольору.

Рисунок 2 - Алгоритм запису прихованих даних

Отже, розроблений алгоритм орієнтований для вирішення завдання прихованої передачі даних. Він, як і будь-який інший стеганографічний алгоритм може виконувати процедури впровадження/витягання інформації. Такою інформацією бувають ідентифікаційні номери, ЦВЗ і так далі. Істотною і дуже корисною властивістю є автоматизація функціональності по впровадженню/вбудовуванню і дружність програмної реалізації стеганоалгоритма по вхідних/вихідних параметрам.

Перелік посилань

1. Юдін О. К. Аналіз стеганографічних методів приховування інформаційних потоків у контейнери різних форматів / О. К. Юдін, Р. В. Зюбіна, О. В. Фролов // Радиоэлектроника и информатика. — Х. : НХНУРЕ, 2015. — № 3. — С. 24-31.

2. Steganography and Digital Watermarking: a global view [Електроний ресурс] - Режим доступу до ресурсу: http://lia.deis.unibo.it/Courses/RetiDiCalcolatori/Progetti00/fortini/proiect.pdf.

3. LSB стеганография [Електроний ресурс]. - 2019. - Режим доступу до ресурсу: https://habr.eom/ru/post/112976

4. Рейда О.В. Аналіз та дослідження форматних стеганоалгоритмів на основі графічних контейнерів / Рейда О.В., Джулій В.М. // Тези доповідей Всеукраїнської науково-практичної конференції "Інтелектуальний потенціал – 2018". – 2018 – С. 86-90.

5. Cristi Cuturicu, JPEG - Алгоритм стиснення, Code Net [Електронний ресурс] / Формати файлів, - Режим доступу: http://www.codenet.ru/progr/forrnt/jpeg_00.php

Метод створення віртуальних полігонів на основі технологій хмарних обчислень системи управління базами даних

Джулій В.М., Лукін В.С., Чешун В.М.
Хмельницький національний університет

При виконанні дослідження було поставлено наступні задачі:

– дослідження і вибір існуючих систем, придатних для реалізації цілей і задач цієї роботи;

– проектування апаратно-програмного комплексу, включаючи дослідження і побудову всіх його підсистем;

– створення працюючого прототипу апаратно-програмного комплексу;

– визначення ефективності прототипу.

В рамках поставлених завдань розроблені наступні підсистеми апаратно-програмного комплексу та забезпечено їх взаємодію для виконання цілей цієї роботи:

– обчислювальна підсистема (система віртуалізації);

– мережева підсистема;

– система зберігання даних;

– система автоматизації надання послуг та забезпечення універсального доступу.

Для вирішення поставлених завдань розроблена архітектура інфраструктури віртуальних полігонів.

Робочі сервери віртуалізації [1,2] з встановленим гіпервізором є обчислювальним ядром інфраструктури, що забезпечує створення і управління роботою віртуальних машин, диспетчеризацію розподілу ресурсів між віртуальними машинами. Робочі сервери об'єднуються в ресурсний пул для забезпечення еластичності інфраструктури, можливості масштабувати інфраструктуру прозоро для користувачів. У ресурсному пулі виділяється керуючий master-сервер, за допомогою якого здійснюється централізоване адміністрування як всією обчислювальною системою, так і управління системою зберігання даних і мережевий підсистемою. Так само, master-сервер є центральним інтерфейсом управління всією системою за допомогою наданого їм API.

Обчислювальна підсистема забезпечує наступну функціональність: забезпечення продуктивності всіх компонентів віртуальних машин на рівні продуктивності пристроїв фізичних серверів;

– можливість створення готових шаблонів віртуальних машин з попередніми налаштованими пакетами програмного забезпечення;

– об'єднання фізичних серверів в ресурсні пули для динамічного розподілу віртуальних машин між фізичними серверами;

– можливість динамічної міграції віртуальних машин між серверами;

– підтримка режиму паравіртуалізації для найбільш ефективного використання обчислювальних та інших ресурсів операційними системами сімейства Linux.

Оскільки обчислювальна система є ядром всієї інфраструктури віртуальних полігонів, деякі вимоги до інших підсистем продиктовані функціональними можливостями обраного рішення віртуалізації.

Додавання нових серверів, а також штатне або аварійне вимкнення наявних серверів, відбувається прозоро для адміністратора системи і головне - для користувачів. Операція не вимагає переналаштовування системи і будь-яких додаткових дій з боку людини, не призводить до втрати інформації і збоїв в роботі інфраструктури. При цих діях користувач або взагалі не помічає змін, що відбулися, або перерви в роботі сервісів мінімальні.

Така можливість є суттєвою з точки зору хмарної моделі надання послуг, тому що забезпечує об'єднання ресурсів в ресурсні пули для динамічного перерозподілу потужностей між користувачами в умовах постійної зміни попиту на потужності, функції динамічної міграції віртуальних машин при проведенні технічного обслуговування окремих серверів, балансування навантаження між робочими серверами.

В архітектурі системи дана вимога виражається в необхідності використання зовнішнього сховища даних для зберігання файлів віртуальних машин, шаблонів віртуальних машин, віртуальних жорстких дисків користувачів. Гіпервізор повинен підтримувати роботу не тільки з локальною

системою зберігання даних робочого сервера, але і з зовнішніми системами зберігання даних.

Мережева підсистема забезпечує створення довільних топологій мережевої інфраструктури віртуальних машин, мережеву зв'язаність віртуальних машин як в межах внутрішніх віртуальних мереж між віртуальними машинами, так і відносно зовнішніх до системи мереж і пристроїв, нормальне функціонування віртуальних машин з точки зору мережевих протоколів.

Віртуальні машини, розгорнуті на різних фізичних серверах, мають можливість бути об'єднаними в єдиний віртуальний домен комутації на рівні L2. Для забезпечення вимоги до високої еластичності інфраструктури при реалізації мережевої підсистеми максимально використовуються програмні мережеві рішення – вбудовані можливості гіпервізора і стандартних мережевих засобів операційної системи Linux. З апаратних рішень використовується тільки фізичний комутатор рівня L2 для підключення серверів до локальної обчислювальної мережі.

Технічне управління і адміністрування системою в режимі нормальної експлуатації здійснюється з окремої програми з графічним інтерфейсом користувача, встановленої на керуючому сервері адміністратора системи. Адміністраторів системи може бути будь-яка кількість, так само можливо наділяти користувачів розширеними правами на управління частинами віртуальної інфраструктури. У цій програмі існує можливість створювати всі типові настройки, шаблони віртуальних машин, управляти життєвим циклом віртуальних машин, здійснювати адміністрування віртуальних машин, в тому числі в режимі консолі.

Конфігурація робочих серверів віртуалізації так само має здійснюватися підключенням до них безпосередньо по SSH через інтерфейс командного рядка (CLI). Звичайні користувачі отримують параметри доступу до віртуальних машин, а так доступ до консолі віртуальної машини через веб-сервер (портал) управління. Портал доступний як всередині локальної мережі організації, так і через мережу Інтернет.

Веб-сервер написаний на мові програмування Java і релизована на стандартній платформі, контйнері сервлетів, що підтримує функціонал віддалених процедур XML-RPC. Необхідний фунцонал веб-сервера забезпечується з допомогою APImaster-сервера.

Стосовно запропонованих рішень проведене дослідження підсистем інфраструктури віртуальних полігонів і їх взаємодії для того, щоб прийти до найбільш оптимальної схеми побудови всієї системи.

Проведені дослідження стали основою для створення віртуальних полігонів на основі технологій хмарних обчислень системи управління базами даних.

Перелік посилань
1. Michelle Bailey. The Economics of Virtualization: Moving Toward an Application- Based Cost Model. IDC.URL: http://www.vmware.com/files/pdf/Virtualization-
2. Peter Mell, Timothy Grance. The NIST Definition of Cloud Computing. NIST Special Publication 800-145. URL: http://csrc.nist.gov/publications/nistpubs/800- 145/SP800-145.pdf

Метод захисту від загрозливих програм, заснований на реалізації контролю доступу до файлових об'єктів

Казіміров В.О., Мостовий С.В., Нагребецький О.В., Орленко В.С.
Хмельницький національний університет

Використання сучасних систем інформаційної безпеки вимагає, з одного боку, відстеження швидких змін в інформаційних технологіях та нових загроз, а з іншого - з урахуванням реальних характеристик апаратного та програмного забезпечення корпоративних мереж та систем. Процедура придбання пристроїв захисту інформації проста. Набагато складніше вирішити проблему - як захистити і які заходи безпеки застосовувати, мінімізуючи витрати. Впроваджуючи різні засоби захисту, необхідно визначити баланс між можливим збитком від несанкціонованого витоку інформації та обсягом інвестицій, які витрачаються на забезпечення безпеки інформаційних ресурсів. З метою підвищення ефективності захисту інформаційних ресурсів необхідно дослідити підходи до оцінки рівня їх захисту та систем захисту. Ця оцінка для кожного випадку індивідуальна і залежить від багатьох факторів (вартості інформації, статусу організації, важливості інформації, рівня технічного та програмного забезпечення тощо).

В роботі здійснено дослідження основних типів загрозливих програм, та запропоновано класифікацію шкідливого програмного забезпеченні (ШПЗ) за способом їх виконання. Враховуючи аналіз існуючої статистики зроблено висновок, що найбільш актуальними для захисту є виконувані двійкові і файли сценаріїв.

Можна виділити два найбільш поширених способи зараження: соціальна інженерія; технічні прийоми впровадження ШПЗ, що заражається без відома користувачf [1].

Ці види ШПЗ передбачають обов'язкове збереження файлу на вінчестері перед виконанням.

Тому можна зробити висновок що застосування розмежувальної політики доступу до виконуваних об'єктів, дозволяє мінімізувати загрози.

Проведено дослідження існуючих підходів до оцінки ефективності методів і засобів захисту від загрозливих програм, в результаті якого зроблені

висновки про неможливість з використанням відомих підходів ні кількісно оцінити актуальність окремої загрози для інформаційної системи в цілому (з урахуванням безлічі інших потенційних загроз), в тому числі загрози занесення і запуску загрозливих програм, ні кількісно оцінити основні стохастичні характеристики безпеки системи від загрозливих програм. В результаті чого сформульована задача розробки відповідних математичних моделей і наступного проведення на них досліджень, що дозволяють отримати необхідні кількісні оцінки.

Бінарні та скриптові виконувані файли будуть діяти як об'єкти доступу. Розглянемо варіант, коли всі користувачі мають однакові права доступу.

У Windows найпоширеніші двійкові виконувані файли. Найпоширеніший їх тип - аплікація. Додатки мають розширення EXE і можуть працювати самостійно. Крім того, існують динамічні бібліотеки (їх розширення - DLL), які містять загальні функції для різних додатків. Існують також драйвери (DRV або VXD) - спеціальні програми, необхідні для того, щоб система могла взаємодіяти з конкретними моделями певних пристроїв. Виконувані файли (особливо в Windows) можуть залежати один від одного: наприклад, для запуску будь-якої програми потрібні певні системні динамічні бібліотеки, а вони, в свою чергу, потребують драйверів [2].

Слід зазначити, що виконувані файли містять не тільки самі програми, але і різні додаткові дані. Це можуть бути різні графічні ресурси, що відображаються програмою, тексти написів, описи діалогових вікон тощо. Яскравим прикладом цього є архіви, які містять великі обсяги упаковки з метою зменшення її обсягу при передачі або зберіганні інформації [3].

Отже суб'єктом доступу є будь-який користувач системи $S_i : S = \{S_1, \dots, S_k\}$. Об'єкти доступу поділяються на виконувані, системні та інформаційні: $O = \{ O_{вик1}, \dots, O_{викq}, O_{сист1}, \dots, O_{системm}, O_{інф1}, \dots, O_{інфn} \}$.

Системні файли мають розширення: *.config, *.manifest, *.fon, *.ttf, *.log.

Ми додамо захист від виконуваних файлів сценарію. Для їх виконання потрібно встановити інший інтерпретатор, в якому запуститься виконуваний файл. Основою моделі захисту від виконуваних файлів сценарію є захист від витоку повноважень на зміну функцій дозволеної програми. Особливістю методу є поділ методу доступу "Запис" для створення нового об'єкта доступу («*ЗпН*») та зміни існуючого об'єкта доступу шляхом перейменування («*ЗпІ*»). Введемо додатково право доступу - заборона читання («*Чт*») [4].

Тоді в моделі будемо використовувати таку множину прав:

$$R = \{ Чт, \ Чт \ , В, Зп, ЗпН, ЗпІ, Пв, ПвВ, Д \}.$$

Права адміністратора:

- читання, інсталювання і запуск виконуваних файлів;
- читання, інсталювання системних файлів;
- читання і запис інформаційних файлів;
- все інше забороняти.

Опишемо права інших користувачів:

- читання і запуск виконуваних файлів, які вже знаходяться на системному диску;
- заборонити створення нового файлу, зміни існуючого, перейменування існуючого виконуваного файлу, видалення існуючих виконуваних файлів на системному диску;
- читання системних файлів;
- заборонити для системних файлів створення нового, зміну та перейменування існуючого файлу;
- читання і запис інформаційних файлів;
- заборонити запис, перейменування і видалення системних файлі;
- заборонити перейменування і видалення інформаційних файлів;
- заборонити для інформаційних файлів перейменування існуючого файлу, видалення;
- все інше заборонити.

$$
M = \begin{array}{c|ccc}
 & O_{вик1}, \dots, O_{викq} & O_{сист1}, \dots, O_{системm} & O_{інф1}, \dots, O_{інфn} \\
S_1 & Чт, В, \sout{ЗпН}, \sout{ЗпІ}, & Чт, \sout{ЗпН}, \sout{ЗпІ}, & Чт, Зп, \\
\dots & \sout{Пв, ПвВ, Д} & \sout{Пв, ПвВ, Д} & \sout{Пв, ПвВ, Д} \\
\dots & & \dots & \\
S_k & Чт, В, \sout{ЗпН}, \sout{ЗпІ}, & Чт, \sout{ЗпН}, \sout{ЗпІ}, & Чт, Зп, \\
 & \sout{Пв, ПвВ, Д} & \sout{Пв, ПвВ, Д} & \sout{Пв, ПвВ, Д} \\
VM_1 & \sout{Чт}, В, \sout{ЗпН}, \sout{ЗпІ}, & Чт, \sout{ЗпН}, \sout{ЗпІ}, & Чт, Зп, \\
\dots & \sout{Пв, ПвВ, Д} & \sout{Пв, ПвВ, Д} & \sout{Пв, ПвВ, Д} \\
\dots & & \dots & \\
VM_j & \sout{Чт}, В, \sout{ЗпН}, \sout{ЗпІ}, & Чт, \sout{ЗпН}, \sout{ЗпІ}, & Чт, Зп, \\
 & \sout{Пв, ПвВ, Д} & \sout{Пв, ПвВ, Д} & \sout{Пв, ПвВ, Д} \\
A & Чт, Зп, В & Чт, Зп & Чт, Зп
\end{array}
$$

Запропоновано модель захищеної системи, яка дає можливість сформувати вимоги з точки зору запобігання витоку прав доступу.

Такий підхід дає нам впевненість що такі права доступу не дадуть змогу бінарним і скриптовим виконуваним загрозливим файлам зашкодити

безпеці системи.

Але потрібно забороняти співпадання будь-якого користувача з Адміністратором [5] .

Отже в роботі досліджено існуючі способи впровадження та запуску загрозливих програм, в результаті чого зроблено висновок про те, що найбільш актуальними для захисту є виконувані бінарні і скриптові файли і про те, що дані класи загрозливих програм передбачають обов'язкове збереження загрозливого файлу на жорсткому диску перед його виконанням (читанням). Це дозволило зробити висновок щодо того, що захист від загрозливих програм може будуватися реалізацією контролю (розмежування прав) доступу до файлів.

Запропоновано загальний підхід до реалізації захисту від загрозливих програм, заснований на реалізації контролю доступу до файлів по їх типам, які можуть бути ідентифіковані розширеннями файлів. Можливість використання подібного підходу обґрунтована проведеним дослідженням засобів захисту.

Розглянута модель засобів захисту дозволяє сформулювати вимоги до експлуатаційних параметрів засобів захисту - до інтенсивності виявлення помилок засобами захисту, що дозволяє здійснити успішну атаку, та інтенсивності їх виправлення, виконання яких забезпечує необхідні значення експлуатаційних характеристик засобів захисту.

Перелік посилань

1. Лебедев А. Защита компьютера от вирусов, хакеров и сбоев / Алексей Лебедев. – М.: Питер, 2013. – 157 с.

2. Исполняемые файлы: расширения, форматы. // Справочник типов файлов. [Електронний ресурс]. Режим доступу: http://open-file.ru/types/executable/, свободный (10.10.2020).

3. *Cuff P.* Distributed channel synthesis / P. Cuff // IEEE. Trans. Inf. Theory. – 2013. – Vol. 59(11). – P. 7071-7096.

4. Джулій В.М. Оцінка актуальності загрози впровадження загрозливих програм для інформаційної системи / В.М. Джулій, В.О. Бойчук, О.О. Кушнерик, -Хмельницький: Наука й економіка, 2018. - Вип. № 2. – С.107-115

5. Schieler C. Rate-distortion theory forsecrecy systems / C. Schieler, P. Cuff // IEEE Trans. on Inf. Theory. – 2014. – Vol. 66(12). – P.7584-7605.

6. Тарнавський Ю. А. Технології захисту інформації: підручник / Ю. А. Тарнавський ; КПІ ім. Ігоря Сікорського. – Київ : КПІ ім. Ігоря Сікорського, 2018. – 162 с.

7. Остапов С. Е. Технології захисту інформації : навчальний посібник / С. Е. Остапов, С. П. Євсеєв, О. Г. Король. – Х. : Вид. ХНЕУ, 2013. – 476 с.

Імітаційна модель для дослідження хаотичної синхронізації нелінійних динамічних систем

Ковбасовська Н.В., Грищенко В. Ю.
Науковий керівник: к.т.н. доц. Пивовар О.С.
Хмельницький національний університет

До недавнього часу в рамках дослідження синхронізації розглядались в основному процеси взаємного впливу гармонічних осциляторів. Із розвитком теорії динамічного хаосу (ДХ) було знайдено велику кількість нелінійних динамічних систем (НДС) що демонструють хаотичну синхронну поведінку у зв'язаних системах – хаотичну синхронізацію (ХС) [1].

Особливу роль під час дослідження ХС відіграють діючі на систему шуми – стохастичні завади. В класичних системах шум відіграє деструктивну роль та приводить до втрат інформації, а базовий спосіб боротьби з такою завадою – кореляційна обробка. В НДС шум може відігравати не тільки деструктивну, але і конструктивну роль, наприклад, загальний шум може призвести до ХС НДС систем, що слабко взаємодіють між собою. Такий різновид називають ХС, індукованою шумом.

Під час розгляду ХС також вживають термін «хаотичний синхронний відгук», що застосовується для хаотичних осциляторів (ХО) які можуть бути представлені у вигляді кільцевої структури (рисунок 1), де кільце зворотного зв'язку веденого ХО розривається і в певній пропорції додається сигнал ведучого ХО. При цьому виділяють ряд режимів ХС: повну, лаг, узагальнену, частотну, фазову, часткову, масштабування (рисунок 1) [2].

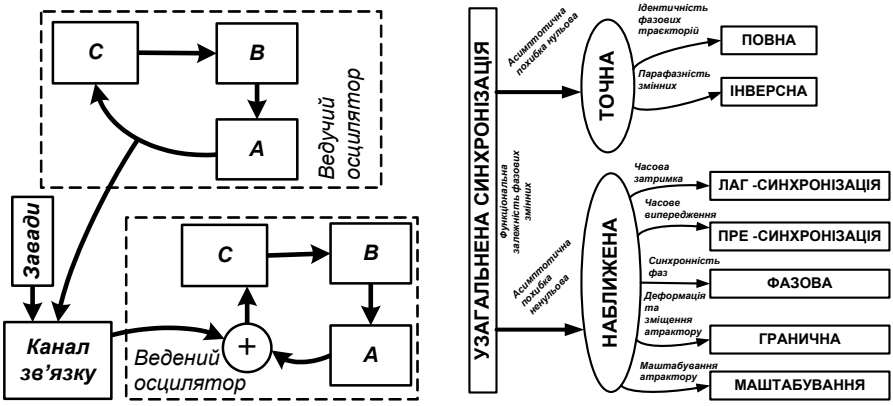


Рисунок 1 – Декомпозиція хаотичних генераторів для отримання хаотичного синхронного відгуку (ліворуч) та класифікаційні ознаки та варіанти узагальненої хаотичної синхронізації

Режим узагальненої синхронізації означає, що після завершення перехідних процесів між станами ведучої та веденої НДС в режимі генерації хаотичних сигналів між ними встановлюється певний чіткий аналітичних зв'язок, виявити який іноді досить важко, а іноді зв'язок може бути і фрактальним. Із загальної точки зору всі різновиди ХС зводяться до узагальненої ХС, в залежності від того, в якому аспекті розглядати функціональну залежність між фазовими змінними ведучої та веденої НДС.

Якщо розглядати функцію в узагальненій ХС в аспекті забезпечення асимптотичної точності синхронізації в реальних ділянках часу, то всі різновиди ХС можливо поділити на точні та наближені (рисунок 1). Для практичного втілення найбільш часто застосовують режим повної синхронізації, що забезпечує тотожність змінних ГДХ на обох боках системи (одинична функція зв'язку).

Діагностувати наявність узагальненої ХС можливо способом допоміжної системи [3], імітаційна модель якої розроблена авторами в середовищі Simulink (рисунок 2). Для реалізації такого способу ідентифікації наявності повної ХС між двома ХО застосовують дві НДС із однаковими операторами еволюції та початковими параметрами, що називають веденими ХО із декомпозицією (рисунок 1), сигнал на які від ведучого ХО подається одночасно та в однаковій пропорції із власним сигналом зворотного зв'язку.

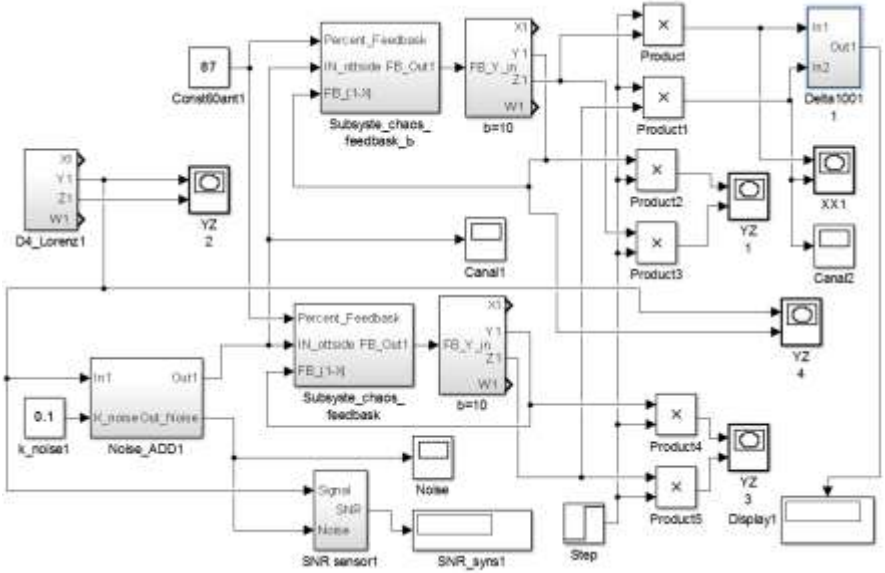


Рисунок 2 - Модель Simulink для дослідження способів хаотичної синхронізації генераторів високої розмірності

З точки зору математичної моделі без впливу шуму, під час запуску динаміки, обидва ведені ХО мають демонструвати ідентичну хаотичну поведінку. На практиці ж ми маємо ситуацію неможливості забезпечення однаковості функціонального опису та початкових параметрів ХО і під час відсутності сингалу синхронізації фазові траєкторії ведених ХО розбігаються, що призводить до декореляції їх часових залежностей.

Для фіксації рівня розбіжності фазових траєкторій можливо застосувати міру [2]: метричної або абсолютної відстані, фазового співпадіння, об'єму атрактора, Ляпунова, модифікованої системи, тощо. В переважній більшості в практичних схемах ХС застосовують міру середньо-квадратичного відхилення однієї фазової змінної.

Ідентифікація режиму ХС в імітаційній моделі може відбуватись на основі аналізу часового ряду однієї фазової змінної, а синхронізація відбуватись по іншій фазовій змінній. Не виключена можливість також роботи по якійсь одній фазовій змінній, або паралельна як синхронізація та аналіз по декільком фазовим змінним.

Імітаційна модель (рисунок 2) має у своєму складі субмодулі, що забезпечують: формування типу каналу зв'язку, генерацію завад певного типу, вимірювання відношення сигнал-завада, підключення ХО із різними операторами еволюції та використання різних алгоритмів встановлення режиму хаотичної синхронізації.

Розроблена імітаційна модель під час використання дозволила зробити ряд висновків та рекомендацій щодо процесу ХС для НДС, що описуються системою диференційних рівнянь:

1. Чутливість появи ХС до зміни біфуркаційних параметрів різко зростає із зростанням кількості рівнянь в системі від 5% для системи 3-го порядку до 0,1% для системи 4 порядку.
2. Чутливість появи ХС зростає із зростанням порядку нелінійності для НДС із поліноміальною нелінійністю.
3. Поява ХС індукованої шумом для НДС високого порядку різко обмежує можливості застосування методу допоміжної системи для виявлення ХС.
4. Під час використання ХО із великою кількістю біфуркаційних параметрів, найбільш доцільно для встановлення ХС застосовувати одночасну варіацію декількох параметрів.

Перелік посилань

1. Прикладне застосування теорії хаотичних систем у телекомунікаціях : монографія / Ю. Я. Бобало, С. Д. Галюк, М. М. Климаш, Р. Л. Політанський; Міністерство освіти і науки України, Національний університет "Львівська політехніка". – Львів; Дрогобич : Коло, 2015. – 184 с.

2.Boccaletti S. The synchronization of chaotic systems / S. Boccaletti, J. Kurths, G. Osipov, D. L. Valladares, and C. S. Zhou // Physics Report. – 2002. – Vol. 366. – № 1–2. – P. 1–101.

3. Галюк С. Д. Особливості синхронізації хаотичних систем (огляд) // С.Д. Галюк., Л.Ф. Політанський, М.Я. Кушнір, Р.Л. Політанський // Складні системи і процеси. – 2011. – №2. – С. 3–29.

Модель програмного комплексу для реалізації методу інтерактивного групового навчання

Машевський В. О.
Науковий керівник – к.т.н., доцент Яшина О. М.
Хмельницький національний університет

В сучасних умовах бурхливо розвиваються інтерактивні технології, зокрема й в освіті, що надає більших можливостей ефективно і якісно організовувати освітній процес. Враховуючи сучасні виклики та реалії, які постають перед освітою в період пандемії COVID-19, питання створення та реалізації методів інтерактивного групового навчання в різних галузях науки в цілому, та з програмування зокрема, стає дедалі актуальнішим.

Поява різноманітних спеціалізованих додатків для навчання останнім часом стала розглядатись, як можливість використання таких мобільних засобів в загальноосвітньому процесі. Аналіз та оцінка сучасних світових тенденцій демонструє життєву необхідність застосовування в навчальній діяльності мобільних засобів для вирішення різноманітних педагогічних задач, організації віддаленого доступу до загальномережевих і спеціалізованих ресурсів та сервісів.

Об’єктом даного дослідження є інтерактивне навчання із застосуванням мобільних технологій. Предмет дослідження – моделі та механізми створення зручного методу групового навчання основ програмування.

З психологічної точки зору інтерактивним навчанням називають спеціальний тип організації освітнього процесу, який здійснюється у форматі спільної діяльності студентів, що дає можливість всім учасникам взаємодіяти один з одним, обмінюватися інформацією, спільно вирішувати проблеми, моделювати ситуації, оцінюють дії інших, занурюватися в атмосферу так званого ділового співробітництва з вирішенням поставлених викладачем проблем. Проте саме поняття «інтерактивність» прийшло з лексики комп'ютерних технологій.

На даний момент не представлена єдина класифікація інтерактивних технологій, однак, розглядаючи їх з позиції форм навчання, можна виділити наступні типи:

а) кооперативне навчання;

б) колективно-групове навчання;

в) ситуативне моделювання;

г) відпрацювання дискусійних питань.

Формула моделі інтерактивного групового навчання має наступний вигляд:

$$S_1 \leftrightarrows (S_{21} \leftrightarrows S_{22} \leftrightarrows S_{23} \dots)$$

де S_1 – викладач, а $S_{21,}S_{22,}S_{23}$ … - студенти

На основі вищезазначеної інформації можна сформулювати основні вимоги до моделі розроблюваного комплексу:

- можливість публікації статей для певної групи студентів;
- реалізація тестування серед учасників освітнього процесу;
- проведення вікторини та підсумкових письмових заходів.

Отже, система може використовуватись двома групами людей. Це студенти та викладач. Головним робочим напрямком для викладача є проведення інтерактивних занять, а для студентів – участь в даному навчальному процесі. Для кожної групи користувачів було сформовано функціональні вимоги. Вимоги для викладача виглядають наступним чином:

- створення, редагування та публікація статей в чаті;
- створення та редагування запитань тестів та вікторини з варіантами відповіді по категоріях;
- можливість проведення підсумкових заходів та прийом письмових робіт в електронному форматі;
- можливість проведення ігрової вікторини на випередження.
- контроль за успішністю учасників;
- модерація чату.

Вимоги для студентів:

- проходження тестування та вікторини;
- перегляд статей
- завантаження готових письмових робіт через спеціальний режим чат-боту.

За нефункціональними вимогами система повинна мати можливість синхронізуватись з віддаленим сервером для коректної роботи адміністративної панелі викладача. Надалі розглянемо аналоги сервісів.

Серед великої кількості чат-ботів не було помічено аналогів, які би забезпечували вищезазначений функціонал. Такі веб-сервісів та застосунки, як Skype, Zoom, Microsoft Teams, Moodle дають змогу реалізувати методи інтерактивного навчання у повному обсязі.

Можливості Skype та Zoom дозволяють організовувати групове навчання за допомогою методу групових відеодзвінків. В даному випадку викладач проводить і класичне групове навчання і діалоги між окремими студентами з можливістю обміну повідомленнями у вбудованому чаті.

Microsoft Teams – це потужний інструмент та корпоративна платформа, яка об'єднує в робочому просторі чат, зустрічі, замітки і вкладення.

Teams дає можливість по-іншому поглянути на процес комунікацій під час групового інтерактивного навчання завдяки створенню команд і каналів всередині команд. Команда – це об'єднання людей і інструментів для організації спільної роботи над проектом, а канали – це виділені розділи всередині команди, що дозволяють організувати бесіди з певних тем. В рамках навчального процесу, можна сказати, що команди – це класи, а канали – це предмети в класах.

Moodle (Modular Object-Oriented Dynamic Learning Environment) – безкоштовна система електронного навчання. Це відкритий веб-додаток, на базі якого можна створити спеціалізовану платформу для навчання студентів або співробітників. Через систему електронного навчання Moodle можна навчати і тестувати учнів дистанційно. Важливу роль в платформі грають плагіни – модулі, які допомагають змінити дизайн і розширити функціональні можливості системи. Модулі розробляють учасники спільноти Moodle, і здебільшого вони в безкоштовному доступі. Зараз налічується близько 1500 плагінів.

Проаналізувавши вищезгадані сервіси, що забезпечують методи інтерактивного навчання, можна зробити висновок, що усі вони мають один суттєвий недолік – зручність використання, в деяких випадках до кінця не адаптований інтерфейс в залежності від пристрою, повільна робота на малопотужних мобільних девайсах. Так як Telegram є кросплатформеним додатком, то не має бути проблем з використанням чат-боту на всіх пристроях (смартфони, ноутбуки, планшети) на базі операційних систем iOS, MacOS, Linux, Android, Windows.

Діаграма варіантів використання буде виглядати наступним чином (рисунок 1)

При пошуку можливих рішень для написання чат-бота було встановлено, що для розробки бота підходить декілька мов серверного програмування: Python, Ruby, Node.JS, PHP. Так само важливо вміти працювати з REST (Representational State Transfer) API (Application Programming Interface), який надають месенджери, а саме Telegram Bot API.

Також потрібно спроектувати схему взаємодії користувача з базою даних, яка визначає як сервер має взаємодіяти з Telegram Bot API. (рисунок 2).

Отже, в ході роботи було проведено аналіз предметної області, актуальних технологій і програмних рішень та спроектовано модель, яка реалізовує метод інтерактивного групового навчання. Основні задачі, які будуть виконані розроблюваним комплексом, мають вдосконалити освітній процес під час реалізації колективно-групового методу інтерактивного навчання.

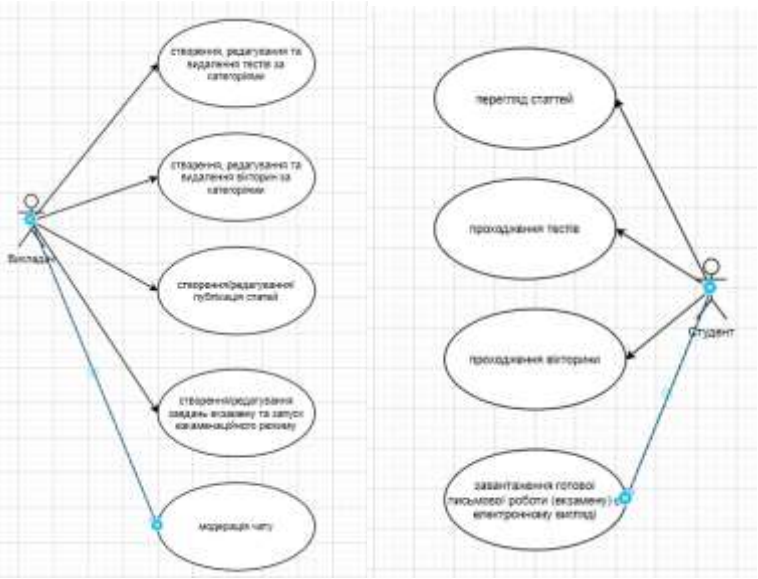


Рисунок 1 – Діаграма варіантів використання

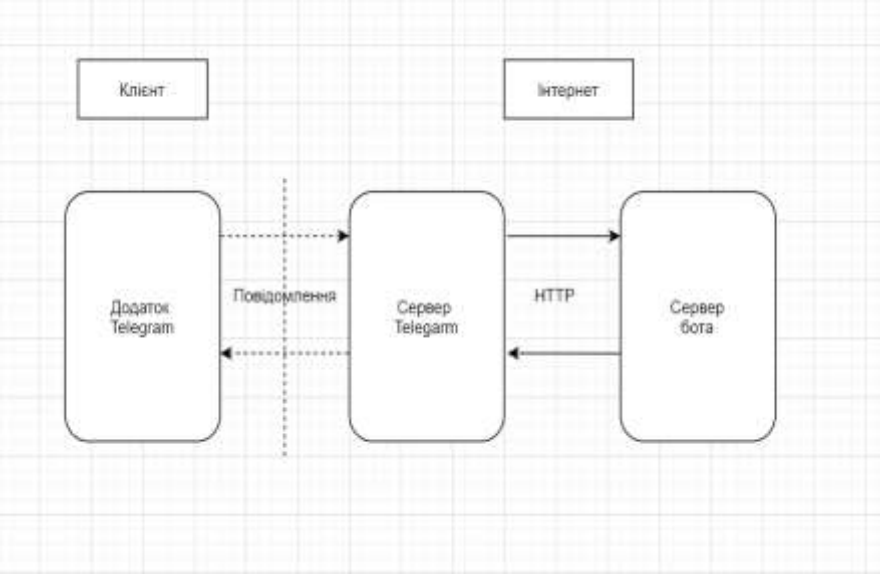


Рисунок 2 – Взаємодія сервера та Telegram Bot API

Перелік посилань

1. Антонов С. Что такое чат-боты и зачем они нужны? [Електронний ресурс] / Святослав Антонов. – 2018. – Режим доступу до ресурсу: https://informburo.kz/cards/chto-takoe-chat-boty-i-zachem-oni-nuzhny.html.
2. Гатулин Р. Р. Использование мессенджера Telegram для реализации технологии электронного обучения в вузе [Електронний ресурс] / Р. Р. Гатулин, Д. А. Колупаева // Науки об образовании. – 2017. – Режим доступу до ресурсу: https://cyberleninka.ru/article/n/ispolzovanie-messendzhera-telegram-dlya-realizatsii-tehnologii-elektronnogo-obucheniya-v-vuze.
3. Интерактивное обучение [Електронний ресурс]. – 2016. – Режим доступу до ресурсу: https://studme.org/157663/pedagogika/interaktivnoe_obuchenie.
4. Справочник по Bot API [Електронний ресурс]. – 2016. – Режим доступу до ресурсу: https://tlgrm.ru/docs/bots/api.
5. Можаров М. С. Использование современных технологий в области интерактивного обучения программированию: тенденции и перспективы [Електронний ресурс] / М. С. Можаров // Вестник ТГПУ. – 2017. – Режим доступу до ресурсу: https://cyberleninka.ru/article/n/ispolzovanie-sovremennyh-tehnologiy-v-oblasti-interaktivnogo-obucheniya-programmirovaniyu-tendentsii-i-perspektivy/viewer.

Вдосконалення методу проєктування вебдодатків на основі об'єктно-реляційного перетворення

Мілер В.М., Орленко В.С.
Хмельницький національний університет

З метою покращення розробки вебдодатків проектувальники намагаються знайти рішення, яке дозволить швидко і ефективно використовувати компонентну базу існуючих бібліотек.

Найпоширенішою платформою для розробки вебдодатків є LAMP [1]. LAMP - абревіатура набору вільного ПЗ з відкритим кодом, в який входять ОС Linux, веб-сервер Apache, СКБД MySQL, та інтерпретатор Perl/PHP/Python - основні компоненти для побудови життєздатного багатоцільового вебсервера

В процесі роботи пропонується метод, який здійснює реляційно-об'єктне та об'єктно-реляційне перетворення (ОРП) таблиць бази даних у копіях класів базових моделей програмного коду. В даний час є розробки подібних систем, таких як Doctrine, Propel [2], але вони мають суттєвий недолік - великий обсяг програмного коду і надлишок реалізованих функцій. Наприклад, Доктрина включає більше 100 класів. Це спричиняє проблему з продуктивністю, оскільки інтерпретатор мови повинен завантажити значну

кількість файлів. Крім того, на об'єкти, отримані за допомогою ОРП, існує ряд обмежень:

1) Вони можуть знаходитися тільки в одній таблиці. Таким чином, проєктування об'єкта додатка зводиться до композиції об'єктів ОРП, що збільшує трудомісткість розробки.

2) Вони знаходиться в єдиній базі даних. Це випливає з попереднього пункту. Неможливо створити об'єкт, розподілений по декільком БД.

3) Мають фіксований набір властивостей. Зокрема, кількість колонок в таблиці має дорівнювати кількості властивостей об'єкта;

4) Ускладнена багатомовна підтримка, оскільки формується єдина таблиця, в якій частина стовпців "відповідає" за різні мови, а після вибірки необхідно фільтрувати дані..

Враховуючи переваги використання систем PDO та виходячи з недоліків існуючих систем, необхідно сформулювати цілі розробленої системи PDO:

1) Адекватне відображення предметів та їх зв’язків. Основною функцією PDO є відображення об’єктів предметної області веб-програми в реляційній структурі бази даних, а також відображення взаємозв’язків об’єктів. Це може призвести до низки проблем. Наприклад, об'єкт може зберігатися в багатьох таблицях безлічі баз даних на різних серверах. Об’єкти можуть бути різними за своєю будовою. Наприклад, нові елементи в процесі розробки можуть бути додані до об'єкта профілю користувача, а нові властивості можуть з'являтися в об'єкті.

2) Підтримка багатомовності. При розробці веб-додатків часто доводиться підтримувати кілька мов. Багатомовна підтримка повинна бути прозорою, тобто розробник повинен бути впевнений, що об’єкт зараз відображається поточною мовою для користувача. Але в той же час система повинна мати достатню гнучкість, щоб змінити мову відображення на таку, яку вимагає розробник.

3) Розроблені можливості пошуку. Система повинна підтримувати складні запити, наприклад, "знайти електронний лист першого менеджера компанії, який розмістив останній продукт".

В рамках даного дослідження було запропоновано систему об'єктно-реляційного відображення «Активна модель», яка вирішує вищевказані завдання. Для здійснення реляційно-об'єктних -перетворень необхідно на стороні об'єктної моделі реалізувати декларативний опис реляційної моделі. Для цього зараз використовуються мови опису об'єктів (*ODL - Object Definition Language*). Вони призначені для таких потреб:

1) Визначення схеми бази даних.

2) Забезпечення універсальності опису схеми бази даних. Можна змінити базу даних на іншу, тоді як опис на ODL не зміниться. ODL використовується як "загальний знаменник" при описі схеми бази даних.

3) Трансформація типів даних, тобто генерування типів зі схеми бази даних певною мовою програмування, з якої планується доступ до неї.

У свою чергу, мови сімейства OQL (Object Query Language) використовуються для реалізації RO-перетворень. Це мови запитів об'єктів, декларативні мови доступу до бази даних, подібні до мови SQL для баз даних. Стверджується, що вирази в OQL на 90% сумісні з синтаксисом оператора select з SQL'92 [2]. Відмінності між OQL та SQL полягають у тому, що вхідними даними запитів OQL є об'єкти, а не таблиці. Конструкція select-from-where використовується для написання запиту, як у SQL. Результатом запиту, як правило, є набір об'єктів - набір. Тоді цей набір можна перетворити на список, масив. Набір може оброблятися в циклі та отримувати окремі його елементи - об'єкти бази даних повністю або набір значень з бази даних у вигляді будь-якого типу мови програмування. Таким чином, взаємна модельна трансформація складається з інтерпритації OQL. В існуючих системах (наприклад, в Doctrine) модельне перетворення включає в себе такі операції як парсинг OQL-запиту, валідацію, кешування, перетворення OQL в SQL, виконання SQL, отримання «сирих» даних, гідрацію (перетворення сирих даних в об'єктний вид). З цим пов'язані значні витрати обчислювальних потужностей сервера, тому в даній роботі пропонується ряд змін, спрямованих на збільшення продуктивності операцій перетворення:

1) Зведення ODL до діалекту основної мови розробки. Для досліджуваної платформи LAMP − це PHP.

2) Визначення кінцевої множини перетворюваних типів і їх відображень.

3) Використання діалекту основної мови платформи для здійснення OQL-операція, а інтерпретатора платформи − для інтерпретації OQL команд.

4) Визначення спеціальних типів таблиць і їх ODL.

Діалект МП PHP, який використовується в якості ODL, буде називатися AMDL (ActiveModel Definition Language), а діалект, який використовується в якості OQL − AMQL (ActiveModel Query Language). Результат модельної трансформації на стороні об'єктної моделі буде називатися AM (ActiveModel).

Перед визначенням AMDL і AMQL-діалектів необхідно визначити структури, які піддаються РО перетворенню. Як відомо, інформація в РСУБД організована у вигляді множини таблиць (сукупності схем відносин і даних). У даній роботі пропонується класифікація схем на три типи: T_C − звичайна (classic), T_P-схема додаткових полів (x-properties), T_F − схема прапорів (x-flags) і T_{FT} − схема прапорів (x-flags-temporary) з темпоральною валідацією.

Реляційна модель задається наступним чином. Нехай $A_1, A_2, ..., A_n$ імена атрибутів. Кожному імені атрибута A_i відповідає допустима множина значень, які може приймати атрибут A_j. Це множина значень D_i називається доменом атрибуту A_i, $i = 1, n$. За визначенням, домени є непорожніми кінцевими або зліченними множинами. Поняттю домену D_i відповідає множина значень, що знаходяться в стовпці A_i розглянутої таблиці [4].

Схемою відношення $R\{A_{R1}, A_{R2}, ..., A_{Rn}\}$ називається кінцева множина імен атрибутів $\{A_{R1}, A_{R2}, ..., A_{Rn}\}$, причому атрибут A_i, приймає значення з множини D_{Ri} (i = l, 2, ..., n), де n - розмірність відношення.

Об'єктна модель задається наступним чином. Об'єктом O називається представлення сутності предметної області, яке використовується при моделюванні. $A\ \{A_{O1}, A_{O2}, ..., A_{On}\}$ є множиною атрибутів об'єктної моделі. Класом C називається загальна сутність, яка може бути визначена як сукупність елементів (реалізацій класу). Клас є родовою ознакою об'єктів.

Після визначення операцій перетворення типу даних можна визначити набір операцій перетворення. Об'єктно-реляційна система відображення перетворює набір кортежів (записів, рядків) даних, що складають набір значень атрибутів, у форму, з якої можуть взаємодіяти функції мови програмування PHP. У запропонованому методі (і системі ORP "Активна модель") набір кортежів відображається на екземплярі класу. Об'єктом класу "Активна модель" є сукупність відносин, отриманих в результаті дії природного зв'язку, застосованих до безлічі кортежів (обов'язково по одному для кожного зв'язку) та набору функцій (поведінки) над сукупність відносин, успадкованих від класу програмного забезпечення. Натуральним зв’язком є операція SQL NATURAL JOIN, яка повертає відношення, яке містить усі можливі кортежі (K), які є комбінаціями двох (або більше) кортежів, що належать до двох (або більше) заданих відносин, за умови, що в комбінованих кортежах є однакові значення в одному (або декількох) загальних для вихідних атрибутів (і ці загальні значення з’являються в результуючому кортежі рівно один раз).

Основна відмінність методики «Активна модель» від інших методик в тому, що n (кількість відношень, які використовуються для відображення) може бути більше 1, тоді, як для існуючих методик об'єктно-реляційного відображення на платформі LAMP (маються на увазі методики Doctrine, Propel, Yii Active Record) $n = 1$. З цього випливає, що реалізація моделі даних, подібної до «Активної моделі», вимагає створення в них n моделей. Відповідно, вартість пам'яті, необхідної для зберігання моделей даних, збільшиться за рахунок збільшення кількості зразків класів.

Перелік посилань

1. Майк Кон. Scrum: Гибкая разработка ПО. / Майк Кон. — Изд-во: Диалектика-Вильямс, 2016. — С. 576.

2. Doctrine OPvM [Електронний ресурс]. – Режим доступу: URL: http://doctrine-project.org (дата звернення: 04.08.2017).

3. Роберт Мартин Гибкая разработка программ на Java и C++. Принципы, паттерны и методики. /Роберт С. Мартин, Джеймс Ньюкирк, Роберт Косс - Изд-во: Диалектика-Вильямс, 2016. - 704с.

4. Джулій В.М. Методи та алгоритми розробки web-додатків / В.М. Джулій, Ю.О. Гунченко, Д.В. Чешун // Збірник наукових праць Військового інституту Київського національного університету імені Тараса Шевченка. – К.: ВІКНУ, 2017. – Вип. № 56. – С.107-115

Дослідження проблем ідентифікації об'єктів в базах даних

Мозолюк В.О., Джулій В.М.
Хмельницький національний університет

На даний момент СУБД широко використовуються в організації сучасних інструментальних, промислових, аналітичних та інформаційних систем. Однак такий бурхливий розвиток інформаційних технологій баз даних поставило також ряд нових проблем і визначило напрямки подальших досліджень у цій області. Не припиняюча робота дослідників та аналітиків відноситься до питань оптимізації виконання запитів і структур зберігання даних, новітніх способів виконання реляційних операцій, організації пошуку, і багато інших моментів, що визначають результативність роботи СУБД. Програмне забезпечення на даний момент розвивається в умовах швидкого зростання обчислювальних потужностей, апаратних можливостей, швидкості доступу до пам'яті, обсягу пам'яті, пропускної здатності та надійності каналів передачі даних. Все більшого значення набувають засоби, що забезпечують взаємодію в розподіленої системі функціонування інформаційних систем.

Розглянемо більш докладно основні напрямки розвитку сучасних баз даних і СУБД:

1. Стандартизація мови SQL. У сучасних СУБД на даний момент основною мовою написання запитів і доступу до баз даних є мова SQL (Structured Query Language). Міжнародний стандарт даної мови розроблений в 1989 році, і більшість виробників СУБД призвели свої системи у відповідність даному стандарту. Потрібна постійна актуалізація мови SQL до мінливих вимог сучасних програмних продуктів та апаратних засобів.

2. Використання мультипроцесорних організацій. Промислові комерційні СУБД реалізуються на основі архітектури "клієнт-сервер". При даній організації всі операції над базами даних виконуються на сервері, що володіє достатньою продуктивністю і набором обчислювальних ресурсів. Після появи мультипроцесорних симетричних апаратних архітектур в

багатьох СУБД була переглянута організація серверних платформ і реалізована можливість розпаралелювання обчислень.

3. Інтеграція та інтероперабельність. Залишається актуальним рішення проблеми використання баз даних попередніх поколінь і версій. Прагнення до спрощення технологічних процесів і необхідність інтеграції інформаційних ресурсів призвели до розробки СУБД здатних підтримувати поряд зі структурованими даними також і текстові документи і виконувати їх пошук по запитах користувачів. Розвинені засоби текстового пошуку присутні в даний час в DB2 (IBM), Oracle, Microsoft SQL Server та ін.

Ефективність управління сучасним бізнесом заснована на можливості отримання управлінським персоналом всебічної інформації з усіх напрямків діяльності. При цьому важливо встановлення контролю над зростаючими потоками інформації, прискорення процесу їх обробки, пошуку та аналізу даних. Існуючі в даний час і розроблювальні нові автоматизовані системи характеризуються великою різноманітністю підтримуваних інформаційних ресурсів, способів організації даних, функціональними можливостями користувацьких інтерфейсів та інших їх технологічних характеристик. В розробках інформаційних систем даної категорії затребуваний практично весь спектр ключових технологій управління інформацією. В даний час багато організацій накопичили значні обсяги інформації. Внаслідок цього стає актуальною проблема розробки корпоративної системи управління знаннями. Однак серйозною перешкодою на даному шляху є розрізненість інформації в корпоративній системі, нерідко дані по одному напрямку діяльності зберігаються в різних додатках і форматах. Крім того, обсяги оброблюваної електронної інформації наростають по експоненті – цьому сприяє активне впровадження мультимедіа, широке поширення корпоративних і глобальних мереж, відхід більшості компаній від паперового документообігу та перехід на автоматизовані системи управління. В подібній ситуації значно зросла необхідність у створенні та впровадженні ефективних систем пошуку та аналізу даних. Традиційними є системи пошуку, що розвиваються в тісному взаємозв'язку з СУБД і в основному орієнтовані на роботу зі структурованими текстовими даними. Однак інтегровані в СУБД системи пошуку слабо адаптовані для обробки мультимедійної і неструктурованої інформації. За статистикою, частка структурованих даних в сучасних базах даних становить не більше 35-50%, решта ж припадають на частку різних довідників, сканованих документів і іншої розрізненої інформації. У цьому випадку виникає проблема пошуку і вибірки необхідної інформації з великого неструктурованого масиву.

Для багатьох організацій інформація є основним активом. Спотворення або пошкодження важливої інформації може призвести до суттєвих фінансових втрат і репутаційним ризикам. Аналізуючи дані, отримані з відкритих джерел і наукових публікацій, можна виділити основні

види втрат, що виникають внаслідок помилок і спотворень інформації в базах даних: втрати внаслідок невірного, поганого надання послуг («брак» в інформації). Даний вид втрат присутній майже в будь-якій організації. В середньому організація втрачає 25-40% часу співробітників, від втрат даного виду; втрати оплачуваного часу співробітників на непродуктивну діяльність. В тому чи іншому виді даний вид втрат зустрічається в будь якій організації, може досягати, наприклад, у менеджерів середньої ланки більше 50% робочого часу, у менеджерів низової категорії до 80%; втрати внаслідок використання «не оптимальних технологічних ланцюжків. Даний вид втрат присутній майже в будь-якій організації. За цими причинами в середньому організація втрачає близько 35% робочого часу задіяних співробітників і це може призвести до подорожчання однієї операції до 100%; втрати часу, грошових коштів, клієнтів по причині відсутності або дублюванні інформації. Даний вид втрат присутній майже в будь-якій організації. Втрати становлять близько 15% часу співробітників, що спричиняє збільшення вартості виконуваної операції.

Основним чинником, що стимулює розвиток технологій пошуку, є поява великої кількості електронних бібліотек і архівів, що містять значні обсяги актуальних знань. Продуктивність і ефективність будь-якої системи зберігання інформації безпосередньо залежить від ефективності та продуктивності пошукових систем. Саме пошукова система визначає, чи перетворяться в знання численні розрізнені дані, що надходять по різних каналах зв'язку і накопичуються в різноманітних базах даних та електронних архівах. Найбільш поширеним видом інформаційних ресурсів для організацій, що працюють з персональними даними (бюро кредитних історій, банки, страхові організації, будь-які організації з досить крупним штатом співробітників) є тексти на природних мовах. Цим обумовлено широке застосування в таких системах технологій текстового пошуку. Дані технології використовуються при цьому не тільки в системах, побудованих за принципом традиційних текстових систем, але і для пошуку в колекціях, організованих у вигляді веб-сайтів, а також для пошуку в глобальної мережі Інтернет.

При організації пошуку в базах персональних даних клієнтів виникають характерні проблеми, пов'язані з наявністю в запитах орфографічних і фонетичних помилок, помилок введення інформації, а також відсутністю єдиних стандартів транскрипції з іноземних мов. Внаслідок цього задача пошуку в базах персональних даних не може бути повною мірою вирішена тільки методами перевірки на точну відповідність. Стає актуальною задача розробки спеціальних методів і технологій текстового пошуку з використанням нетривіальних рішень, в тому числі на основі операцій несуворої відповідності. Однак універсальної методики пошуку в умовах зашумленості даних не існує, оскільки кожна проблема має власну

оригінальну специфіку. Для рішення виниклих проблем потрібно використовувати алгоритми здатні відшукати всі лексикографічно близькі до шаблону пошуку слова, що відрізняються замінами, пропусками і вставками символів. Таким чином, автоматично стає допустимою помилка, як у вхідних даних, так і в термінах запиту. В даний час можливості виконання пошуку за подібністю не використовуються в СУБД. Таким чином, виникає задача розробки алгоритмів виконання спеціальних реляційних операцій, що виникають в задачі ототожнення записів. Проведений аналіз напрямків розвитку сучасних баз даних показує, що склалися і формуються за останні роки тенденції розвитку інформаційних технологій істотно впливають, у тому числі і на функціональні можливості автоматизованих систем. Задача встановлення відповідності між окремими об'єктами - побудова процедур ототожнення в даний час не має задовільного рішення. Існуючі роботи, присвячені інтеграції БД, дозволяють здійснити тільки інтеграцію схем БД, але не пропонують способів побудови процедур ототожнення. Побудова процедур ототожнення ускладнюється відсутністю серед загальних атрибутів відповідних один одному таблиць різних БД первинних ключів і наявністю помилок операторського введення. Існуючі СУБД не пропонують можливості для використання пошуку за подібністю, що усуває викликані помилки операторського введення.

З урахуванням специфіки роботи з персональними даними пропонується вирішення наступних прикладних задач: повна ідентифікація клієнта при наявності спотворень інформації в базі даних або в пошукових запитах; усунення дублікатів записів при надходженні до БД з множинних джерел зі слабоструктурованою інформацією; пошук і коректування помилок в персональних даних клієнтів (фізичних і юридичних осіб).

В області технологій можна виділити появу принципово нових програмних засобів аналізу фрагментарної, слабоструктурованої, пошкодженої, нечіткої, неповної інформації. До таких засобів можна віднести технологію інформаційного моніторингу комплексних процесів і засоби Business Intelligence (бізнес аналітики). Використання принципово нового інструментарію на основі алгоритмів нечіткого пошуку в міжнародних компаніях і державних організаціях, великих корпораціях і фінансових установах показало їх ефективність і величезний потенціал для вирішення прикладних задач.

Перелік посилань

1. Васильєв В.І. Інтелектуальні системи захисту інформації: навч. посібник / В. І. Васильєв. - 2-е изд., Испр. - М .: Машинобудування, 2012. - 171 с.

2. Гордейчик С.В. Безпека бездротових мереж. / С.В. Гордейчик, В.В. Дубровін - М .: Гаряча лінія - Телеком, 2008. - 288 с.

3. Гузаіров М.Б. Управління захистом інформації на основі інтелектуальних технологій: навчальний посібник./ М.Б. Гузаіров, І.В. Машкіна - М .: Машинобудування, 2013. - 241 с.

Проблеми та перспективи побудови систем управління ресурсами інформаційних комунікаційних мереж

Просянюк В.В
Науковий керівник – д.т.н.,проф. Андрощук О.С
Хмельницький національний університет

Дослідження параметрів інформаційного потоку сучасних комунікаційних мереж показує, що припущення про справедливість властивостей процесів надходження та обслуговування потоків можуть реалізовуватися на практиці при малих навантаженнях, в умовах низького навантаження і невисоких швидкостях передавання потоків даних. Актуальними завданнями для дослідження та моделювання роботи інформаційних комунікаційних мереж є:

- моніторинг роботи інформаційної мережі на інфраструктурному, проміжному і базовому рівні моделі;
- аналіз потоку навантаження на кожному рівні, що враховує характер навантаження;
- синтез методів розрахунку ймовірнісно-часових параметрів мережевих вузлів при обробці потоку навантаження у мережевих пристроях із метою побудови адекватного прогнозу значень імовірнісно-часових параметрів для вироблення управляючих впливів на режим функціонування інформаційної мережі в реальному масштабі часу.

Для оцінки якості функціонування інформаційної комунікаційної мережі необхідно оперуватися набором критеріїв висунутих для забезпечення ефективного використання мережевих ресурсів. Це дасть змогу забезпечити низькі затримки обслуговування, високі пропускні здатності, захищеності даних в процесі передавання потоків інформації. Традиційний підхід побудови інформаційних комунікаційних систем та мереж, пов'язаний із чіткою регламентацією на всіх рівнях мережевої взаємодії, для гарантування високої якості обслуговування необхідно планувати інформаційну мережу із значним запасом ресурсів [1]. По скільки такий підхід викликаний властивістю само подібності потоку навантаження, що характеризується суттєвими локальними флуктуаціями пропускної здатності та “тяжкими хвостами”. Тому в процесі організації та плануванні міжмережевої взаємодії крім середнього значення пропускної здатності інформаційної мережі необхідно враховувати її пікові значення. У результаті в такій мережі потрібно передбачити значні запаси за критерієм пропускної здатності, як

наслідок таке рішення призводить до негнучкого використання мережевих ресурсів мережевої інфраструктури.

Тут вирішити зазначений недолік можна за допомогою так званого інтелектуального управління комунікаційними мережами, який включає в себе звичайні механізми управління ресурсами, так і механізми зміни параметрів протоколів взаємодії мережі, конфігурації мережі, а також адаптації до вимог користувачів. Під ними розуміється набір спеціальних засобів управління режимами і інтелектуальним вибором відповідного засобу в конкретному випадку з урахуванням внутрішнього стану інформаційної мережі, впливу факторів, що збурюють та необхідності перерозподілу мережевих ресурсів між різними користувачами та інформаційними додатками.

Для аналізу показників якості передачі було проведено дослідження та аналіз структури сучасних потоків передачі. Результати аналізу структури сучасного мережевого потоку передачі (рис.1) з точки зору складу сервісних потоків і додатків дали змогу зробити висновок про те, що число інформаційного контенту в комунікаційній мережі постійно зростає, як і збільшується число додатків, для яких необхідно забезпечити гарантії за якістю обслуговування [2].

Рисунок 1 - Структура сучасного мережевого потоку передачі із точки зору складу сервісних потоків і додатків

Через статистичну природу мережевого потоку навантаження необхідні характеристики продуктивності в кожному із віртуальних з'єднань чи каналів не можуть бути забезпечені за рахунок гарантованого виділення мінімальної пропускної здатності та необхідного коригування цього значення на основі оперативної інформації про стан інформаційної мережі. Одним з можливих підходів щодо вирішення цієї проблеми є управління статистичними розподілом в наступній послідовності: вибору цільових функцій, що характеризують ймовірність втрат інформаційних даних через невідповідність виділеної пропускної здатності та поточного значення потоку навантаження; контролю за числом дозволених віртуальних з'єднань для кожного класу сервісу; оптимального перерозподілу пропускної спроможності між чергами в мережевих вузлах на основі обраного імовірнісного показника якості. Реалізація зазначеного імовірнісного підходу наштовхується на серйозні труднощі методологічного та обчислювального характеру.

Тому необхідні подальші дослідження щодо вдосконалення методів оптимізації управління для вирішення даного завдання на основі оперативної оцінки стану окремих мережних пристроїв та інформаційної мережі в цілому, а також із врахуванням властивостей інформаційних потоків. До перспективних інформаційних технологій слід віднести новий підхід по управлінню інформаційними ресурсами за допомогою програмних модулів чи інтелектуальних агентів, що забезпечують управління мережними ресурсами із урахуванням вимог користувачів. Управління інформаційними потоками в інформаційній мережі підпорядковується наступній логічній послідовності:

- користувач мережі генерує навантаження, яке буде передавати дані в інформаційну мережу протягом деякого часу. Серед параметрів повинні бути специфіковані «пікова швидкість передачі», «середня швидкість передачі», максимальна допустима затримка тощо;
- якщо ж інформаційна мережа має достатню кількість ресурсів для забезпечення запитаних параметрів, то цей потік починає передавати дані в цю мережу, інакше запит відкидається;
- мережевий маршрутизатор проводить класифікацію пакетів із метою визначення приналежності потокам та класами обслуговування, в результаті чого стає можливим моніторинг навантаження кожного потоку і визначення відповідності поточних значень інформаційних параметрів заявленим вимогам;
- інформаційна мережа проводить моніторинг навантаження, що надходить від цього мережевому потоку, і якщо значення її параметрів перевищують задані в початковий момент, то застосовуються певні механізми щодо обмеження навантаження, звані функціями «політики управління навантаженням».

Аналіз показує, що однією із найважливіших функцій інформаційних мереж із комутацією пакетів є статистичне мультиплексування, що полягає в тому, що дані декількох потоків «змінної швидкості» можуть передавати дані через один спільний інформаційний канал, розмір смуги пропускання якого менше, ніж сума пікових швидкостей всіх потоків. Тут очевидно, що пакети інформаційних потоків спільно використовують не тільки смугу пропускання каналу, але і ресурси мережевих пристроїв. Інформаційні потоки конкурують між собою за мережеві ресурси, і тому чим більше конкурентів, тим менше ймовірність, що деякому потоку дістануться ресурси, що задовольняють запитувані їм параметри якості обслуговування. Тому в перспективних комунікаційних мережах необхідно реалізувати певні механізми, що дають змогу регулювати кількість конкуруючих потоків в залежності від необхідних ними параметрів якості обслуговування та доступних мережевих ресурсів [2].

Дослідивши основні взаємовпливи перешкод на головні елементи каналів передачі інформації із позиції теорії імовірності, можливо оцінити коефіцієнти за «технічною надійністю» основних компонентів та елементів телекомунікаційної системі за допомогою відомого співвідношення. Для інформаційних мереж перед приведенням розрахунків приймемо наступні припущення: для спрощення розрахунків будемо вважати, що якщо станція почала передавати, то колізії відсутня. Це припущення можливо зробити виходячи із високої швидкості розповсюдження сигналу по середовищу передачі

$$v = C \Big/ \sqrt{K} = 3*10^8 \Big/ \sqrt{K} \text{ (м/с)}, \qquad (1)$$

де К – коефіцієнт діелектричної проникливості діелектрику і відносно малою відстанню між кінцевими станціями.

Виходячи з цього припущення маємо, що затримка в інформаційній комунікаційній мережі та виконання завдання визначається формулою:

$$W = t_{д1} + t_{п1} + t_{в} + t_{д2} + t_{п2} \ , \qquad (2)$$

де $t_{д1}$ - час необхідний щоб станція отримала доступ до мережі для передачі завдання в комунікаційну мережу;

$t_{п1}$ - час необхідний для передачі завдання по мережі від комп'ютера замовника до комп'ютера виконавця;

$t_{в}$ - час виконання завдання сервером;

$t_{д2}$ - час необхідний для отримання комп'ютером виконавцем доступу до для передачі відповіді комп'ютеру замовнику;

$t_{п2}$ - час необхідний на передачу комп'ютером виконавцем замовнику.

Виходячи з того, що в комунікаційній мережі із загальним середовищем передачі станції рівноправні в доступі до середовища передачі, то маємо змогу прирівняти $t_{д1}$ та $t_{д2}$ і формула буде мати вигляд

$$W = 2t_{д} + t_{п1} + t_{в} + t_{д2} + t_{п2} \ , \qquad (3)$$

де $t_д$ – час необхідний для отримання доступу до середовища передачі.

Перспективні методи управління інформаційними потоками даних в сучасних інформаційних комунікаційних мережах використовуються в рамках передових концепцій мережевого управління. Вони частково дають змогу усунути обмеження існуючих протокольних рішень щодо управління мережевими ресурсами. При використанні активних телекомунікаційних мереж основний акцент робиться на застосуванні мережевих методів управління у поєднанні із методами математичного та динамічного програмування. Указані методи дають змогу забезпечити раціональне використання ресурсів комунікаційної мережі, підвищити її загальну продуктивність, проте жоден із них не враховує ймовірнісно-часові характеристики агрегованих та окремо взятих потоків даних, що висуває потребу у використанні більш інформативних моделей комунікаційної мережі.

Базуючись на результатах аналізу можна зробити висновок про необхідність розробки нових моделей і методів адаптивного управління потоками даних та мережевими ресурсами, спрямованих на забезпечення ефективної роботи інформаційної мережі. Як показав проведений аналіз, для підвищення рівня якості обслуговування та ефективного розподілу доступного мережевого ресурсу до перспективних моделей управління ресурсами висувається ряд важливих вимог, до яких варто віднести наступні:

- урахування та вивчення потокової структури сучасного мережевого потоку навантаження, у зв'язку із зростанням інформаційних даних;
- оптимізаційна постановка та розв'язання завдання управління чергами, пов'язана із необхідністю використання доступного мережного ресурсу;
- підтримка диференціації обслуговування пакетів на інтерфейсах маршрутизаторів комунікаційної мережі відповідно до їх вимог;
- реалізація динамічних стратегій управління інформаційними чергами;
- агрегування потоків та розподіл пакетів по чергах інтерфейсу із врахуванням параметрів переданих потоків, вимог до якості обслуговування мережі, характеристик створюваних черг та інтерфейсу в цілому;
- розподіл пропускної здатності інтерфейсу між окремими чергами;
- виявлення аномалій мережевого потоку навантаження;
- завчасне обмеження довжини черги потоків інформації;

- забезпечення справедливості обслуговування пакетів одного потоку;
- підтримка розподілених рішень з управління інформаційними чергами;
- простота алгоритмічно-програмної та апаратної реалізації мережі;
- віртуалізація мережевих засобів та пристроїв;
- класифікація та маркування мережевих інформаційних пакетів;
- визначення черговості передавання пакетів з черг в канал передачі.

Дослідження показує, що актуалізується проблематика адаптивного структурно - функціонального синтезу логічної інфраструктури інформаційної мережі приймаючи до уваги цільове призначення процесів, флуктуаційний характер та пікові значення інтенсивності потокового навантаження різних типів, що в процесі динамічного програмного конфігурування ресурсів забезпечило б виконання вимог до продуктивності інформаційної мережі, оперативності доставки даних та якості обслуговування користувачів.

Таким чином, на відміну від ідеалізованої моделі побудови у реальних інформаційних мережах проблеми та перспективи побудови систем управління ресурсами інформаційних комунікаційних мереж мають важливе значення для створення нових мереж. Це необхідно та спостерігається не тільки на рівні інформаційної мережі в цілому, але і на рівні окремих комунікаційних пристроїв.

Перелік посилань

1. Лунтовський А. О. Етапи розвитку сучасних інфокомунікаційних сервісів та енергетична ефективність мережевих технологій / А.О. Лунтовський, П. О. Гуськов, А. Р. Масюк // Вісник Національного університету «Львівська політехніка». Серія: Радіоелектроніка та телекомунікації. — Львів: Видавництво Львівської політехніки, 2014. - № 796. - С. 131-139.

2. Стеклов В.К. Інформаційна система: підручник для студентів вищих навчальних закладів за напрямком «Телекомунікації» / В.К. Стєклов, Л.Н. Беркман. – К.: Тех-ніка, 2014. – 792 с.

Процес визначення початку атаки типу HTTP GET flood

Соколюк Я.В.
Науковий керівник – к.т.н.,доц. Муляр І.В.
Хмельницький національний університет

Для виявлення початку атаки та подальшого виявлення шкідливого трафіку оптимальним буде підхід, який базується на аналізі аномалій, що призводить до порівняння поточного стану системи з її нормальним станом.

При цьому порівнюються різні властивості мережної активності. Ці властивості контексті DDoS-атак можуть включати: тип та кількість запитів, кількість запитів певного протоколу, IP-адресу джерела, час та швидкість запитів, тощо [1].

Атака типу HTTP GET flood використовується нападниками для атаки веб-серверів та серверів веб-додатків. Атака - це сукупність на перший погляд законних запитів GET або POST до сервера [2]. Це спеціально розроблені запити на споживання значної кількості серверних ресурсів. В результаті вони можуть призвести до стану відмови в обслуговуванні, без необхідності переповнювати канал великим обсягом трафіку. Такі запити у випадку розподіленої атаки DoS надсилаються з десятків тисяч заражених вузлів. На рис. 1 схематично показує послідовність пакетів у запиті HTTP GET після з'єднання TCP.

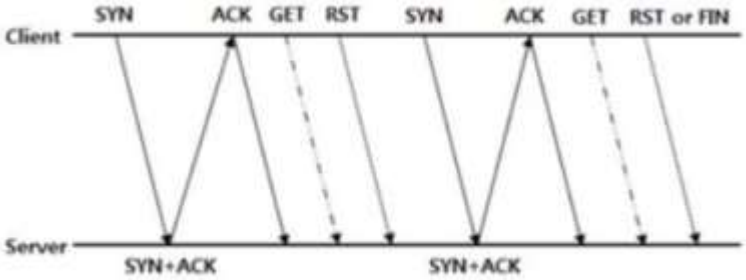


Рисунок 1- Послідовність пакетів при атаці типу HTTP GET

В процесі атаки зловмисник постійно відправляє запити, створюючи при цьому нові TCP з'єднання. Останнім часом [3] також стали розповсюдженими HTTP GET flood атаки в рамках одного TCP з'єднання (див. рис. 2). Цей тип атаки не можливо виявити методом оцінки кількості SYN запитів.

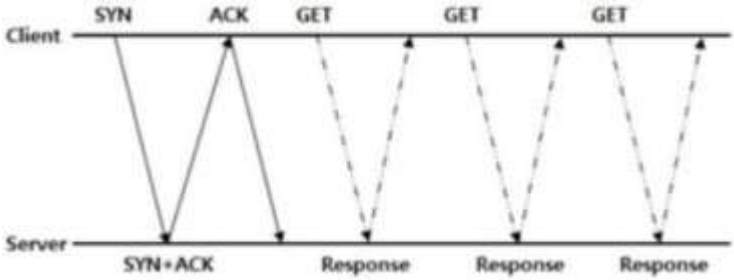


Рисунок 2 - Послідовність пакетів при атаці типу HTTP GET в рамках одного TCP з'єднання

Нині атака HTTP-потоків є однією з найдосконаліших загроз інформаційній безпеці, яка безпосередньо не пов'язана з уразливістю програмного забезпечення. Для обладнання безпеки відрізнити зловмисні HTTP-запити від законних надзвичайно складно, а неправильні методи або налаштування призводять до великої кількості помилкових спрацьовувань. Використання метрик, заснованих лише на оцінці інтенсивності запиту, не є оптимальним методом виявлення DDoS-атак, таких як повеня HTTP, оскільки обсяг трафіку може бути нижче порогового. Тому доцільно використовувати багатокритеріальний метод виявлення DDoS-атак з показниками, які залежать від інтенсивності запитів, і тими, які не залежать від цього показника.

MapReduce - модель проведення розподіленої паралельної обробки великих масивів даних з використанням кластерів (великої кількості обчислюваних блоків). Робота MapReduce складається із двох етапів: Map і Reduce [4].

На етапі Map виконується попередня обробка вхідних даних. Для цього один із обчислювальних елементів кластеру (головний вузол, master node) отримає вхідні дані для розрахунку і розподіляє дані серед робочих вузлів.

На етапі Reduce попередньо оброблені дані обертаються. Основний вузол отримує відповіді від робочих вузлів і на їх основі формує результат - рішення проблеми.

Перевага моделі MapReduce полягає в тому, що вона дозволяє виконувати операції попередньої обробки та згортки паралельно і незалежно, а також горизонтально масштабувати обчислювальну потужність кластера. Операції попередньої обробки діють незалежно одна від одної і можуть виконуватися паралельно (хоча на практиці це обмежується джерелом вхідного сигналу та / або кількістю використовуваних обчислювальних блоків). Аналогічно, група робочих вузлів може конвертувати - для цього потрібно лише те, щоб усі результати попередньої обробки з одним конкретним значенням ключа оброблялися одним робочим вузлом одночасно.

Для роботи багатокритеріального методу виявлення DDoS-атак необхідно провести попередній аналіз та розрахунки: визначити показники (критерії), за якими буде ідентифіковано наявність або відсутність атаки; побудувати модель для звичайного мережевого трафіку; встановити пороги для вибраних показників.

В якості критеріїв оцінки були обрані наступні показники:

- рівень завантаження процесора сервера;
- обсяг зайнятої оперативної пам'яті сервера;
- розмір упаковки;
- поточний рівень трафіку (Мбіт / с);

- розподіл значення адреси джерела запитів (source ip);

- користувач-агент у запиті;

- URI (ієрархічна частина та фрагменти URL-адреси запиту);

Наявність атаки потоку HTTP GET flood може характеризуватися кількістю запитів від джерела за секунду [4]. Зрозуміло, що законний користувач не постійно робить велику кількість запитів на один і той же ресурс, як це може вузол, керований зловмисником (зомбі). Тому деякий час *A*t надходить з IP-адреси джерела x надходить s. запитів.

Для цього необхідно правильно визначити початкову точку атаки. Це дозволить класифікувати весь попередній трафік як законний та відкриє додаткові можливості для розподілу змішаного трафіку, що надходить після атаки, на законний та шкідливий [5]. У цьому випадку метод виявлення шкідливого трафіку, у першому наближенні, буде зведений до наступних етапів:

1. Визначте поточні сезонні періоди.

2. Беручи до уваги сезонність, визначте початкову точку нападу.

3. Ми відносимо весь попередній трафік до початку атаки до законного.

4. Класифікуємо змішаний трафік на законний та шкідливий.

5. Порівняйте законний трафік, вибраний із змішаного, з трафіком, отриманим до атаки.

6. На основі результатів, отриманих на попередньому кроці, та розроблених критеріїв успіху, скоригуємо вибірки.

7. Весь вхідний трафік аналізується на основі отриманих даних.

Серед основних методів можна виділити методи, що базуються на статистичному аналізі. Це допомагає оцінювати різні параметри мережної активності і діагностувати початок атаки або визначати шкідливий трафік.

Основними параметрами, за якими проводиться аналіз, можуть бути:

• Кількість запитів за певний період.

• Швидкість надходження запитів.

• Кількість запитів з певного джерела або з певною мережі.

• Кількість запитів до певного пункту призначення (для вебсервера це конкретний скрипт).

• Час між запитами.

• Інші різні параметри мережевої активності.

За допомогою середньоквадратичного відхилення можна розрахувати допустиму межу для одного з параметрів мережевої активності, наприклад, для кількості запитів за якийсь період часу. У разі якщо межа буде порушена, це стане свідченням початку атаки. Так як в різний час навантаження на мережевий ресурс, так само може бути різною, то для раннього виявлення атаки необхідний постійний моніторинг і перерахунок кордонів для кожного тимчасового кроку. Постійний моніторинг дозволить

визначити атаку, якщо вона почнеться в період невеликої мережевої активності, або, якщо зловмисник шукає потенційно вразливі місця на сервері, проводячи міні- DDOS-атаки і вивчаючи поведінки сервера. У разі якщо верхня межа задана строго і зловмисник проводить міні-атаки в період найменшої мережевої активності, він може не порушувати задану кордон, і його дії будуть не виявлені. Атака буде виявлена тоді, коли зловмисник знайде потенційно вразливе місце, і зробить на нього атаку. Постійний моніторинг активності і перерахунок допустимих меж дозволяє цього уникнути. У період меншою мережевий активності верхня межа знизиться.

Перелік посилань

1. Долішний В.С. Аналіз і моніторинг сучасних DDoS - атак / В.С. Долішний, В.М. Чешун. - Тези доповідей Всеукраїнської науково-практичної конференції молодих вчених, ад'юнктів, слухачів, курсантів і студентів "Молодіжна військова наука у Київському національному університеті імені Тараса Шевченка" [Текст] / за заг. редакцією І.В. Толока. – К. : ВІКНУ, 2018. – С. 147 - 148.

2. DDoS Definitions - DdoSPedia, [Електронний ресурс]. http://security.radware.com/knowledge- center/DDoSPedia/http-flood

3. Zinchenko, V. V, Zinchenko, M. V (2017), Viyavlennya ddos-atak prikladnogo rivnya [Detection of application layer DDoS attacks], Mizhnarodna naukovo-tehnichna konferentsiya «RadIotehnichni polya, signali, aparati ta sistemi», Kyiv, pp. 262-264.

4. Системи і методи виявлення вторгнень: сучасний стан і напрями вдосконалення [Електронний ресурс]. URL:http://citfomm.ni/security/ intemet/ids_overview/#3

5. Холявка Є. П.; Метод виявлення мережевих атак в комп'ютеризованих системах управління: наукова робота, Хмельницький національний університет. - Хмельницький, 2019, [Електронний ресурс]. URL: http://konkurs.khnu.km.ua/wp-content/uploads/sites/25/2019/04/ DP3Eugen.pdf

Прогнозування ризиків завадостійкості в телекомунікаційних системах

Хмельницький Ю.В.
Хмельницький національний університет

Більшість сучасних телекомунікаційних систем визначаються значною кількістю параметрів, функціональними можливостями, вимогами до забезпечення захисту інформації, високою надійністю, розгалуженою інфраструктурою. Для якісної та надійної передачі інформаційних даних у телекомунікаційних системах задача забезпечення завадостійкості та захисту

інформації є однією із головних задач. Сама система має бути запроектована та експлуатуватись так, щоб у разі наявності завад вона забезпечила задану якість передавання інформаційних сигналів. Практично всі розрахунки впливу завад на передавання сигналів та розробка способів зменшення цього впливу є основними задачами, що вирішуються при проектуванні завадостійкості телекомунікаційних систем. Під завадостійкістю каналу передачі інформації тут розуміють здатність такої системи розрізняти та відновлювати сигнали із заданою достовірністю за наявності зовнішніх та внутрішніх завад.

В ряді витоків визначення поняття завадостійкості це здатність системи протистояти шкідливій дії завад, хоча воно більше наближається до розуміння фізичної суті завадостійкості - тут мається на увазі не просто стійкість системи передачі до завад, а її спроможність правильно функціонувати за їх наявності. Завдання визначення завадостійкості усієї телекомунікаційної системи досить складне, тому досить часто визначають завадостійкість окремих ланок системи, наприклад приймача, перетворювача для заданих способів передачі, системи кодування, модуляції. Тому сама завадостійкість телекомунікаційної системи залежить від виду повідомлень, рівня та характеристик завад, параметрів окремих складових частин систем [1]. В умовах реальних динамічних інформаційних завад збільшується ймовірність помилки, стає неможливим забезпечення заданого рівня надійності та вірогідності інформації за допомогою простого використання відомих методів кодування.

Маючи ж необхідну надійність та завадостійкість телекомунікаційної системи можуть забезпечити задану стабільність, захист інформації та безперервність управління. При досліджені та розгляді методів і засобів забезпечення завадостійкої передачі та захисту інформації в телекомунікаційних системах необхідно розглянути, що в широкому розумінні являє собою передача різного роду повідомлень із декількох пунктів у ряд пунктів. В технологіях та засобах передачі і захисту інформації семантична особливість повідомлень не враховується, тому задачею системи передачі інформації в сучасній телекомунікаційній системі є лише транспортування даних у визначене місце, так як оцінка змісту отриманих повідомлень це справа самого одержувача такої інформації.

Теорія та техніка передачі інформації в таких телекомунікаційних системах складалися протягом багатьох років і на сьогодні продовжують швидко та якісно розвиватися. Особливе місце канали передачі інформації займають у сучасних системах управління, в яких необхідно забезпечувати передачу досить великих обсягів потоків інформації із високою швидкістю, достовірністю і надійністю. У процесі функціонування на сучасні телекомунікаційні системи впливають багато різних факторів, що порушують нормальну роботу каналу передачі інформації. Ці фактори призводять до

порушення роботи каналів передачі інформації, фізичного виходу із ладу елементів та компонентів телекомунікаційних систем та інших негативних наслідків. Саму ж основу теорії потенційної завадостійкості розробив ще у 1946 р. академік В.О. Котельников[1]. В теорії потенційної завадостійкості вирішуються такі три основні задачі передачі інформації:

- синтез оптимального приймача – це знаходження правила його роботи та структурної схеми, що забезпечують найкращу якість приймання інформації;
- аналіз роботи оптимального приймача – це обчислення якості приймання сигналів, яка забезпечується цим приймачем потоків інформації;
- порівняння потенційної та реальної завадостійкості такої системи передачі інформації в телекомунікаційній системі.

В цих дослідженнях для практичного використання порівняння завадостійкості має особливе значення. Тут порівнювати реальну завадостійкість різних систем, схем, пристроїв, методів оброблення, видів модуляції не має ніякого сенсу. Таких схем та методів існують досить багато та зростання їх числа триває, а мала завадостійкість якоїсь системи чи схеми ще не означає, що вона є невдала чи неякісна. За таких завад кращої якості вже неможливо досягти. Тому порівняння реальної та потенційної завадостійкості системи дає можливість оцінювати якість реальної телекомунікаційної системи та знайти ще не використані резерви. Аналіз показує, якщо знати потенційну завадостійкість приймача каналу передачі інформації, можна завжди оцінити, наскільки близька до неї реальна завадостійкість існуючих способів приймання та наскільки доцільне їх подальше удосконалення для заданого методу передавання по каналах передачі інформації у системі.

Знання про потенційну завадостійкість за різними методами передавання потоків інформації дають змогу порівнювати ці методи між собою та знайти, які із них у цьому відношенні є найбільш оптимальними. Розглянемо кількісну міру завадостійкості. Для теоретичних розрахунків як потенційної, так і реальної завадостійкості застосовуються прямі методи оцінки якості передачі інформації. У разі передавання дискретних первинних сигналів для обчислень використовують ймовірність помилки.

Також розглянемо деякі принципи та засоби побудови систем передачі інформації по каналам із деякими шумами та перешкодами. В загальному випадку [2] структурна схема системи передачі інформації із завадами складається із джерела та одержувача повідомлень, перетворювачів повідомлення в сигнал та сигналу в повідомлення, каналу зв'язку. Джерелом повідомлень та одержувачем в одних системах передачі може бути людина, в інших різного роду пристрої – автомат приймання, комп'ютер, периферія тощо. Перетворення повідомлення у сигнал повинне бути оборотним.

В досліджені видно, що в цьому випадку по вихідному сигналу можна відновити вхідний первинний сигнал, тобто одержати усю інформацію, що є

в переданому повідомленні. В противному випадку деяка частина інформації буде загублена при передачі потоку. При передачі необхідних потоків інформації канальний сигнал може спотворюватися та на нього можуть накладатися завади. Приймальний пристрій системи обробляє прийняте коливання, яке є сумою перекрученого сигналу та завади, відновлює по ньому повідомлення, що із деякою похибкою відображає передане повідомлення. Тобто приймач повинен на основі аналізу коливання визначити, яке із можливих повідомлень у системі передавалось.

Аналіз показує, що одноразове використання каналу передачі інформації полягає у тому, що передавач певним чином впливає на канал передачі, а приймач спостерігає деякі характеристики каналу, що відображають цей вплив. Якщо ж канал передачі інформації дискретний, то для передавача існує кінцеве число впливів, які називаються вхідними сигналами [2]. Приймач розрізняє тільки визначене число класів результатів спостереження, що називаються загалом вихідними сигналами. В системі співвідношення між вхідними та вихідними сигналами у загальному випадку має імовірнісний характер. Канал передачі визначається встановленням умовних ймовірностей для кожної вхідної і вихідної послідовності.

Дослідивши основні взаємовпливи перешкод на головні елементи каналів передачі інформації телекомунікаційної системи із позиції теорії імовірності, можливо оцінити коефіцієнти за «технічною надійністю» основних компонентів та елементів телекомунікаційної системі за допомогою відомого співвідношення. Ступінь очікуваних ризиків функціонування телекомунікаційної системи можна подати як добуток імовірності небажаних наслідків на відповідну величину втрат аналогічно як у працях [2]:

$$R = \sum_{i=1}^{n} R_i = \sum_{i=1}^{n} p_i \cdot Z_i \quad , \qquad (1)$$

де R – величина ризику передачі;

p_i – ймовірності небажаних впливів каналу передачі інформації;

Z_i – величини втрат каналу передачі.

Для реального оцінювання ризику якості функціонування телекомунікаційної системи також використовують величину середньозваженого модуля відхилення ΔZ (тут n=12):

$$\Delta Z = \sum_{i=1}^{n} p_i \cdot (Z_i - \overline{Z}) \cdot \qquad \overline{Z} = \frac{1}{n} \sum_{i=1}^{n} Z_i \qquad (2)$$

Також визначають середньоквадратичне відхилення [3]:

$$\sigma = \sqrt{\sum_{i=1}^{n} p_i \cdot (Z_i - \overline{Z})^2}\,, \qquad (3)$$

Якщо ж взяти до уваги негативні відхилення від запланованих даних від параметра $\overline{Z}$, то ступінь ризику якості функціонування та захисту інформації телекомунікаційної системи оцінюється показником варіації S_Z і його значення визначається за допомогою такого відомого співвідношення:

$$S_Z = \sqrt{\sum_{i=1}^{n} p_i \cdot (Z_i - \overline{Z})^2 \cdot I_{vi}} \Big/ \sum_{i=1}^{n} p_i \cdot I_{vi}\,, \qquad (4)$$

де $I_i = \{I_{vi}\}$ – індикатор несприятливих відхилень якості роботи телекомунікаційної системи, якому відповідають:

0, для сприятливого відхилення $I_{vi} = 0$,

1, для несприятливого відхилення $I_{vi} = 1$.

Основним показником оцінювання ризику передачі невірогідної інформації в телекомунікаційних системах може бути також коефіцієнт можливих втрат каналу передачі, який враховує обсяг втрат по відношенню до суми абсолютних значень ймовірних втрат в завадостійких системах [3]:

$$K_Z = M_{ZV} / (M_{ZV} + M_{ZP})\,. \qquad (5)$$

де M_{ZV}, M_{ZP} – відповідно ймовірні величини сприятливих та несприятливих відхилень відносно значень показників θ_V, θ_P при розгляді рівнів втрат при передачі інформації Z і позитивних результатів.

Якщо тут розглядати завадостійкість телекомунікаційної системі як здатність системи протидіяти завадам, для цього треба знати, чим протидіяти та на що протидіяти, тобто для боротьби із завадами потрібні апріорні відомості про властивості носія потоків інформації і про самі завади. До таких властивостей у системі можливо віднести [3]:

– величина струму та напруги вхідного сигналу та завади в каналі передачі телекомунікаційної системи;

– середні потужності сигналу та завади в системі;

– вид та структура переносника інформації в телекомунікаційній системі;

– закон розподілу сигналу передачі тощо.

Дослідження та розгляд таких методів, способів та засобів забезпечення завадостійкої передачі інформації в телекомунікаційних системах показав, що завдання оптимального прийому полягає у використанні властивостей корисного сигналу, завади та каналу передачі для збільшення ймовірності правильного прийому. Для збільшення ймовірності правильного прийому потоків інформації має бути проведене попереднє оброблення прийнятого сигналу, яке забезпечує збільшення відношення сигнал та завада. Метод же накопичення застосовується у тому випадку, коли корисний сигнал протягом часу прийому є постійним та являє собою періодичну функцію. Він полягає у багаторазовому повторенні сигналу та підсумовуванні окремих його реалізацій в приймальному пристрої телекомунікаційної системи. Величину відношення сигнал та завада можна підвищити, якщо використати різницю між кореляційними функціями сигналу та завади. Цей метод є ефективним у випадку застосування в системах передачі періодичних та квазіперіодичних інформаційних сигналів.

Таким чином, на основі досліджень і аналізу методів та засобів забезпечення завадостійкої передачі інформації в сучасних телекомунікаційних системах можливо зробити висновок, що завдання прогнозування ризиків завадостійкої передачі, оптимального та якісного прийому та захисту інформації полягає у використанні властивостей корисного сигналу, завади та каналу передачі інформації для збільшення ймовірності правильного прийому. Для збільшення ймовірності правильного прийому має бути проведене попереднє оброблення прийнятого сигналу, яке забезпечує збільшення спів відношення величини сигнал та завада. Канали передачі інформації, що застосовують технології, які дозволяють у режимі реального часу гарантувати якісну, надійну та вірогідну передачу інформації в умовах впливу завад, краще забезпечують величину заданих значень показників вірогідної передачі інформації здійснюється за рахунок використання необхідного кодування. Знаючи властивості сигналу і завади, можна встановити певні відмінності між ними та використати їх для розроблення способів, засобів та методів забезпечення завадостійкої передачі. На відміну від спотворень завади носять випадковий характер та заздалегідь невідомі і тому не можуть бути повністю усунені. Таким чином, можна зробити висновок про те, що знання методів та засобів побудови сучасних каналів передачі телекомунікаційних систем в умовах дії завад, дозволить будувати надійні канали передачі інформації.

Перелік посилань

1. Бабич В. Д. Завадостійкість каналів зв’язку : навч. посібн. / В.Д.

Бабич, О.Д. Кувшинов, О.П. Лежнюк, С.П. Лівенцев // К. : КВІУЗ, 2001. - 150 с.

2. Хмельницький Ю.В. Забезпечення вірогідної передачі інформації при впливі перешкод в телекомунікаційних мережах / Ю.В Хмельницький, Г.Б.Жиров, Н.В. Кульпак // Збірник наукових праць Військового інституту Київського національного університету імені Тараса Шевченка. – К.: ВІКНУ, 2018. – Вип. № 59. – С. 161-170.

3. Хмельницький Ю.В Методи та засоби забезпечення завадостійкої передачі інформації в телекомунікаційних мережах / Ю.В Хмельницький, О.А. Каблуков, Л.О. Ряба, Л.В. Солодєєва, А.О. Ткач // Збірник наукових праць Військового інституту Київського нац. університету імені Тараса Шевченка. - К.: ВІКНУ, 2019. - № 64. – 133-144 с.

Оцінка ефективності роботи генератора криптоключів підвищеної ентропії для системи клієнт-банк

Чешун В.М., Чорненький В.І.[1], Яцків В.В.[2]
Хмельницький національний університет [1]
Західноукраїнський національний університет[2]

Після виконаної розробки засобів реалізації алгоритму роботи системи клієнт-банк із застосуванням генераторів криптоключів підвищеної ентропії, важливим етапом стає апробація здатності засобів, що реалізують алгоритм, виконувати передбачені функції відповідно до наявних вимог.

Розроблені алгоритм і засоби орієнтовані на накопичення пулу ентропії від джерел з передачею даних в систему клієнт–банк. Парадокс оцінки ентропії полягає в тому, що вона потребує зазначення того, наскільки непередбачувана послідовність. Якби можна було зробити абсолютний доказ непередбачуваності, то за визначенням послідовність була б передбачуваною.

Слід зазначити, що далеко не всі гіпотези про підвищену ентропію застосовуваних джерел і отримуваних на їх базі даних проходять перевірку на відповідність вимогам випадковості значень, тому черговою задачею дослідження постає оцінка якості отримуваного від джерел ентропії пулу тестами на випадковість.

Перевірка якості генерованих різними методами чисел на випадковість є однією із найактуальніших і найскладніших задач при розробці і впровадженні генераторів псевдовипадкових чисел.

Актуальність задачі зумовила до активного пошуку її розв’язку, а складність – до відсутності єдиного універсального рішення.

Як наслідок, на сьогоднішній день розроблено велику кількість методів перевірки якості послідовностей псевдовипадкових чисел, що

базуються на різних принципах [1, 2]:

- графічні тести;
- евристичні тести;
- статистичні тести.

Графічні тести [1] відображують властивості послідовностей графічними залежностями (графіками або просторовими моделями: гістограма розподілу елементів послідовності, розподіл на площині, графіки перевірки на монотонність), з аналізу яких робляться висновки щодо характеристик досліджуваних послідовностей.

Евристичні тести [2] формують відносну оцінку кількох версій генераторів, висновок дається за результатами порівняння або за гіпотезами оцінювання.

Статистичні тести [1, 2] вишукують в послідовностях повторювану складову і дають оцінки якості генератора за її статистичними властивостями.

Згідно [1], статистичні тести на основі оціночних критеріїв роблять висновки про ступінь близькості властивостей аналізованої і істинно випадкової послідовності. На відміну від графічних і евристичних тестів, де результати інтерпретуються користувачами або базуються на випадкових вибірках (тест днів народження, мавпячий тест тощо), в результаті чого можливі відмінності в трактуванні результатів, статистичні тести характеризуються тим, що вони видають чисельну характеристику, яка дозволяє однозначно сказати, пройдений тест чи ні.

З цього слідує перевага статистичних тестів відносно графічних і евристичних.

До категорії статистичних тестів можна віднести наступні популярні тести [1, 2, 3, 4, 5]:

– тести Кнута – один з перших наборів статистичних тестів, запропонований Д. Кнутом в 1969. Тести обчислюють значення статистики, воно порівнюється з табличними результатами. Залежно від ймовірності появи отриманої статистики робиться висновок про її якість. Перевага тестів – мала кількість і швидкі алгоритми виконання. Недолік – невизначеність трактування результатів.

– тести Diehard – набір тестів для вимірювання якості набору випадкових чисел. Набір Diehard з 14 тестів Джорджа Марсалі був першим для комплексного тестування генераторів, розроблявся декілька років і опублікований в 1995р. Тести Diehard розглядаються як один з найбільш суворих існуючих наборів тестів, але і їм притаманний ряд недоліків [3]:відсутній докладний опис тестів і методика трактування результатів; параметри тестування жорстко задані, через що тести не адаптовані для перевірки послідовностей різних розмірів; більшість тестів є наближеними і засновані на результатах емпіричних випробувань, а не на статистичних

моделях;

– набір статистичних тестів Crypt-X розроблений дослідниками науково-дослідного центру з інформаційної безпеки технологічного університету Квінсленда (Австралія). Це комерційний пакет програмного забезпечення. Тести застосовуються в залежності від типу алгоритму генератора і спрямовані на тестування генераторів псевдовипадкових чисел. Підтримуються потокові шифри, блокові шифри, генератори потоку ключів. У набір включені наступні тести: частотний, на послідовність однакових бітів, лінійна складність, складність послідовності, двійкова похідна, зміна точки. Основний недолік для даної роботи – відсутність детального опису тестів та комерційна основа розповсюдження.

– тести NIST – перший крок до стандартизації набору статистичних тестів (в 1994р. в національному стандарті США «Вимоги безпеки до криптографічних модулів»). Однак вимоги і методика стандарту носили більше технологічний характер. У 1999 р фахівцями NIST (Національний інститут стандартів і технологій (ність) США), в рамках проекту AES (Advanced Encryption Standard) був розроблений набір статистичних тестів «NIST STS» (NIST Statistical Test Suite) і запропонована методика проведення статистичного тестування генераторів, орієнтованих на використання в задачах криптографічного захисту інформації, яка, на погляд багатьох фахівців в даній області, на даний момент найкращим чином відповідає потребам всіх зацікавлених сторін. Пакет NIST STS включає в себе 15 статистичних тестів, які розроблені для перевірки гіпотези про випадковість послідовностей довільної довжини. Всі тести спрямовані на виявлення різних дефектів випадковості;

– тести стандарту FIPS140 є складовою стандарту FIPS (федеральний стандарт по обробці інформації) – державного стандарту США, що описує вимоги до шифрування і пов'язаних з ним заходів безпеки IT-продуктів, які використовуються для обробки конфіденційної інформації без грифу секретності. FIPS 140-1 був випущений в 1994 році, йому на зміну прийшов стандарт FIPS 140-2 в 2001 році, а в 2019 році з’явився FIPS 140-3 – це нова версія стандарту, актуальна на сьогодні.

Це далеко не повний перелік тестів, але достатній для формування уяви про тенденції розвитку методів тестування і прийняття рішення щодо вибору одного з методів.

За результатами проведеного аналізу можна зробити висновок про доцільність вибору статистичних тестів, які найкраще себе зарекомендували і стали стандартами перевірки якості генераторів псевдовипадкових чисел.

Найновішим стандартом тестування псевдовипадкових послідовностей є стандарт FIPS 140-3, виданий у 2019 році. Новизна стандарту зумовлює відсутність його перекладів та програмних реалізацій тестерів. Тому в апробації алгоритму роботи системи клієнт-банк із

застосуванням генераторів криптоключів підвищеної ентропії застосуємо аналітичне дослідження властивостей генерованих алгоритмом даних на відповідність вимогам FIPS 140-3.

Тести стандарту FIPS140 виконуються над послідовностями довжиною 20000 біт і включають 4 тести:

- монобітний тест;
- блоковий тест (тест покеру);
- тест серій;
- тест довжин серій.

Для проведення випробувань обрано фрагмент пулу ентропії довжиною 20000 біт, сформований розробленим програмним додатком на підставі накопичення ентропії з датчиків мобільного телефону.

Монобітний тест полягає в підрахунку кількості нулів і одиниць в послідовності певної довжини. Тест вважається пройденим, якщо кількість нулів (n_0) і одиниць (n_1) лежить в діапазоні від 9654 до 10346.

За результатами статистичного аналізу тестованої послідовності отримуємо:

- n_0=9996, $9654 < n_0 < 10346$;
- n_1=10004, $9654 < n_1 < 10346$.

З отриманих результатів робимо висновок – монобітний тест успішно пройдено.

Блоковий тест (тест покеру) полягає в наступному. Потік даних довжиною 20000 біт розбивається на чотирьохрозрядні двійкові коди (5000 кодів по 4 біта кожен), після чого виводиться статистика появи кожного коду. Статистичні дані підставляються в формулу:

$$X = \frac{16}{5000} * \sum_{i=0}^{15} K_i - 5000\,, \qquad (1)$$

де K_i – кількість входжень коду зі значенням i в тестовану послідовність (для чотирьохрозрядних кодів $0 \le i \le 15$).

Блоковий тест вважається пройденим, якщо розрахункове значення попадає в діапазон від 1,03 до 57,4.

В деяких джерелах [5] зазначається, що статистичні характеристики розбиття можуть змінюватися при зсуві вхідного коду, тому для якісної оцінки потрібно дослідити всі 4 варіанти розбиття з циклічним зсувом тестованого коду пулу ентропії довжиною 20000 біт (після четвертого зсуву статистичні дані будуть циклічно повторюватися з кожним зсувом).

Результати експерименту наведені в таблиці 1.

За даними експериментів (Експ. 1-4), наведеними в таблиці 1, побудовано гістограми статистичних даних кількості входжень i-го коду в послідовність даних пулу ентропії (рис. 1-4).

Таблиця 4.1 – Результати блокового тесту (4 можливих варіанти)

i	*i*-й код	Статистичні дані кількості входжень *i*-го коду в послідовність даних пулу ентропії			
		Експ. 1: зміщення 0	Експ. 2: зміщення 1	Експ. 3: зміщення 2	Експ. 4: зміщення 3
0.	0000	304	309	306	292
1.	0001	306	305	304	332
2.	0010	289	287	311	281
3.	0011	362	336	310	342
4.	0100	301	294	319	308
5.	0101	321	315	310	325
6.	0110	300	298	307	261
7.	0111	326	365	309	361
8.	1000	310	301	318	318
9.	1001	316	316	319	319
10.	1010	319	343	322	341
11.	1011	301	280	312	284
12.	1100	316	343	318	318
13.	1101	302	320	315	295
14.	1110	363	335	306	357
15.	1111	264	255	314	266

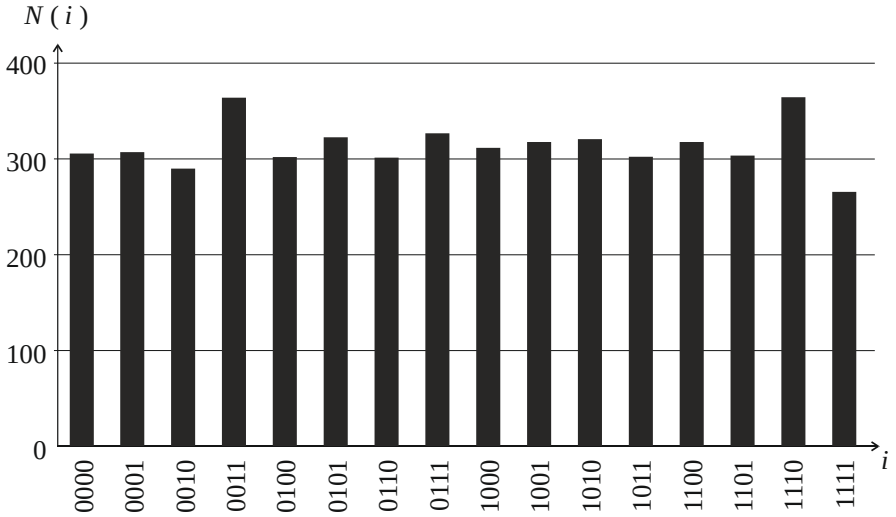


Рисунок 1 – Гістограми статистичних даних кількості входжень *i*-го коду в послідовність даних базового пулу ентропії без зміщення

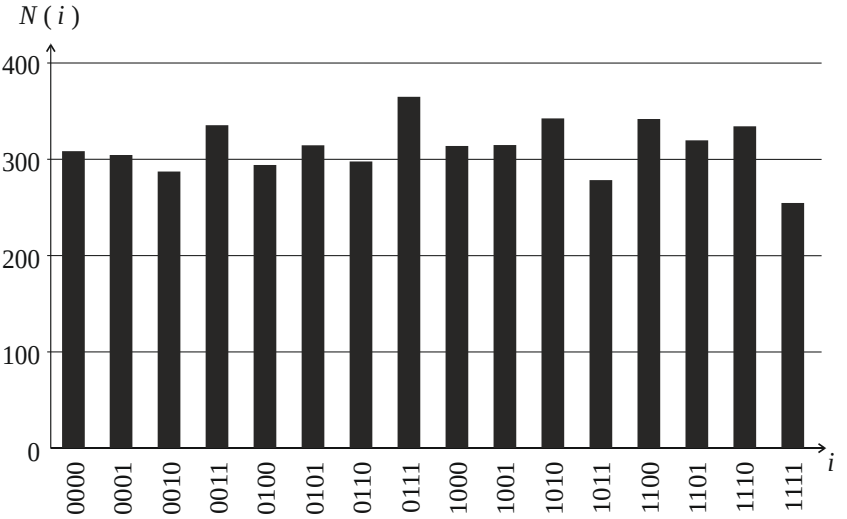


Рисунок 2 – Гістограми статистичних даних кількості входжень *i*-го коду в послідовність даних пулу ентропії зі зміщенням пулу на 1 розряд

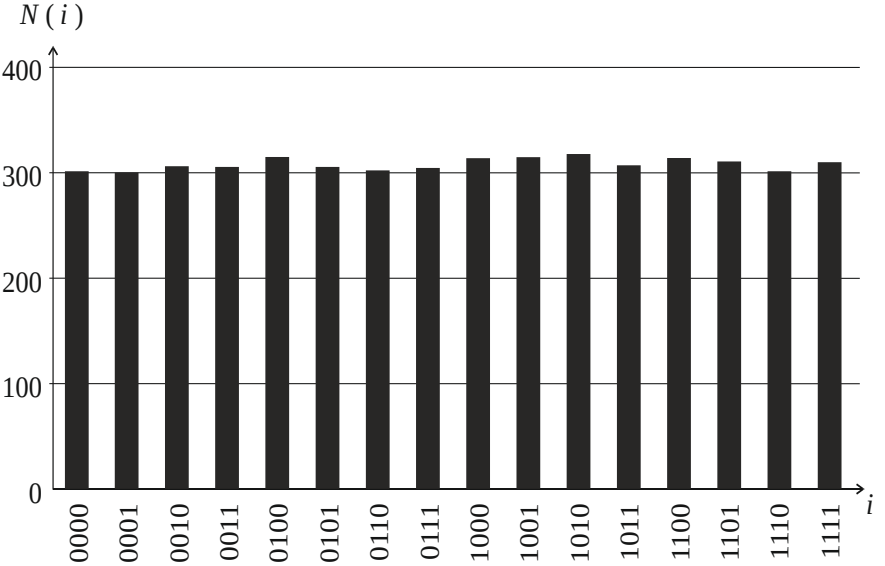


Рисунок 3 – Гістограми статистичних даних кількості входжень *i*-го коду в послідовність даних пулу ентропії зі зміщенням пулу на 2 розряди

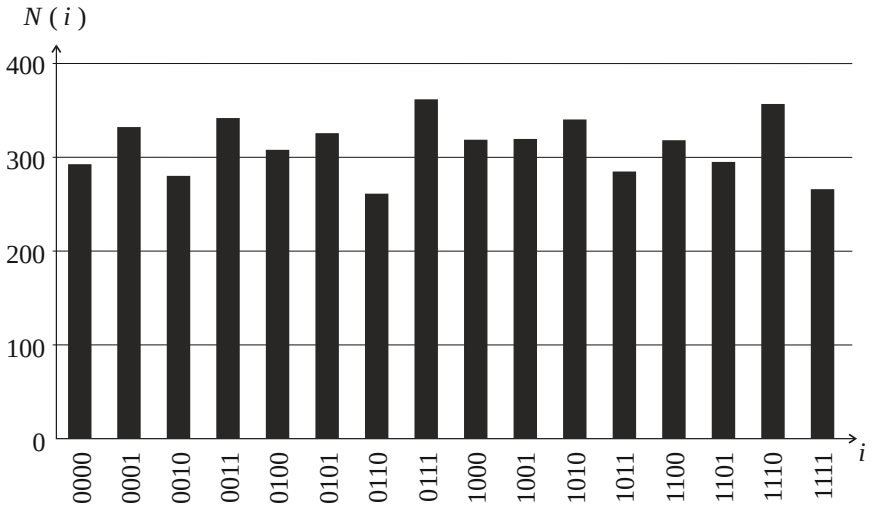


Рисунок 4 – Гістограми статистичних даних кількості входжень *i*-го коду в послідовність даних пулу ентропії зі зміщенням пулу на 3 розряди

За даними таблиці 1 маємо наступні розрахункові дані показника X_j (*j* - номер експерименту):

– експеримент 1: X_1=28.4096, 1,03 < X_1 < 57,4;
– експеримент 2: X_2=40,8512, 1,03 < X_2< 57,4;
– експеримент 3: X_3=1.4656, 1,03 < X_3 < 57,4;
– експеримент 4: X_4=44,864, 1,03 < X_4 < 57,4.

З отриманих результатів висновок – блоковий тест успішно пройдено.

Тест серій – базується на підрахунку кількостей послідовностей (серій) однакових символів (нулів або одиниць) в послідовності. Послідовність вважається випадковою, якщо появи серій певної довжини лежить в заданих діапазонах. Послідовності довжиною більше 6 біт розглядаються як серія довжиною 6 біт. Результати виконання тесту наведені в таблиці 2 (окремо для серій одиниць і нулів). Діапазони оцінки нижньої і верхньої допустимої межі взято з [4].

Таблиця 2 – Результати тесту серій

Дані серій	Розрядність серії				
	2 біти	3 біти	4 біти	5 біти	6 біт
Нижня межа	2267	1079	502	223	90
Серії нулів	2328	1414	618	335	191
Серії одиниць	2279	1307	613	314	135
Верхня межа	2733	1421	748	402	223

З наведених в таблиці 2 результатів може бути зроблено висновок – тест серій успішно пройдено.

Заключним і одним із найпростіших є тест довжин серій. Даний тест визначає максимальну допустиму серію нулів або одиниць в послідовності Якщо послідовність випадкова, то максимальна довжина серії не повинна перевищувати 34 розряди [проц], оскільки ймовірність появи такої серії у випадковому потоці дуже низька.

Дослідження пулу ентропії показали, що довжина серії нулів l_0 і одиниць l_1 мають наступні характеристики: l_0=12, l_0<34, l_1=11, l_1<34.

З отриманих результатів робимо висновок – тест довжин серій успішно пройдено.

Оскільки тести стандарту FIPS140 включають 4 тести, кожен з яких для досліджуваного значення пулу ентропії, отриманого на підставі алгоритму роботи системи клієнт-банк із застосуванням генераторів криптоключів підвищеної ентропії, пройдено, то можна стверджувати, що тести стандарту FIPS140 в дослідженні пройдено.

Тести стандарту FIPS140 є дійсними на сьогоднішній день, але зазнають певної критики через малу кількість тестів і невелику складність деяких з них. Зокрема, наголошується на більшій складності тестів вільного доступу.

Дійсно, для реалізації тестів розроблено ряд програмних продуктів, частина яких є доступною як інтернет-ресурс:

– програма Statistica компанії StatSoft містить тести для перевірки приналежності послідовності заданому розподілу;

– програма Statistics Toolbox / Hypothesis Tests в програмі MathLab містить

– функції тестування статистичних гіпотез;

– NIST Statistical Test Suite – тестування на відповідність NIST;

– TEST-U01 – пакет статистичних емпіричних тестів, реалізований на мові ANSI C;

– CRYPT-X – містить частотний тест, тест підпослідовностей, тест перевірки лінійної складності. В комерційному варіанті має 10 тестів;

– The pLab Project – набір тестів (деталізація відсутня);

– Diehard – класичні тести Diehard (14 тестів, в тому числі тест дні народження, тест пересічні перестановки, тест ранги матриць, мавпячі тести, тест підрахунок одиничок, тест на парковку, тест на мінімальну відстань, тест випадкових сфер, тест стиснення, тест пересічних сум, тест послідовностей, тест гри в кості тощо);

– Dieharder – альтернативна реалізація тести Diehard.

Серед перелічених продуктів тестування послідовностей на випадковість найбільшу кількість тестів (15) реалізує тест NIST Statistical

Test Suite, який вважається найбільш повним статистичним тестом. Це стало підставою для вибору цього тесту.

Тест пройдено онлайн на сайті https://randomness-tests.fi.muni.cz/, настроювання програми тестування, використані для проходження тесту, наведено на рис. 5.

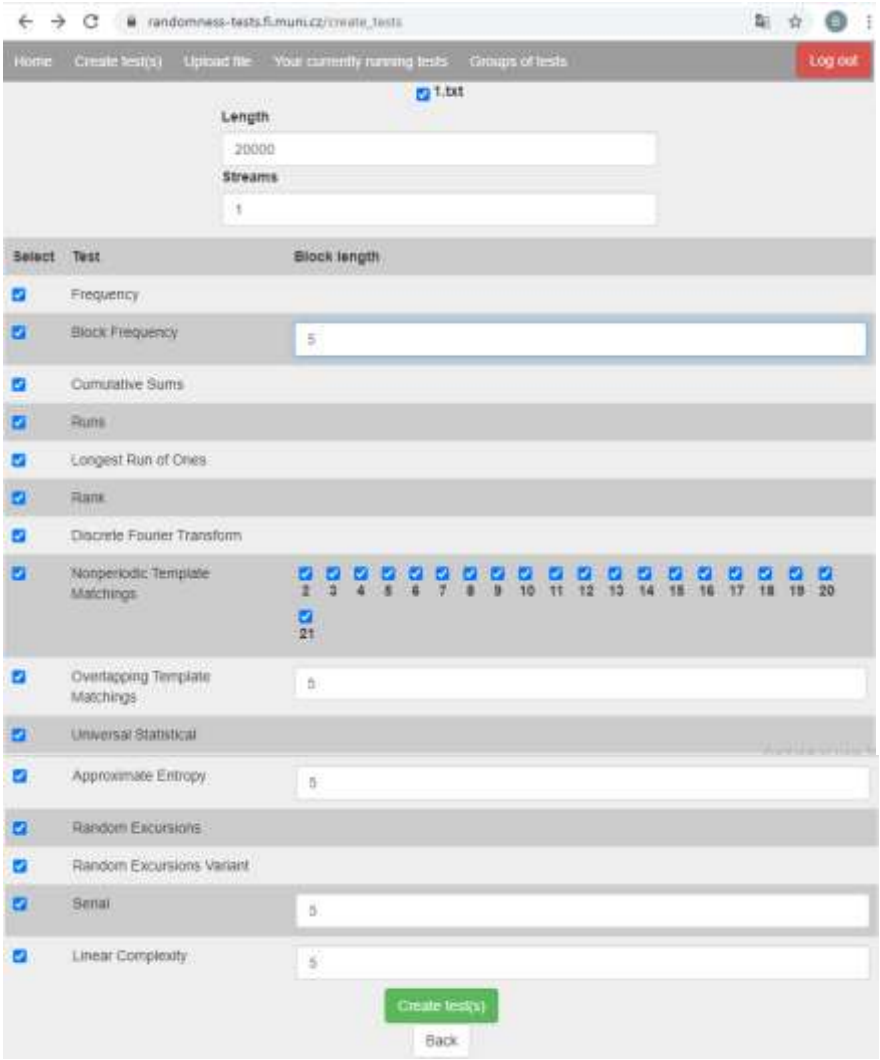


Рисунок 5 – Настройки програми проходження тесту на випадковість за стандартом тестування NIST STS

З рисунку 5 видно, що кількість різновидів тестів 15, але частина тестів дозволяє настроювання параметрів довжин серій або інших параметрів, що дозволяє значно посилити якість тестування і достовірність висновку.

Загалом результати випробувань пулу ентропії на випадковість складалися загалом з 209 тестів. Всі тести успішно пройдено, що свідчить про досягнення очікуваного результату при розробці і апробації алгоритму роботи системи клієнт-банк із застосуванням генераторів криптоключів підвищеної ентропії та засобів його реалізації.

Перелік посилань

1. Григорьев А. Ю. Методы тестирования генераторов случайных и псевдослучайных последовательностей / Григорьев А. Ю. // Ученые записки УлГУ. Сер. Математика и информационные технологии. – 2017. – № 1. – С. 22-28.

2. Слеповичев И.И. Генераторы псевдослучайных чисел / И.И. Слеповичев Саратов: СГУ, 2017. – 118 с.

3. Шевченко Д.Н. Методика тестирования и использования генераторов псевдослучайных последовательностей / Д.Н. Шевченко, С.В. Кривенков // Проблемы физики, математики и техники, № 2 (19), 2014

4. Проценко А.Г. Тестирование генераторов псевдослучайных чисел систем программирования на основе стандарта FIPS140-1 / А.Г. Проценко, И.В. Лысенко // Системи обробки інформації, 2010, випуск 2 (83) 130-132.

5. Шелест М. Є. Експериментальне дослідження методу генерування тритових псевдовипадкових послідовностей для криптографічних застосувань / М. Є. Шелест, С. О. Гнатюк, Т. О. Жмурко, В. М. Кінзерявий, Х. І. Юбузова // Захист інформації. - 2017. - Т. 19, № 1. - С. 67-79.

Симетричний алгоритм нелінійного шифрування з можливістю стеганографічного застосування

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

На сьогоднішній день захист інформації є одним з найбільш актуальних напрямків кібербезпеки, враховуючи стрімкий ріст інформаційних технологій та збільшення цінності інформації як ресурсу. Вже давно як рядові користувачі, так і великі корпорації діджиталізують особисті дані, серед яких велика кількість персональної, конфіденційної інформації, даних що становлять корпоративну, лікарську, чи навіть державну таємницю.

Більше того, дана інформація пересилається через публічні мережі, зберігається на онлайн ресурсах, що окрім комфорту та зручності створює серйозні безпекові загрози.

Для усунення, чи мінімізації цих загроз, для збереження цілісності, доступності та конфіденційності – трьох базових аспектів захисту інформації, впродовж багатьох століть люди використовують шифрування, яке на сьогоднішній день вивчає наука під назвою криптографія.

Криптографія – це прикладна наука, яка розробляє і впроваджує системи захисту інформації шляхом перетворення вихідних осмислених повідомлень в зашифровані повідомлення, які неможливо, або вкрай важко розшифрувати без криптографічного ключа, хоча вони й вразливі для криптоаналітичних атак [1].

В будь-які часи люди намагалися захистити різноманітну важливу інформацію, ховаючи, маскуючи, або видозмінюючи її. Класична криптографія ґрунтується на припущенні, що ніхто не може вирішити певну складну задачку за реалістичний проміжок часу, або покладатися на аргументи теорії інформації [2].

Сучасна криптографія є математично-комбінаторною наукою, що вивчає способи перетворення інформації з метою її захисту від несанкціонованого використання.

Викликом перед сучасною криптографією є значний прорив у області комп'ютерних електронно-обчислювальних машин, в силу чого незламні до цього криптосистеми стали вразливими. Через це в сучасній криптографії, починаючи з другої половини минулого століття стійкість криптосистеми оцінюється на основі складності обчислень, необхідних для успішного проведення криптоатаки, з використанням усіх можливих ресурсів. Даний підхід запропонував криптолог Джон Неш і він залишається актуальним до сьогодні [3].

Стеганографія також є невід'ємною складовою сучасного захисту інформації, оскільки криптографічне повідомлення, яке представляє собою неосмислений набір байт, неодмінно приверне до себе увагу, в той час як стеганографічне повідомлення у вигляді зображення, чи будь-яких інших даних, що самі по собі жодної цінності не представляють, з великою ймовірністю залишиться не поміченим.

Комплексні алгоритми захисту, що включають в себе як криптографічний так і стеганографічний аспект здатні забезпечити надійний захист важливої інформації.

На підставі цього пропонується використання нелінійності у криптографічних алгоритмах яка з одного боку забезпечить додаткову складність обчислень, необхідних для криптоаналізу шифрованих повідомлень, а з іншого – дасть можливість отримати множину можливих

шифротекстів, при одному і тому ж вхідному повідомленні та ключі, що дасть підгрунтя для стеганографічного використання даного алгоритму.

Пропонується метод шифрування оснований на принципі поліалфавітної блочної підстановки, де система (алфавіт) підстановки для кожного блоку обирається на основі випадкових значень, що генеруються в процесі шифрування. Дешифрування можливе завдяки присвоєнню унікальних ідентифікаторів для кожної підстановочної системи. Кожен алфавіт може мати безліч ідентифікаторів, проте будь-який ідентифікатор може належати лише одному алфавіту заміни.

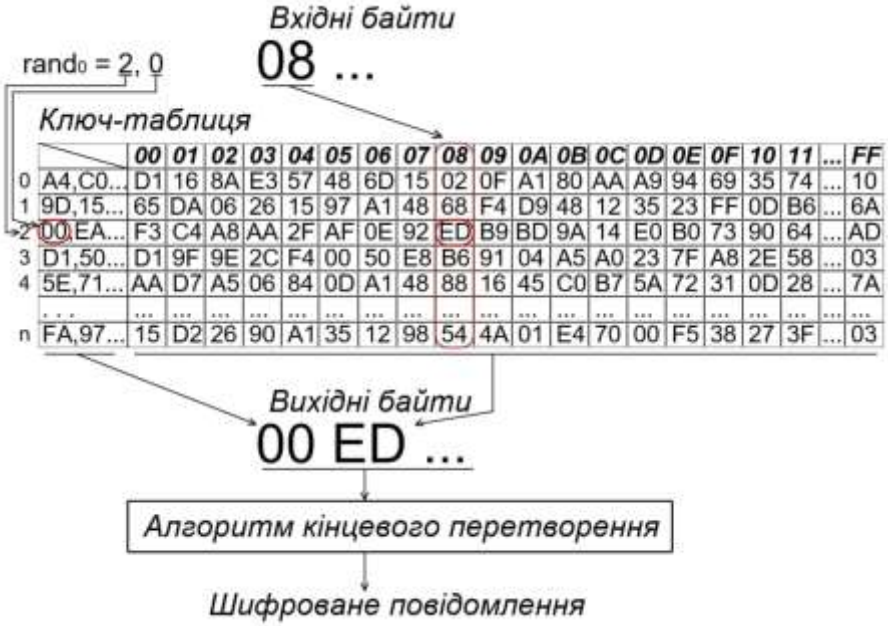
Логікою шифрування в даному алгоритмі буде заміна одного байту на інший, відповідно до деякої поліалфавітної таблиці перестановок. Дана таблиця міститиме в собі n рядків та 256 стовпців, якщо за умовний блок даних ми візьмемо один байт. В кожному рядку буде послідовність від 0 до 255, перемішана випадковим чином. Кількість рядків, а відповідно «алфавітів перестановки», може бути якою завгодно.

Кожному рядку присвоюється m випадкових двійкових ідентифікаторів. Розрядність цих ідентифікаторів також може бути обрана різна, проте вона повинна бути однаковою для всіх ідентифікаторів. Від обраної розрядності залежатиме, по-перше, кількість алфавітів, яку ми можемо визначити, наприклад якщо розрядність дорівнюватиме двом, то максимум у нас може бути 4 алфавіти, а з урахуванням рекомендації використовувати не менше двох ідентифікаторів на один алфавіт – всього 2. По-друге, від розрядності залежатиме те, наскільки збільшиться шифрований текст, порівняно з вхідним повідомленням. Якщо розрядність ідентифікатора буде рівною розрядності нашого блоку даних, а у нашому випадку – це 8 біт, то шифроване повідомлення буде вдвічі більше за вхідне. Чим менша розрядність ідентифікатора, по відношенню до розміру блоку даних, тим менше буде збільшення вихідного повідомлення при шифруванні і навпаки.

Дана таблиця замін, разом із відповідними ідентифікаторами, складатиме собою секретний ключ. Для її компактного запису можна використовувати різноманітні технології стиснення.

Схема шифрування даним методом показана на рис. 1: спочатку ми обираємо два випадкових числа, з яких перше – в діапазоні $[0; n)$, де n – кількість алфавітів, а друге – в діапазоні $[0; m)$, де m – кількість ідентифікаторів, відповідних даному алфавіту. Після цього ми заміняємо вхідний блок даних на вираз, що складається з обраного ідентифікатора та числа, що знаходиться в обраному алфавіті на позиції, номер якої відповідає числу, утвореного з блока вхідних даних. Тобто, якщо ми шифруємо 8-бітні блоки, і на вході, як приклад, отримали блок «10001010», що у десятковій формі відповідає числу 138, а випадкові числа випали 2 і 0, за умови, що вони відповідатимуть межам зазначених діапазонів, то шифрований вираз для даного блоку складатиметься з 0-го ідентифікатора 2-го алфавіту

перестановок та числа, що знаходиться на 138-й позиції у 2-му алфавіті. Такі перетворення почергово проводяться для кожного блоку даних, до кінця відкритого повідомлення.



	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	...	FF
0	A4,C0...	D1	16	8A	E3	57	48	6D	15	02	0F	A1	80	AA	A9	94	69	35	74	...	10
1	9D,15...	65	DA	06	26	15	97	A1	48	68	F4	D9	48	12	35	23	FF	0D	B6	...	6A
2	00,EA...	F3	C4	A8	AA	2F	AF	0E	92	ED	B9	BD	9A	14	E0	B0	73	90	64	...	AD
3	D1,50...	D1	9F	9E	2C	F4	00	50	E8	B6	91	04	A5	A0	23	7F	A8	2E	58	...	03
4	5E,71...	AA	D7	A5	06	84	0D	A1	48	88	16	45	C0	B7	5A	72	31	0D	28	...	7A
. . .	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
n	FA,97...	15	D2	26	90	A1	35	12	98	54	4A	01	E4	70	00	F5	38	27	3F	...	03

Рисунок 1 – Схема шифрування

Криптографічним ключем у даній криптосистемі є таблиця з довільною кількістю випадково перемішаних підстановочних алфавітів, кожному з яких належить набір ідентифікаторів.

Алгоритм шифрування складається з наступних етапів: представлення відкритої інформації у бітовому (двійковому) вигляді, генерація випадкових значень (номеру алфавіту та ідентифікатора) для кожного n-бітного вхідного блоку, підстановка вхідних блоків за випадково обраним алфавітом, зчеплення підставленого блоку та ідентифікатора використаної підстановочної системи, кінцеве перетворення з метою «затирання» ідентифікаторів.

Алгоритм дешифрування є протилежним до алгоритму шифрування. Однозначне дешифрування у даному алгоритмі можливе завдяки ідентифікаторам, які однозначно вказують на те який підстановочний алфавіт було використано для конкретного блоку даних.

Відсутність складної ітеративності забезпечує високу швидкість роботи даного алгоритму.

Таким чином досягається висока ентропія зашифрованої інформації, а випадковість та відсутність закономірностей у перестановках унеможливлює та серйозно ускладнює різні види криптоаналізу. Надійність даного алгоритму напряму залежить від стійкості генератора випадкових чисел.

Стеганографічне функціонування алгоритму базується на припущенні: «Якщо з використанням нелінійного шифрування кожен n-бітний блок може бути замінено багатьма способами, в залежності від того яку підстановочну систему було обрано, а самі підстановочні системи генеруються без будь-яких закономірностей, то, фактично, будь-який вхідний блок після зашифрування може бути перетворений на будь-який вихідний блок, при чому ця відповідність є контрольованою».

Інакше кажучи, даний алгоритм дає можливість створити такий ключ, при якому шифротекст матиме деякий конкретний вигляд, а контрольована генерація ключа дасть можливість контролювати вихідний шифротекст.

На основі цього можна припустити, що можливо створити такий ключ, при якому деяке повідомлення, зашифроване одним із можливих способів на виході дасть шифротест, що представляє собою деяке інше осмислене повідомлення.

Перелік посилань

1. Jara Vera V, Sánchez Ávila C (2019) Graphemic-phonetic diachronic linguistic invariance of the frequency and of the Index of Coincidence as cryptanalytic tools. PLoS ONE 14(3): e0213710 10.1371/journal.pone.0213710
2. Cavaliere, Fabio, John Mattsson, and Ben Smeets. «The security implications of quantum cryptography and quantum computing.» Network Security 2020.9 (2020): 9-15.
3. Прикладна криптологія: системи шифрування : підручник / О.Г. Корченко, В.П. Сіденко, Ю.О. Дрейс. – К. : ДУТ, 2014. – 448 с.

Для нотаток

Наукове видання

«Інтелектуальний потенціал – 2020» - збірник наукових праць молодих науковців і студентів / Колектив авторів – Хмельницький: ПВНЗ УЕП, 2020. – Частина 2. – 100 с.

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

Комп'ютерна верстка: Чешун В.М.
Комп'ютерна верстка: Чешун В.М.
Дизайн Хмельовський В.Р.

Здано до складання 09.11.20. Підписано до друку 09.11.20. Формат 60х84/16. Папір друкарський. Тираж 50 прим. Умовних друкованих аркушів – 7,2.

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

ББК 74.480.278
С.88