НПК МНІС ІП-2018

ЗБІРНИК НАУКОВИХ ПРАЦЬ МОЛОДИХ НАУКОВЦІВ І СТУДЕНТІВ

ЧАСТИНА 3

ПРИСВЯЧУЄТЬСЯ 30-РІЧЧЮ ПІДГОТОВКИ ІТ-ФАХІВЦІВ В ХМЕЛЬНИЦЬКОМУ НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ

ХНУ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Хмельницький національний університет

Військовий інститут Київського національного університету ім.Тараса Шевченка

ПВНЗ "Університет економіки і підприємництва"

Тернопільський інститут агропромислового виробництва

Інтелектуальний потенціал - 2018

збірник наукових праць молодих науковців і студентів

Присвячується 30-річчю підготовки ІТ- фахівців в Хмельницькому національному університеті

сформовано за матеріалами
Всеукраїнської науково-практичної конференції
молодих науковців і студентів «Інтелектуальний потенціал – 2018»

14-16 листопада 2018р.

Частина 3
Кібербезпека та актуальні проблеми комп'ютерних систем і мереж

Хмельницький
2018

ББК 74.480.278
С.88

«Інтелектуальний потенціал – 2018» - збірник наукових праць молодих науковців і студентів з нагоди 30-річчя підготовки ІТ- фахівців в ХНУ/ Колектив авторів – Хмельницький: ПВНЗ УЕП, 2018. – Ч.3: Кібербезпека та актуальні проблеми комп'ютерних систем і мереж – 108 с.

ЗМІСТ

Синтез дискретних діагностичних тестів із застосуванням генетичних алгоритмів

Бадіміна Л.А.
Науковий керівник – к.т.н.,доц. Чешун В.М.
Хмельницький національний університет

Ускладнення задач діагностики дискретних систем зумовило створення різноманітних спеціалізованих засобів з елементами інтелектуального опрацювання діагностичної інформації, серед яких можна виділити нейромережні засоби діагностування, системи на основі нечіткої логіки, експертні діагностичні системи тощо [1-3]. Зазначені засоби мають різні функціональні можливості та призначення і використовуються для вирішення різних видів завдань, що виникають в задачах технічної діагностики, але метою їх застосування завжди є підвищення ефективності діагностичних випробувань і зменшення ролі людського фактору як передумови виникнення великої кількості помилок.

Як перспективний напрямок інтелектуалізації обробки діагностичної інформації визнано застосування генетичних алгоритмів. Практика застосування генетичних алгоритмів в задачах технічної діагностики не є новою, в роботах [5-8] описуються дослідження в цьому напрямку, що проводилися ще в 90-х роках минулого сторіччя. Одним із основних застосувань генетичних алгоритмів в задачах технічної діагностики є оптимізація тестових послідовностей і алгоритмів діагностування, де генетичні алгоритми розглядаються як альтернатива таким традиційним методам оптимізації, як метод гілок та меж, метод динамічного програмування тощо [4, 8-11]. Існують також застосування генетичних алгоритмів для відсіву надлишкових тестів з тестових послідовностей [12]. Відомі роботи, де генетичні алгоритми застосовуються для підбору контрольних точок об'єкта діагностування (ОД), що використовуються для реалізації тестових випробувань, а також для дослідження і модифікації структури ОД, зокрема, із застосуванням ROBDD-графів [13].

В умовах постійного збільшення інтегральної складності дискретних електронних компонентів актуальною залишається задача синтезу якісних тестів, для розв'язування якої також може бути ефективно застосована теорія генетичних алгоритмів [9] .

Генетичні алгоритми відносяться до категорії процедур пошуку, які виникли як спроба копіювання природних процесів наслідування та селекції (природного відбору) як основних рушійних факторів еволюційного розвитку. Генетичні алгоритми застосовувались в різних сферах, де може бути використана теорія еволюційного розвитку і відбору, що визначалося проведенням попередніх досліджень [9, 14].

Для визначення принципів синтезу дискретних діагностичних тестів із застосуванням генетичних алгоритмів першочергово були досліджені

особливості сучасних цифрових пристроїв як об'єктів діагностування.

В результаті дослідження було визначено, що постійне збільшення функціональних можливостей і внутрішньої складності сучасних цифрових вузлів постійно загострює актуальність питання автоматизованого синтезу якісних тестів для їх діагностування, а також застосування новітніх методів вирішення цієї задачі.

Також було зроблено аналіз можливостей застосування генетичних алгоритмів для синтезу дискретних діагностичних тестів.

Було проаналізовано загальну схему дії генетичних алгоритмів, в якій можна виділити чотири основних операції, що дозволяють отримати результат, а саме : формування початкової популяції, відбір батьківських особин, схрещування і мутація. Основною операцією для отримання більш якісного рішення є схрещування, що виконується над двійковими кодами хромосом із застосуванням спеціального оператора – кросингвера.

При дослідженні можливостей генетичних алгоритмів було розглянуто особливості різних їх видів, а також застосовуваних варіантів кросинговерів.

Для вирішення поставленої задачі було визначено набір параметрів математичної моделі, які відображують класичні дані задач технічної діагностики і теорії генетичних алгоритмів:

$$M=< L, P, T, R, C, H >,$$

де

– L:{$l1$, $l2$, … , li, … , lk} - множина можливих несправних технічних станів ОД;

– P:{$P(l1)$, $P(l2)$, … , $P(li)$, ... , $P(lk)$} - множина значень статистичних даних імовірності знаходження ОД в кожному з можливих технічних станів $li \in L$;

– T:{$t1$, $t2$, … , ti, … , tm} - множина тест-векторів, розроблених для ідентифікації станів ОД $li \in L$;

– R:{$r1$, $r2$, … , ri, … , rm} - множина отримуваних векторів відповідних реакцій при поданні на ОД блоку тест-векторів $ti \in T$;

– C:{$c1$, $c2$,… , ci, … , cm} - множина хромосом початкової популяції для реалізації генетичного алгоритму;

– H:{ $h1$, $h2$, … , hi, … , hm} - множина векторів рекомбінації хромосом $ci \in C$. (це множини C,H).

Також розроблено основні оператори, необхідні для переведення діагностичних даних в дані генетичних алгоритмів і для виконання типових операцій над даними, передбачених схемою дії генетичних алгоритмів.

Оператор формування коду хромосоми $ci \in C$ з діагностичних векторів $ti \in T$ і $ri \in R$:

$$c_i = text(t_i) + text(r_i) \ . \qquad (1)$$

Оператор оцінювання можливості схрещування особин, представлених

хромосомами $ci \in C$ і $cj \in C$:

$$y_{ij} = \left(c_i \oplus c_j\right) \wedge h_i \wedge h_j, \tag{2}$$

$$y_{ij} \begin{cases} = 0 \ - \text{ схрещування хромосом } c_i \in C \ i \ c_j \in C \text{ можливе;} \\ \neq 0 \ - \text{ схрещування хромосом } c_i \in C \ i \ c_j \in C \text{ неможливе.} \end{cases}$$

Оператор кросингвера (оператор схрещування хромосом $ci \in C$ і $cj \in C$):

$$c_{ij}^1 = c_i \vee c_j \tag{3}$$

Оператор кросингвера векторів рекомбінації $hi \in H$ і $hj \in H$:

$$h_{ij}^1 = h_i \vee h_j \tag{4}$$

Функція пристосованості особини з кодом хромосоми $cj \in C$:

$$f_j = \sum_{i=1}^{|L|} q_i P(l_i) \tag{5}$$

де $|L|$ – розмірність L; $P(li)$ – імовірність знаходження ОД в стані $li \in L$; qi – ознака здатності тест-вектора $tj \in T$ виявляти ознаки технічного стану $li \in L$ ($qi = 1$ за наявності у тест-вектора $tj \in T$ відповідної здатності, $qi = 0$ за відсутності у тест-вектора $tj \in T$ відповідної здатності).

Синтез дискретних діагностичних тестів із застосуванням генетичних алгоритмів базується на наступних основних принципах:

– застосовуваний вид генетичних алгоритмів – гібридний;

– математичний апарат – математична модель M і запропоновані в ній математичні оператори перетворення діагностичних даних і їх обробки за правилами теорії генетичних алгоритмів;

– генерація двійкових слів кодів хромосом (множини C) виконується зі значень діагностичних векторів елементарних тестів і відповідних реакцій оператором математичної моделі, що реалізує відповідне перетворення діагностичних даних;

– спосіб відбору претендентів для схрещування – пропорційний, на основі ймовірнісної функції пристосованості і контрольного оператора оцінки сумісності;

– тип оператора кросингвера – рівномірний;

– спосіб формування нового покоління – нащадки схрещувань особин за результатами пропорційного відбору і елітарно відібрані особини, що не допускають схрещувань за контрольним оператором оцінки сумісності;

– основний спосіб зупинки роботи генетичного алгоритму – втрата можливості реалізації схрещувань особин в отриманому поколінні;

– додаткові (можливі) способи зупинки роботи алгоритму – обмеження на час виконання або на кількість ітерацій (генерованих поколінь).

Згідно із переліченими основними положеннями методу було розроблено алгоритм його реалізації, в якому можна виділити етап формування початкової популяції, операції схрещування, обробки отримуваних результатів і перевірки умови зупинки роботи генетичного алгоритму (третій з перевірки).

Для підтвердження ефективності розробленого методу і алгоритму його реалізації було проведено його апробацію на реальних діагностичних даних. В результаті застосування запропонованого методу було отримано скорочений набір тестів, деякі з яких набули здатність виявляти декілька несправностей, що дозволило скоротити тестову послідовність майже на 36 відсотків.

У результати виконаного дослідження можна зробити висновки, що свідчать про досягнення поставленої мети і загальну ефективність розробленого методу.

Література

1. Сівак В.А. Методи функціональної належності компонентів діагностичних засобів, які використовують технологію нечіткої логіки / В.А. Сівак // Вісник Хмельницького національного університету. Технічні науки. – Хмельницький : 51, 2015. – № 6 (231). – С. 203-208.

2. Чорненький В.І. Алгоритми діагностування цифрових пристроїв апаратними засобами на базі штучних нейронних мереж / В.І. Чорненький, В.М. Чешун // Зб. наук. праць Військового інституту Київського НУ ім. Тараса Шевченко. – К. : ВІКНУ, 2010. – Вип. 27. – С. 154–161.

3. Рутковская Д. Нейронные сети, генетические алгоритмы и нечеткие системы / Д. Рутковская, М. Пилиньский, Л. Рутковский. - М.: Горячая линия-Телеком, 2006. - 452 с.

4. Абдуллаев П. Ш. Применение генетических алгоритмов при диагностировании авиационных ГТД / П. Ш. Абдуллаев, А. Дж. Мирзоев // Авіаційно-космічна техніка і технологія. – Харків : ХАІ, 2016. – № 7(134)– С. 139-146.

5. Prinetto P. An automatic test pattern generator for larges equential circuits based on genetic algorithms / P. Prinetto, M. Rebaudengo, M. Sonza Reorda // Proc. Int. Test Conf. – 1994. – P.240–249.

6. Rudnick E.M. Sequential Circuit Test Generation in a Genetic Algorithm Framework / E.M. Rudnick, J.H. Patel, G.S. Greenstein, T.M. Niermann // Proc. Design Automation Conf. – 1994. – P.698–704.

7. Городилов А.Ю. Генетический алгоритм диагностирования цифровых устройств / А.Ю. Городилов // Вестник Пермского национального исследовательского политехнического университета. Электротехника, информационные технологии, системы управления. – Пермь: ПНИПУ, 2013. – № 7. – С. 54-62.

8. Дубровин В.И. Диагностика на основе генетических алгоритмов /

В.И. Дубровин, Е. Н. Федорченко // Радіоелектроніка, Інформатика, Управління. – Запоріжжя: ЗНТУ, 2006. – № 2. – С. 115-120.

9. Попов В.А. Оптимизационные задачи на основе генетического поиска / В.А. Попов, А.В. Бердочник // Системи обробки інформації – Харків: ХУПС, 2010. – Вип. 9 (90). – С.217-220.

10. Соколова Э.С Оптимизация коэффициента глубина поиска дефектов методом генетических алгоритмов / Э.С. Соколова, С.Н. Капранов // Контроль. Диагностика. – М. : ООО "Издательский дом "Спектр", 2004. – №4. – С. 32-39.

11. Иванов Д.Е. Генетические алгоритмы построения входных идентифицирующих после-довательностей цифровых устройств. / Д.Е. Иванов – Донецк: ТОВ «Цифровая типография», 2012. – 240 с.

12. Миронов С.В. Генетические алгоритмы для сокращения диагностической информации / С.В. Миронов, Д.В. Сперанский // Автоматика и телемеханика. – М. : Академиздатцентр «Наука» РАН, 2008. – No7. – С.146-156.

13. Дмитриев Д.В. Адаптация генетических алгоритмов к решению задач назначения точек контроля в объектах с большим числом состояний / Д.В. Дмитриев, Э.С. Соколова, С.Н. Капранов // Нейрокомпьютеры, М: Издательство «Радиотехника» № 11, 2007. – С.59-64.

14. Juang C.F. A TSK-Type Recurrent Fuzzy Net-work for Dynamic Systems ceasing by Neural Network and Genetic Algorithms / C.F. Juang //IEEE Trans. on Fuzzy Systems. 2002. – Vol. 10, Issue 2. – P. 155-170.

Інтелектуальна експертна система оцінки та аналізу анкетування медичних працівників про прихільність до гігієни рук при виконанні професійних обов'язків

Баліцький В.В.
Науковий керівник – к.т.н. Чумаченко Д.І.
Національний аерокосмічний університет ім. М. Є. Жуковського «Харківський авіаційний інститут»

Інфекції, пов'язані з наданням медичної допомоги (ІПМД), спричиняють негативні медичні та соціальні наслідки та суттєві економічні збитки для пацієнтів та систем охорони здоров'я світу. В той же час якісна гігієна рук в належний час та належним чином може зберегти життя багатьох людей. Чітке виконання правил гігієни рук рекомендується для профілактики всіх ІПМД.

Керівництво установ охорони здоров'я несе відповідальність за забезпечення профілактики і контролю випадків ІПМД і попередження передачі епідеміологічно важливих патогенів. Медичні працівники, які беруть безпосередню участь у наданні медичної допомоги пацієнтам

(наприклад, лікарі та медсестри), а також допоміжний персонал, несуть відповідальність за постійне використання практики профілактики і контролю ІПМД, у тому числі, вимоги гігієни рук.

З метою оцінки якості знань медичних сестер декількох лікувально-профілактичних закладів з питань гігієни рук, а також вивчення їх обізнаності у питаннях правил виконання медичних маніпуляцій, прийнятих у стаціонарі, і виявлення причин, що ускладнюють виконання медсестрами правил гігієни рук, нами була розроблена відповідна анкета. Анкета включала питання, що характеризують респондентів за статтю, віком і стажем роботи, а також питання, що відображають зміст чинного регламентуючого документа: Наказ МОЗ України від 21.09.2010 № 798 «Про затвердження методичних рекомендацій «Хірургічна та гігієнічна обробка рук медичного персоналу».

Для автоматизації оцінки знань і збору даних у медичного персоналу розроблено web-додаток, платформою для якого обраний .net core, тому що при розробці більшість необхідних компонент додатки можуть завантажуватися як окремі модулі через пакетний менеджер NuGet. Це дозволяє зменшити кількість надлишкових залежностей і загальний розмір готового продукту. Також проект на базі .NET Core досить легко перенести в хмару. Microsoft Azure вже підтримує розміщення .NET Core проектів як в службах Application Services, так і на віртуальних машинах. .NET Core дозволяє невеликим проектам отримати всі переваги платформи корпоративного рівня, при цьому надаючи зручні та засоби розробки, а також недорогу інфраструктуру. Також проект на базі .NET Core найкраще підходить для обчислювальних і аналітичних задач.

Даний web-додаток реалізовано у вигляді web-сторінки, на якій користувачу пропонується відповісти на запитання розробленої анкети. Результатом роботи програми є рекомендації, засновані на аналізі отриманих відповідей користувача системи в процесі анкетування . Рекомендації є результатом розробленої у web-додатку експертної системи.

Було проведено анкетування 817 середніх медичних працівників лікувально-профілактичних закладів. Перед проведенням дослідження респондентам було роз'яснено мету проведення опитування і правила заповнення анкети. Анкетування проводилося на добровільних засадах. Варіант відповіді респонденти обирали самостійно. Результати дослідження були оброблені статистично.

Був проведений аналіз даних, отриманих в ході анкетування. Результати були перевірені на валідність. Валідність характеризує придатність тесту вимірювати певну величину. Слід зазначити, що не можна говорити про валідність тесту, не вказавши умов його застосування. Також валідність означає придатність тесту вимірювати ту властивість, для визначення якої він призначений. Даний тест спрямований на оцінку рівня знань середнього медичного персоналу з приводу профілактики гігієни рук

при виконанні професійних обов'язків. Вимірюваною властивістю в цьому випадку є рівень знань медичного персоналу. В ході проведення тесту рівень знань не змінювався.

На проходження тесту не впливав критерій «посада», впливав лише рівень знань середнього медичного персоналу, що і є вимірюваною властивістю. Можна стверджувати, що даний тест валідний, і подальший аналіз отриманих даних буде нести якісну і правдоподібну оцінку знань середнього медичного персоналу.

Опитувальник був розділений на два блоки питань: «Вміння» і «Знання».

Блок «Вміння» був сформований так, щоб виявити рівень практичних вмінь медичного персоналу. Наприклад: Як часто Ви дотримуєтеся правил гігієни рук до і після контакту з пацієнтом?

Блок «Знання» був сформований для перевірки рівня знань медичного персоналу. Наприклад: З яких етапів складається хірургічна обробка рук?

Однією з цілей роботи було визначити залежність між двома цими блоками. Для подальшого аналізу всі результати анкетування були відповідно перекодовані. Відповіді на перший блок питань були згруповані за двома категоріями: «Рівень вмінь вище середнього» і «Рівень вмінь менше середнього». Відповіді на другий блок питань були згруповані за категоріями: «Правильно» і «Неправильно». Отримані дані були оброблені на мові R в програмному середовищі RStudio. RStudio є в доступі у версіях з відкритим початковим кодом і має корисні функції як для новачків, так і для досвідчених розробників R, включаючи завершення коду, виконання з джерела, історію пошуку і підтримку для розробки документів Sweave. В результаті було отримано набір даних, який складався з середніх значень кожної групи для кожного блоку питань. Було виявлено, що в середньому на перший блок питань відповіло 791 чоловік (на рівні вище середнього), на рівні нижче середнього – 26 осіб. Правильно на питання другого блоку відповіло 492 людини, неправильно 326 – осіб. Всього було опитано 817 осіб.

Оскільки всі дані представлені категоріальної шкалою, аналіз проводився за допомогою Критерію Хі-квадрата Пірсона. В результаті було отримано p-value = 1, (Оскільки p-value > 0.05), отже можна стверджувати, що прямого зв'язку між першим блоком питань і другим немає.

Було виявлено що, чим старше співробітник, тим він більш досвідчений і його рівень знань з приводу профілактики гігієни рук не залежить від таких двох ознак, як «вік» і «досвід». Обидві ознаки представлені категоріальною шкалою, для визначення значимості відмінностей використовувався критерій Хі-квадрат Пірсона. Це непараметричний метод, який дозволяє оцінити значимість відмінностей між фактичною (виявленою в результаті дослідження) кількістю випадків або якісних характеристик вибірки, що потрапляють в кожну категорію, і теоретичною кількістю, яку можна очікувати в досліджуваних групах при

справедливості нульової гіпотези. В результаті було виявлено, що p-value<2.2e-16 (відмінностей немає).

Був проведений статистичний аналіз, який дозволяє зробити висновки щодо питань гігієни рук у медичних закладах. Встановлено, що в лікувальних закладах, в яких проводилося анкетування існують чітко розроблені правила обробки рук, засновані на діючих нормативних документах. Відомо, що у стаціонарах проводяться навчальні семінари з питань гігієни рук. Серед опитаних більшість знає алгоритми миття і гігієнічної антисептики рук, менше половини знає і може перерахувати послідовність дій при хірургічній обробці рук.

Результати анкетування середнього медичного персоналу показали, що при проведенні навчальних семінарів медичних працівників правилам гігієни рук особливу увагу слід приділяти питанням профілактики КД, в тому числі забезпечення медичних працівників зволожуючими захисними кремами, акцентуючи увагу на правильне виконання всіх етапів гігієнічної та хірургічної обробки рук; необхідно посилити контроль за безперебійним забезпеченням відділень лікарень спиртовими антисептиками і рідким милом і переглянути норми навантажень на медичних сестер для забезпечення можливості якісного медичного обслуговування пацієнтів.

Результати анкетування були перевірені на валідність і було з'ясовано, що дане анкетування валідне. У майбутньому на основі цих результатів можливо розробити програмний комплекс, у виді веб-додатку, який буде спрямований на індивідуальну оцінку кожного опитаного, загальну статистику і рекомендації для адміністрації лікувального закладу для усунення недостатнього рівня знань та можливих факторів, які заважають питанням гігієни рук.

Дослідження параметрів системи управління для SDN архітектури

Бойко Р.В.
Науковий керівник – к.т.н., доц. Хмельницький Ю.В.
Хмельницький національний університет

В умовах постійного зростання обсягів потоку інформації та кількості користувачів сервісів потокового контенту актуальним є розроблення методів та моделей управління процесами передавання даних у інформаційних мережах програмним конфігуруванням для підвищення якості та ефективного використання мережних ресурсів. Аналіз матеріалів щодо стану забезпечення управління інформаційними мережами, шляхів створення та тенденції розвитку систем управління, показує можливість виділити наступне. Розширення функцій інформаційної мережі та послуг, що надаються користувачеві, ставлять підвищені вимоги до гнучкості систем та оперативності управління, їх здатності адаптуватися до умов роботи та

особливостей таких мереж, до забезпечення необхідної якості роботи та живучості як самої мережі, так і системи управління. З огляду на це основним недоліком існуючих методів управління є відсутність інформації про стан мережі в режимі реального часу. Більшість рішень про маршрутизацію приймаються на основі інформації, яка є відносно застарілою в умовах динамічного мультисервісного середовища. Згідно проаналізованих систем та методів моніторингу, збір статистики в існуючих системах моніторингу відбувається за допомогою агентів, що встановлені на мережних вузлах та збирають статистичну інформацію, яка передається з певним інтервалом у центр моніторингу. Існуючі системи моніторингу дають змогу збирати обмежену інформацію про інформаційний потік, яка зводиться максимум до завантаження портів чи окремих черг у буфері. В той же ж час протокол Open-Flow надає можливість проводити моніторинг більшої кількості параметрів.

Організація передачі потоків даних у мережних структурах вимагає нових підходів до управління у зв'язку із лавиноподібним збільшення комутаційних правил та трудомісткості управління мережною інфраструктурою. Традиційний загальний підхід до вирішення проблеми мережних взаємодій припускає послідовну обробку одиниць передачі (пакетів) на кожному рівні еталонної моделі мережної взаємодії ISO/OSI. Одним із напрямів "модернізації" класичного підходу до організації мережної архітектури є створення програмно - конфігуруємо мереж, що використовують протокол Open-Flow. До основних переваг програмно - конфігуруємо мереж відносять - централізоване управління в середовищі, зменшення складності мережі за рахунок автоматизації, вищу швидкість впровадження інновацій, збільшення надійності та безпеки мережі, забезпечення узгодженості політик управління доступом, інжинірингу потоку передачі, параметрів послуг, безпеки, вузько спрямоване управління мережею, поліпшення якості сприйняття послуг користувачами. Основна ідея П-КМ полягає в тому, щоб не змінюючи існуючого мережного устаткування відокремити чи перехопити управління цим устаткуванням за рахунок створення спеціального програмного забезпечення, яке може працювати на звичайному комп'ютері та знаходиться під контролем адміністратора мережі.

В загальновідомій архітектурі SDN виділяють три рівні (рис.1) [1]:

- інфраструктурний рівень, що надає набір мережних пристроїв;
- рівень управління, що включає в себе мережну операційну систему, яка забезпечує мережні сервіси та програмний інтерфейс для управління мережними пристроями;
- прикладний рівень - для гнучкого та ефективного управління інформаційною мережею - прикладні рішення управління мережею.

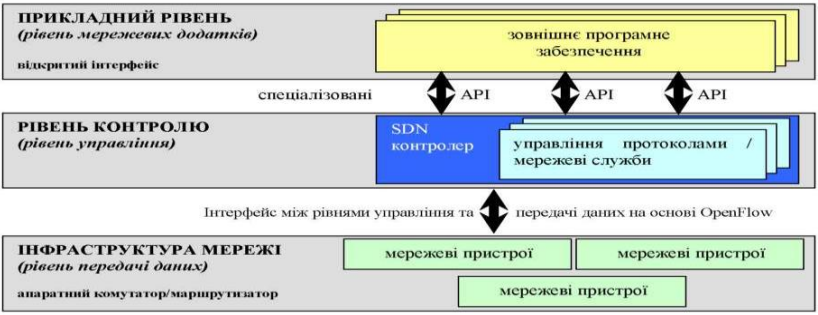

Рисунок 1- Архітектура мереж (SDN)

Основні інтелектуальні функції такої мережі зосереджені у централізованому мережному контролері, який відстежує загальний стан мережної інфраструктури та потоків, що протікають по ній. У такій системі управління всією мережею відбувається в єдиній логічній точці, що значно спрощує завдання конфігурації та управління. Для налаштування такої мережі досить надбудувати програмний контролер мережі, замість того, щоб змінювати сотні рядків кодів у безлічі мережних пристроїв мережі. Поведінку мережі можна змінювати в реальному часі, а нові рішення впроваджувати за набагато коротший час, ніж в традиційній архітектурі побудови мережі. Мережні контролери володіють набором прикладних інтерфейсів, які дозволяють реалізувати типові завдання по маршрутизації, зокрема багато адресність, безпека, контроль доступу, управління смугою пропускання, якість обслуговування, які вузько направлено та налаштовані під завдання конкретного споживача. У комутаторі такої архітектури реалізований тільки рівень передачі даних. Замість контролера використовується набагато простіший пристрій, завдання якого полягає в отриманні даних, які надходять, визначення їх адресів та якщо адресат є в таблиці комутації, негайної передачі даних комутаційній матриці. Інакше комутатор по захищеному каналу відправляє запит на центральний контролер мережі, та на підставі отриманої від нього інформації вносить необхідні зміни у таблицю комутації, після чого здійснюється обробка отриманих даних. Ідея SDN у створенні уніфікованого, незалежного від виробника мережного устаткування, програмно-керованого інтерфейсу між контролером та транспортним середовищем мережі знайшла віддзеркалення в протоколі Open-Flow, що дозволяє користувачам самим визначати і контролювати, хто з ким, за яких умов та із якою якістю може взаємодіяти в такій мережі (рис.2). Сьогодні адміністратор може вручну налаштовувати устаткування за заданими параметрами, а будь-які подальші зміни здійснюються переважно на апаратному рівні. Open-Flow дозволяє відійти від такого управління мережею, що позитивно позначається на її масштабованості.

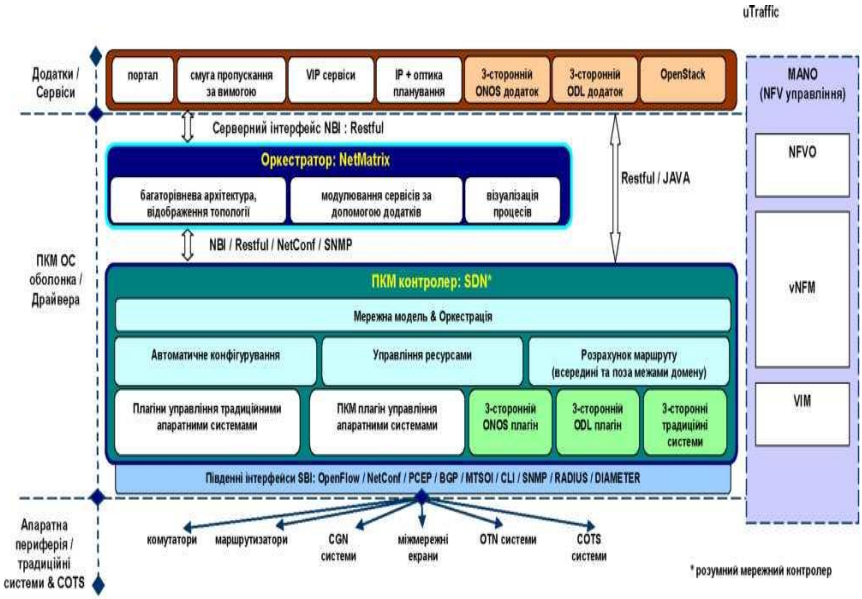

Рисунок 2 – Загальна розширена структура мереж SDN

Сам же комутатор Open-Flow послідовно порівнює зміст переданого кадру із записами таблиці та при збігу виконує вказані в записі дії. Якщо збіг не знайдений, то залежно від налаштувань комутатору пакет може бути відкинутий або відправлений Open-Flow запит контролеру для ухвалення рішення. Контролер мережі може додавати, модифікувати, видаляти записи з таблиць як на основі аналізу пакетів, що отримуються від мережного устаткування, так і виходячи з власних алгоритмів роботи. Мережа SDN дає можливість абсолютної гнучкості в управлінні потоком передачі, а теоретично - легке балансування потоку без залучення окремого приладу. За допомогою сучасних маршрутизаторів зазвичай вирішуються два основних завдання: передача даних - просування пакету від вхідного порту на певний вихідний порт та управління даними - обробка пакету та ухвалення рішення про тих, куди його передавати далі, на основі потокового стану маршрутизатору. Це відповідає рівню передачі даних, на якому зібрані засоби передачі - лінії зв’язку, канало-утворюючі устаткування, маршрутизатори, комутатори) і рівню управління станами засобів передачі даних.

Для вирішення питання оптимізації параметрів управління для SDN архітектури розглянемо вплив завад при передачі даних на достовірність у таких мережах. Періоду збереження надійної роботи каналу передачі

відповідає коефіцієнт збереження каналу – значення показника використання об'єкта за призначенням за певну тривалість експлуатації до номінального значення цього показника, розрахованого за умови, що відмови каналу передачі протягом того ж періоду не виникають. Надійність роботи каналу передачі мережі SDN оцінюємо аналогічно [2] за допомогою відомого співвідношення:

$$P_s(t) = \prod_{i=1}^{n} P_i(t), \quad (1)$$

де $Ps(t)$ – ймовірність безвідмовної роботи каналу передачі мережі SDN; $Pi(t)$ – ймовірність безвідмовної роботи складової частини каналу мережі.

Дослідження та аналіз мереж SDN показав, що дозволяють програмно-конфігурувалися, розділення рівня управління мережею і передачі даних за рахунок перенесення функцій управління мережевими пристроями в додатки, працюють на окремому сервері чи контролері. Це дає можливість у залежності від масштабу мережі SDN, можливо використовувати сам контролер як сервер або група серверів, на яких встановлено спеціалізоване програмне забезпечення. Мережеві елементи, у яких відібрали функції управління мережею, виконують тут суто базові завдання - працюють по просуванню пакетів у мережі. Така архітектура мережі SDN дозволяє виділити із мережевого устаткування рівень управління та зробити його програмованим рівнем. Базова інфраструктура передачі даних також відділяється від мережевих сервісів та додатків. Розглянута узагальнена схема основних напрямів стандартизації та розробки мереж SDN показує, що різні організації при стандартизації мереж SDN, ставлять за мету формування такої архітектури мережі та устаткування, що припускає відділення площини управління від площини передачі та докладають значних зусиль до подолання виникаючих проблем, пов’язаних із складнощами міграції від традиційних мереж до SDN. Це в подальшому надасть можливість користувачам отримувати необхідні послуги із необхідною надійністю, достовірністю та вартістю.

Література

1. Кривуца В.Г. Управління телекомунікаціями із застосуванням новітніх технологій/В.Г.Кривуца, В.К.Стеклов, Л.Н.Беркман, Б.Я.Костік, В.Ф.Олійник, С.М.Скляренко//Підручник для ВНЗ.– К.: Техніка, 2007. – 384с.

2. Климаш М.М. Забезпечення відмовостійкості багаторівневої ієрархії управління у програмно-конфігурованих мережах / М.М.Климаш, М.О.Селюченко, О.А.Лаврів // Сучасні проблеми телекомунікацій і підготовка фахівців в галузі телекомунікацій: Матеріали конференції (м. Львів, 30 жовтня - 2 листопада 2014 р.). - Львів, 2014. - С.225-228.

Метод ідентифікації особи комп'ютерним зором

Ботвін В.Ю.
Науковий керівник: к.т.н. доц. Муляр І.В.
Хмельницький національний університет

Ідентифікація особи комп'ютерним зором застовується для аутентифікації користувача в системі. Здавалося б що на даний час є багато алгоритмів для ідентифікації, але постає необхідність оброки потокового відео з подальшим визначенням особи в режимі online для підтвердження користувача для подальшого надання доступу.

Рішенням даної задачі є модифікований метод гнучкого порівняння на графах[1], який зводиться до еластичного зіставлення графів, що описують зображення осіб. Особи представлені у вигляді графів зі зваженими вершинами та ребрами. На етапі розпізнавання один з графів – еталонний – залишається незмінним, в той час як інший деформується з метою найкращої підгонки до першого. У подібних системах розпізнавання графи можуть являти собою як прямокутну сітку, так і структуру, утворену характерними (антропометричними) точками особи. У вершинах графа обчислюються значення ознак, найчастіше використовують комплексні значення фільтрів Габора або їх впорядкованих наборів – Габорівских вейвлет (строї Габора), які обчислюються у деякої локальної області вершини графа локально шляхом згортки значень яскравості пікселів з фільтрами Габора (рис. 1).

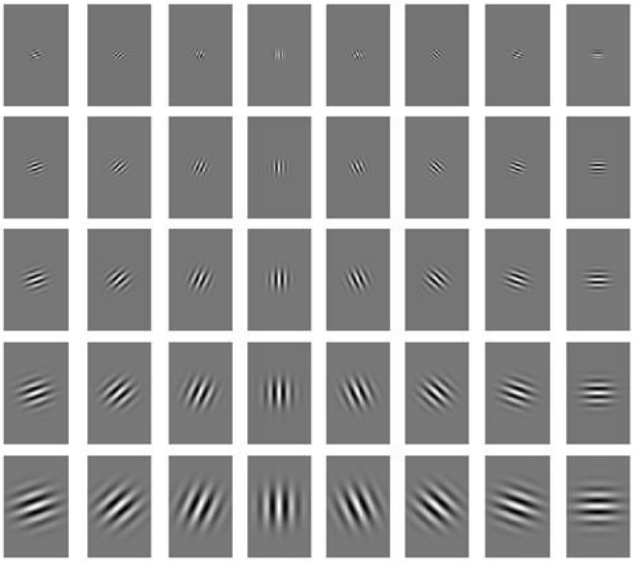

Рисунок 1 – Набір (банк, jet) фільтрів Габора

Ребра графа зважуються відстанями між суміжними вершинами. Різниця (відстань, дискримінаційна характеристика) між двома графами обчислюється за допомогою деякої функції цінової деформації, що враховує як розходження між значеннями ознак, обчисленими в вершинах, так і ступінь деформації ребер графа.

Деформація графа відбувається шляхом зміщення кожної з його вершин на деяку відстань в певних напрямках щодо її початкового місця розташування і вибору такої позиції, при якій різниця між значеннями ознак (відгуків фільтрів Габора) у вершині деформованого графа і відповідної їй вершині еталонного графа буде мінімальною. Дана операція виконується по черзі для всіх вершин графа до тих пір, поки не буде досягнуто найменша сумарна різниця між ознаками деформованого і еталонного графів. Значення цінової функції деформації при такому положенні графа, що деформується, і буде мірою відмінності між вхідним зображенням обличчя і еталонним графом. Дана «релаксаційна» процедура деформації повинна виконуватися для всіх еталонних осіб, закладених в базу даних системи. Результат розпізнавання системи – еталон з найкращим значенням цінової функції деформації.

В окремих публікаціях вказується 95-97% - а ефективність розпізнавання навіть при наявності різних емоційних виразах і зміні ракурсу обличчя до 15 градусів. Однак розробники систем еластичного порівняння на графах посилаються на високу обчислювальну вартість даного підходу. Наприклад, для порівняння вхідного зображення обличчя з 87 еталонними витрачалося приблизно 25 секунд при роботі на паралельній ЕОМ з 23 трансп'ютерами [2] (Примітка: публікація датована 1993 роком). В інших публікаціях з даної тематики час або не вказується, або кажуть, що воно велике.

Даний метод сегментації заснований на застосуванні серії фільтрів Габора. Відмінною особливістю даного фільтра є те, що він здатний виділяти прямі лінії певного розміру і під певним кутом.

Дійсна частина цього фільтра виглядає наступним чином:

$$g(x, y, \lambda, \psi, \sigma, \gamma) = exp\left(-\frac{x'^2+\gamma^2 y'^2}{2\sigma^2}\right) cos\left(2\pi\frac{x}{\lambda}+\psi\right)$$

де x,y – координати центру ядра в наперед заданих межах; λ – період ядра в пікселях; θ – нахил ядра; σ – дисперсія Гауссіана; ψ – зміщення фази ядра; γ – стиснення Гауссіана;

$$x = x\cos\theta + y\sin\theta;$$
$$y = x\sin\theta + y\cos\theta.$$

Таким чином, щоб виділити образ, потрібно застосувати фільтр Габора з різними кутами нахилу ядра і порахувати максимальний відгук кожного пікселя на серію фільтрів.

Оптимізація алгоритму досягається за рахунок обробки зображення

фільтром Габора шляхом усереднення значень оброблюваного зображення за деякою області в кожній точці. відповідно, накладення фільтра Габора на зображення має вид:

$$I`(x,y)=\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}I\left(x-\frac{n}{2}+i,y-\frac{n}{2}+j\right)G(i,j)$$

Підібраний фільтр Габора представлений на рисунку 2 і задається формулою:

$$Filter(x,y)=\exp\left(-\frac{x`^2+100y`^2}{2*49}\right)\cos(2\pi x`+90)$$

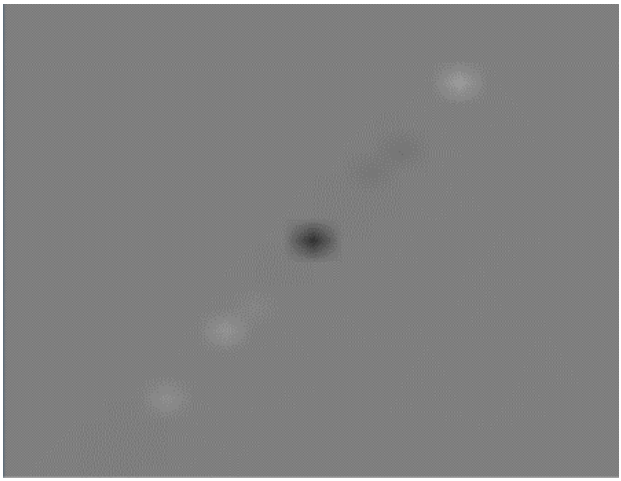

Рисунок 2 – Графічне представлення фільтру Габора

В результаті застосування даного фільтру вдалось виокремити характерні риси обличчя. Далі до результатів фільтрації необхідно застосувати алгоритми аналізу зображення. Наприклад алгоритм Фрімена[3], для виділення контуру зображення, відфільтрованого Габорівським фільтром, для виділення на вихідному зображенні обличчя.

Метод Габора має ряд незаперечних переваг перед більшістю інших: при досить високій точності визначення він дозволяє проводити перевірку на відстані, допускає таємну перевірку і вимагає наявності тільки відеокамери. Розроблено досить велике число алгоритмів, що забезпечують не тільки високу швидкодію і точність визначення, але і дозволяють системі працювати в самих різних умовах. Сукупність цих якостей зумовила дуже швидкий розвиток цього методу, поставивши його за поширеністю в один ряд з дактилоскопічної перевіркою.

Виникає необхідність модифікації методу для проведення

ідентифікації і авторизації за умов поганого освітлення та надання доступу для користувача.

Проаналізовано існуючі методи розпізнавання обличь на зображеннях, а також найбільш поширені методи фільтрації зображення. Продемонстровано доцільність застосування фільтрів Габора для фільтрації зображень з метою виділення на них обличь.

Наукова новизна дослідження полягає в розробці ефективного методу виявлення обличь з отриманням інформації про їх біометричні властивості. Сутність запропонованого методу полягає в фільтрації вихідного зображення, застосуванні порогової обробки, виділення контурів отриманого об'єкту на зображенні і підтвердження особи з білого списку.

Практичне застосування цього методу може бути використане як складову розумного будинку, «розумний замок», який ідентифікує особу і на основі проведеного аналізу оптичного зображення виносе рішення про надання чи ненадання доступу до помешкання.

Література

1. Face Recognition by Elastic Bunch Graph Matching [Electronic resource] URL:http://www.face-rec.org/algorithms/ebgm/wisfelkrue99-facerecognition-jainbook.pdf

2. Distortion invariant object recognition in the dynamic link architecture [Electronic resource] URL: https://ieeexplore.ieee.org/document/210173

3. Сообщество любителей робототехники Robocraft [Electronic resource] URL: http://robocraft.ru.

Автоматизація пошуку оптимального алгоритму поведінки агента з використанням нейронних мереж

Великий Я.О., Погудіна О.К.

Національний аерокосмічний університет ім. М.Є.Жуковського «Харківський авіаційний інститут»

В роботі описано метод навчання нейронних мереж на основі генетичного алгоритму. Наведено короткий огляд параметрів, що відрізняють нейронні мережі. Наведено архітектура розробленого програмного забезпечення. Розглянуто результати навчання за допомогою еволюційного алгоритму нейронних мереж з різними топологіями.

У сучасному світі все частіше застосовуються системи з автоматичним управлінням, починаючи від контролерів освітленості [1] закінчуючи безпілотними автомобілями і літальними апаратами. З часом підвищується не тільки ступінь автоматизації, але і ступінь автономності, незалежності пристроїв від людини. На цьому тлі зростає складність створення алгоритмів поведінки, особливо прийняття рішень. Наприклад, зараз перспективною

галуззю є автопілотовані автомобілі та літальні апарати, а також написання для них алгоритмів поведінки. Всі розглянуті приклади є складними завданнями, тому існує багато спільнот, що займаються відкритими або комерційними проектами розробки програмного забезпечення (ПЗ) в даних галузях.

Нейронна мережа (НМ), як модель обробки даних, що застосовується в ПЗ алгоритмізації поведінки складних об'єктів, займає лідируючі позиції по вживаності. Завдяки реалізованої можливості самонавчання НМ, зараз немає необхідності будувати математичну модель середовища і прописувати в ній всі варіанти прийняття рішень. А тому, зменшується обсяг робіт і проблема вирішення математично складних завдань.

У даній роботі буде розглянута можливість створення і використання НМ для управління групою автономних агентів, наприклад, як описано в [2]. Метою роботи є підвищення якості поведінки агента з урахуванням мінімізації часу його навчання. Для досягнення поставленої мети в роботі формуються і вирішуються такі задачі: аналіз предметної області; створення моделі середовища функціонування агента; реалізація алгоритму навчання НМ у вигляді ПЗ, визначення критеріїв ефективності навчання; тестування розробленого ПЗ відповідно до обраних критеріїв.

Розглянуто принципи побудови і навчання НМ. Розглянуто різні варіанти топологій НМ і обрано найбільш підходящі з них для вирішення поставленого завдання. Детально описано особливості використання генетичного алгоритму для навчання НМ. За обраними методиками і алгоритмами створена UML діаграма архітектури програмного забезпечення [3]. На підставі спроектованої архітектури описано алгоритми, які були реалізовані у вигляді програмного забезпечення, призначеного для автоматизованого проведення експериментів з навчання НМ.

Було зроблено тестування роботи програми. В ході тестування були виявлені недоліки обраного алгоритму:

- залежність від випадкових величин і від початкових даних;
- великі витрати за часом.

У плюси обраних алгоритмів слід віднести: наявний потенціал, а також те, що навчені з їх допомогою мережі легко пристосувати для дій при змінних умовах середовища.

Література

1. Великий Я. А. Анализ принципа распознавания объектов на изображении методом Виолы–Джонса / Я. А. Великий // Открытые информационные и компьютерные интегрированные технологии. – 2015. – Вып. 68. – С. 162-166. – Режим доступу: http://nbuv.gov.ua/UJRN/vikt_2015_68_22.

2. Погудина О. К. Разработка имитационной модели взаимодействия беспилотных летательных аппаратов для исследования возможности

совместного полета [Текст] / О. К. Погудина // Системи обробки інформації, 2012, випуск 7 (105). – С. 140-145. – Режим доступу: file:///I:/vidnew/soi_2012_7_30.pdf

3. Великий Я. А. Автоматизация поиска оптимального алгоритма поведения агента с использованием нейронных сетей / Я. А. Великий // Открытые информационные и компьютерные интегрированные технологии. – 2017. – Вып. 72. – С. 219-223. – Режим доступу: http://nbuv.gov.ua/UJRN/ vikt_2015_68_22.

Інформаційна система моніторингу та управління вуличним освітленням

Герус В. В., Лажевський В. І.
Науковий керівник – ст. викладач Петросян Р. В.
Житомирський державний технологічний університет

В даний час міжмашинна взаємодія (M2M) та інтернет речей стала досить поширеною концепцією [1]. Її основна ідея полягає в тому, що глобальна мережа інтернет стала вже не просто глобальною мережею для спілкування людей за допомогою комп'ютерів, а й середовищем для пристроїв, що дозволяє їм взаємодіяти між собою та з навколишнім світом.

M2M — загальна назва технологій, які дозволяють машинам обмінюватися інформацією одна з одною. Це можуть бути дротові та бездротові системи моніторингу датчиків або будь-яких параметрів пристроїв. Наприклад, банкомати або платіжні термінали, які можуть передавати інформацію по GSM-мережам про відсутність готівки або ж навпаки. M2M дозволяє об'єднати різноманітне обладнання, що використовувалось раніше локально і автономно, з різними протоколами взаємодії між собою та єдиним протоколом доступу до глобальної мережі.

«Розумне місто» – концепція інтеграції декількох інформаційних і комунікаційних технологій та інтернету речей для управління міським майном: місцеві відділи інформаційних систем, бібліотеки, транспорт, лікарні, електростанції, система управління світлом та інші служби. Метою створення «розумного міста» є поліпшення якості життя за допомогою інформаційних технології для підвищення ефективності обслуговування і задоволення потреб громадян. За рахунок використання датчиків, інтегрованих в режимі реального часу, накопичені дані обробляються та аналізуються.

Мережі вуличного освітлення є прикладом системи, що відіграє важливу роль в структурі комунального господарства, яка на сьогоднішній день ще не набула достатнього рівня технологічного розвитку. В багатьох країнах рівень контролю систем вуличного освітлення обмежується

функцією ручного включення та виключення окремих ліній світильників. В таких системах відсутній зворотній зв'язок, а контроль працездатності світильників відбувається візуально. Такий підхід не зручний і не ефективний, оскільки з моменту виходу з ладу світильника до моменту виявлення даної події проходить досить багато часу, в результаті чого може виникнути ситуація поганого освітлення на певних ділянках вулиць.

Для автоматизації включення і виключення вуличного освітлення найчастіше використовують датчики рівня освітленості [2]. Принцип роботи таких систем гранично простий: при зниженні рівня яскравості нижче заданого порогу лампи включаються та виключаються при перевищенні порога спрацьовування. Однак даний підхід не вирішує всіх поставлених задач. Недоліками таких систем є труднощі калібрування датчиків, чутливість датчиків до забруднення, неможливість реалізації енергозберігаючих алгоритмів роботи (наприклад, виключення частини ламп в глухий нічний час, коли повне освітлення не потрібне).

Для регулювання яскравості на сьогоднішній день в Україні активно використовується система управління вуличним освітленням з комутацією силових ліній. Дана система працює наступним чином: на вулиці встановлюється спеціальний блок комутації. З даного блоку виходять декілька силових лінії, до яких підключено світильники. Такі лінії можуть розподілятися по вулицях, районах та дорогах між містами. Отримавши вхідну команду, блок комутації може включити або виключити відповідну силову лінію (рис. 1).

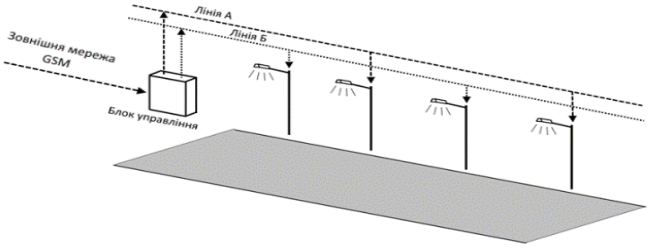

Рисунок 1 – Метод регулювання яскравості комутацією силових ліній

Для підвищення ефективності управління вуличним освітленням пропонується створення мережі світильників, яка надавала б можливість віддаленого моніторингу та управління шляхом передачі даних по радіоканалу (рис. 2).

У кожний світильник вбудовується модуль контролю LCM (Luminary control module), група таких модулів об’єднується у мережу, яку в свою чергу обслуговує модуль NSC (Network segment controller). Усі вхідні та вихідні дані відправляються на сервер з модуля NSC по каналу GSM.

Рисунок 2 – Загальна системи управління вуличним освітленням

Для перевірки роботи запропонованої інформаційної системи моніторингу та управління вуличним освітленням було створено діючій макет (рис. 3).

Всі модулі LCM та концентратори NSC знаходяться на одній частоті (868 MHz) з потужністю 25 мВт. Випробування проводилось на різних конфігураціях мережі (рис. 4).

На даному етапі досягнуто стабільну роботу макету системи управління вуличним освітленням.

Рисунок 3 – Демонстраційний стенд

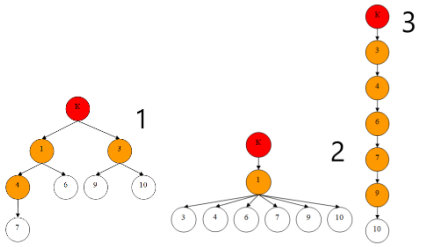

Рисунок 4 - Конфігурації мереж, які використовувались при тестуванні основних функцій системи

Література

1. Межмашинное взаимодействие [Электронный ресурс]. – Режим доступа к ресурсу: URL : https://ru.wikipedia.org/wiki/Межмашинное взаимодействие. – Загл. с экрана. – Дата обращения: 03.12.2017.

2. Управление уличным освещением — принципы и устройство [Электронный ресурс]. – Режим доступа к ресурсу: URL : http://elektrik-a.su/elektricheskoe-osveshhenie/obshhaya-chast/upravlenie-ulichnym-osveshheniem-384. – Загл. с экрана. – Дата обращения: 04.12.2017.

Реалізація штучного інтелекту для покрокової стратегічної гри

Глинська К.С.
Науковий керівник – к.т.н., доц. Костюкова Н.С.
Донецький національний технічний університет

Алгоритми та евристики, що використовуються для імітації дій шахового гравця, є поширеними засобами реалізації штучного інтелекту в багатьох ігрових та інших додатках. Не існує єдиного алгоритму гри в шахи, але є декілька базових методів та прийомів, оптимальна комбінація яких дозволить створити конкурентоздатну програму. Будь-яка шахова програма повинна реалізовувати пошук та моделювання можливих ходів гравців на деяку глибину та оцінку отриманих кінцевих позицій для їх порівняння та вибору кращого варіанту. Від того, які алгоритми та методики використає програміст для генерації ходів та оцінки ситуації на полі, залежить рівень штучного інтелекту його гри.

Дослідження та розробка штучного інтелекту відбувалась на основі створення програмного модуля для гри в шахи.

Проблема створення шахової програми полягає в тому, щоб знайти оптимальну «золоту середину» між швидкістю та якістю штучного інтелекту. Якщо кількість можливих позицій в шахах після першого кроку білих дорівнює 20, то це число стрімко зростає і після 6 ходів (3 ходу білих і 3 ходу чорних) складає 9 132 484 різних або 120 921 506 всього можливих позицій [1]. Повне дерево перебору в шахах, хоча і є кінцевим, містить приблизно 10120 позицій. Звичайно, формувати та аналізувати все дерево варіантів неможливо, але це і не потрібно. Шахіст рахує варіанти в середньому на 5-6 ходів, тому досить навчити програму будувати й оцінювати ходи на обмежену глибину, знаходити цікаві й відкидати некорисні, відсікаючи зайві гілки дерева рішень. Існує декілька модифікацій та доповнень до загального алгоритму шахової програми, які дозволяють уникнути повного перебору. Автором було реалізовано алгоритм NegaMax та його оптимізацію – альфа-бета відсікання для пошуку кращого ходу; алгоритм пошуку з нульовим вікном та NegaScout; евристика нульового ходу; PV-таблиці, Zobrist-ключи та хеш-таблиці для запам’ятовування кращих стратегій гри та оцінок позицій; алгоритм ітераційного занурення та алгоритм пошуку спокою при побудові дерева рішень; принцип MVV/LVA для сортування захоплень фігур; евристики вбивця та історії для сортування «тихих» ходів; алгоритми матеріальної та позиційної оцінки; подання дошки через бітборди, генератори

ходів, атак, магічні бітборди.

Для перебору варіантів ходів використано дерева рішень: кожному вузлу відповідає деякий хід гравця, рівні дерева – це послідовні ходи «білих» та «чорних», кожен шлях дерева, від вершини до листа, позначає деяку стратегію гри. Кожен лист дерева отримує оцінку – ступінь привабливості такого ходу для гравця. Рекурсивна обробка дерева рішень дозволяє обрати кращий поточний хід, проаналізувавши результати різних стратегій гри. Автором використано алгоритм альфа-бета-відсікання – оптимізацію алгоритму мінімакс, завдяки якій деякі гілки пошуку ігноруються. З метою оптимізації виконується сортування ходів за принципом MVV/LVA (Most Valuable Victim – Least Valuable Aggressor, найбільш цінна жертва – найменш цінний нападаючий), коли спочатку розглядаються потенційно кращі ходи [2]. Після генерації можливих ланцюжків ходів оцінюється результат – поточна ситуацію на полі. Існує матеріальна та позиційна оцінка. При формуванні позиційної оцінки можуть надаватися бонуси або штрафи за такі умови ігрової ситуації, як кількість полів під боєм своєї сторони та сторони суперника; зайняття ключових позицій: у центрі дошки для коня або ферзя, відкритих прямих для тур, головних діагоналей для слона; наявність прохідних пішаків; наявність заблокованих (здвоєних) пішаків; здійснення рокіровки; зменшення рівня безпеки короля, відстані між королем та ворожим ферзем; захист своїх фігур, x-ray позиція – розташування ферзя та слона на одній діагоналі під взаємним захистом посилює фігури та сприяє атаці; кількість можливих для гравця ходів та інше.

Крім того, в програмі реалізовано, з одного боку, ефект горизонту – обмеження кількості ходів, що розглядаються, і, з іншого боку, процес збільшення глибини дерева, так званий «пошук спокою» (Quiescence Search) [3]. У інших випадках генерація варіантів завершується або по досягненні заданої глибини пошуку, або по закінченню виділеного часу.

Алгоритм пошуку кращого ходу із використанням цих модифікацій наведений на рисунку 1.

В процесі тестування корегувалися матриці позиційних оцінок, максимальний час та глибина пошуку, використані евристики й їх порядок. Наприклад, евристика Aspiration Search поруч з іншими використаними евристиками не мала вагомого внеску, тому була опущена в кінцевому алгоритмі програми. Результати тестування впливу деяких евристик на швидкість перебору варіантів наведені в таблиці 1.

Таблиця 1 – Вплив різних евристик на роботу алгоритму

	Null Move	Nega Scout	PVS	Killer heuristic	History heuristic
Прискорення	56%	38%	74%	14%	17%

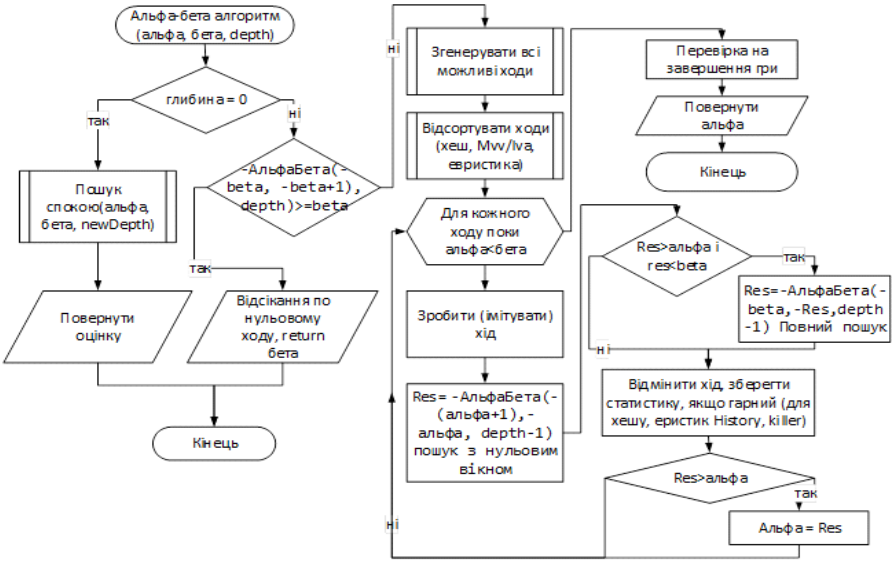

Рисунок 1 – Блок-схема алгоритму пошуку кращого ходу

Результати тестування (гри з людиною та іншими шаховими програмами, пошук кращого ходу для тестового набору позицій, рішення класичних шахових задач на постановку мату в задану кількість кроків та інші) показують, що програма відповідає заявленим вимогам користувача від неї та дозволяє гравцям різного рівня відточувати навички у грі в шахи з комп'ютером. Вдосконалювати програму можна, використовуючи готові бази даних (бібліотеки) дебютів та ендшпілів [4] або створити власні, змусивши движок навчатись на проведених іграх. Також можна виконувати навчання з використанням нейронних мереж (успішний приклад такого використання – програма AlphaZero [5]).

В результаті роботи були досліджені існуючи методи реалізації шахового штучного інтелекту, розроблений та протестований шаховий ігровий додаток. З метою вдосконалення розробленої програми були обрані основні засоби його підсилення, зокрема, досліджені можливості впровадження нейронної мережі й самонавчання шахового штучного інтелекту.

Література

1. Лысенко А. В. Оценка позиции. Компьютерные шахматы / А..В._Лысенко, Е. Я. Гик. – М.: ФиС, 1990. – 176 с.

2. Дамский Я. В. Искусство шахмат. По законам красоты / Я. В.

Дамский. – М.: Рипол Классик, 2005. – 152 с.

3. Heinz E. Scalable Search in Computer Chess: Algorithmic Enhancements and Experiments at High Search Depths / E. Heinz. – Vieweg, Verlag, 1999. – 270_с.

4. Ананд В. Энциклопедия шахматных дебютов / В. Ананд. – Н.: Кипр, 1993. – 178 с.

5. AlphaZero, новый проект Google, громит Stockfish в матче из 100 партий [Електронний ресурс]. – Режим доступу: https://www.chess.com/ru/news/view/alphazero-novyi-proiekt-google-ghromit-stockfish-v-matchie-iz-100-partii

Модель прихованих загроз інформаційній безпеці в системах з використанням хмарних технологій

Глінський О.В.
Науковий керівник: к.т.н. доц. Чорненький В.І.
Хмельницький національний університет

Характерною особливістю сучасного середовища хмарних обчислень є активний характер суб'єктів і об'єктів інформаційної взаємодії. Це дозволяє розглядати цільову функцію системи безпеки як збереження конфіденційності, цілісності і доступності програмних і інфраструктурних сервісів, що надаються в режимі видаленого доступу в умовах динамічної зміни стану обчислювальних ресурсів. Побудова перспективних механізмів забезпечення безпеки в середовищі хмарних обчислень зв'язується не із захистом від виявлених вразливостей, а полягає в можливості запобігання новим невідомим методам проведення атак, в розробці нових моделей загроз і методів запобігання або віддзеркалення комп'ютерних атак на інформаційні ресурси, які використовують можливості предикативної ідентифікації прихованих каналів і потенційно небезпечних процесів інформаційної взаємодії [1].

Перспективним напрямком вирішення сформульованої задачі є використання технології між мережевого екранування з урахуванням специфіки захищеності середовища [2]. Для цього необхідна формалізація вимог розмежування доступу до інформаційних сервісів. Така формалізація може бути представлена з використанням динамічно формованого набору правил фільтрації, що забезпечує виконання вимог політики доступу. При цьому зростаюча складність алгоритмів фільтрації пред'являє високі вимоги до продуктивності між мережевих екранів, що робить необхідним використання методів паралельної обробки віртуальних з'єднань за допомогою віотуальних машин. У сучасній літературі підхід до створення складних технічних систем, зв'язаність яких забезпечується за рахунок організації процесів обміну інформацією з мережі, отримав назву мережево-центричний. Цей підхід стосовно задачі розмежування доступу вимагає

забезпечення ситуаційної обізнаності та локальності дій кожного з між мережевих екранів, що входять до складу віртуальних машин, які використовуються в СХО для реалізації політики доступу [3].

Важливим напрямом вдосконалення технологій захисту і систем інформаційної безпеки є протидія білатеральним загрозам, в яких суб'єкт і об'єкт процесів інформаційної взаємодії є потенційним носієм небезпечних дій. У таких випадках необхідно використовувати моделі загроз, які ідентифікують потенційні вразливості як на рівні процесів контролю доступу до ресурсів гостьових операційних систем (ОС) або додатків, так і на рівні системних викликів гіпервізора, який сам може стати джерелом руйнуючих дій що реалізовуються шляхом порушення функціонування планувальника завдань або диспетчера устаткування. Загрози, що виникають при цьому, необхідно не тільки оперативно виявляти, але і блокувати використовувані неавторизовані канали інформаційних дій, які в середовищі хмарних обчислень зазвичай реалізуються в прихованому для гостьових ОС режимах. Тому важливим чинником підвищення ефективності систем захисту від прихованих загроз є облік напряму передачі, синтаксису і контексту потоків даних, які передаються [4].

З врахуванням вищесказаного, захист від загроз, які можуть приводити до розкрадання даних, неконтрольованої модифікації програмного коду, порушенню доступності (блокуванню) або нав'язуванню помилкової інформації в середовищі хмарних обчислень є актуальним науково-технічним завданням, вирішенню якого присвячена дана магістерська робота [2].

Використання традиційних підходів не дозволяє вирішити проблему підвищення рівня захищеності середовища хмарних обчислень з урахуванням гнучкості, масштабованості (підтримка апаратних платформ різного класу) пропонованих програмно-технічних рішень і мінімізації витрат.

Застосування сучасних технологій адаптивних систем захисту інформації не дозволяє здійснювати «прозорий» контроль за інформаційними потоками середовища хмарних обчислень, оскільки вони функціонують на верхніх рівнях ієрархії.

Класичні методи пошуку шкідливого програмного коди не дозволяють виявляти нові зразки шкідливого ПО, що реалізовує технології DKOM і VICE, оскільки вони вбудовуються в операційну систем на «нижчому» рівні, ніж модулі адаптивних систем захисту.

Традиційні методи перехоплення системних функцій гостьових ОС не дозволяють виявляти програмні «закладки», що вшиваються в ОС на етапі завантаження.

Для боротьби з такими загрозами актуальною є розробка нових засобів захисту інформації, заснованих на методах оперативної ідентифікації потенційних вразливостей, що виникають як на рівні процесів контролю доступу до ресурсів гостьових ОС, так і на рівні системних викликів гіпервізора, які за певних умов самі можуть ставати джерелами різних видів

руйнівних впливів. Метою дослідження є підвищення рівня захищеності обчислювальних систем на основі розробки моделей, методів і алгоритмів протидії прихованим загрозам в середовищі хмарних обчислень.

Для досягнення поставленої мети вирішуються наступні завдання:

– розроблена модель прихованих загроз інформаційній безпеці в середовищі хмарних обчислень, що враховує активний характер суб'єктів і об'єктів інформаційної взаємодії.

– розроблена модель операцій, що відбуваються з даними при їх обробці в середовищі хмарних обчислень, що дозволяє формалізувати опис інформаційних процесів у вигляді мультиграфа транзакцій.

– розроблений метод протидії прихованим загрозам з використанням запропонованої моделі операцій, заснований на характеризації ієрархії транзакцій.

– розглянемо компоненти гіпервізора як джерело загрози при проведенні атак зловмисником з подальшим розповсюдженням шкідливого програмного забезпечення на серверах віртуалізації.

Користувачі можуть атакувати компоненти гіпервізора, посилаючи некоректні запити на обробку модулям програмного забезпечення гіпервізора і використовуючи недокументовані можливості системного і прикладного програмного забезепечення, встановленого на серверах віртуалізації. Логіка виконання програм повинна контролюватися з точки зору відмови в обслуговуванні. Це підвищує ризики при реалізації прихованих загроз, не тільки функціональних можливостей, але і безпеки, яка оцінюється величиною ризику їх не документованої роботи. Приховані загрози, що приводять до порушення роботи середовищі хмарних обчислень, реалізуються за допомогою дій з боку шкідливого програмного забезпечення, від яких немає захисту на рівні гостьової ОС].

Під реалізацією прихованих загроз маються на увазі використання механізмів створення і зміни контексту виконання потоків, за допомогою яких можуть передаватися дані від сутностей з високим рівнем безпеки до сутностей з низьким рівнем безпеки в обхід правил і може порушуватися стан захищеності самого гіпервізора.

Гіпервізор забезпечує ізоляцію різних ОС одна від одної, розділення і управління ресурсами. Гостьові ОС – це операційні системи віртуальних машин, що запускаються під управлінням гіпервізора.

У гіпервізорі, як і в будь-якій операційній системі, створюється множина сутностей (об'єктів і суб'єктів доступу) з різним рівнем безпеки. Операція породження суб'єктів $Create\left(Subi, Om\right) \to Subj$ називається породженням з контролем незмінності об'єкту, якщо для будь-якого моменту часу $t > t0$, в який активізована операція породження об'єкту $Create$, породження об'єкту $Subj$ можливо тільки при тотожності об'єкту-джерела

щодо моменту $t0: Om[t] = Om[t0]$, де Sub – суб'єкт, O – об'єкт доступу. У разі середовища хмарних обчислень суб'єкти і об'єкти доступу можуть мінятися ролями [8].

Тому для протидії прихованим загрозам в середовищі хмарних обчислень, в якому діє породження суб'єктів з контролем незмінності об'єкту, необхідно, щоб у момент часу $t0$ через будь-який суб'єкт до будь-якого об'єкту існували тільки потоки, що не суперечать умові коректності: монітор безпеки повинен реалізувати спеціальні механізми ідентифікації контексту контрольованих потоків даних як для суб'єктів, так і для об'єктів доступу, а будь-який суб'єкт доступу (ініціатор доступу) повинен використовувати тільки дозволені механізми доступу. З цією метою вводиться набір який підходить для створення об'єктів доступу, так і при породженні об'єктів у вигляді кортежу $(s, Ord, Context\_type)$.

Таблиці дозволених зв'язків об'єктів і суб'єктів доступу, за допомогою яких здійснюється контроль транзакцій операцій породження нових об'єктів, необхідно розширити на випадок прихованих загроз.

Предикативна функція ідентифікації прихованих загроз - це відображення 8-рівневої моделі операцій на множину його можливих станів - небезпечні, безпечні і невизначені. В цьому випадку модель прихованих загроз описується у вигляді розширеного кортежу:

$$M = \langle Source, Services, Devices, \{proc\}, Actions, \{hv\}, \{vm\}, Security\ Roles \rangle$$

де $Source$ - суб'єкт доступу або процес, джерело загрози; $Services$ - набір шаблонів правил безпеки, використовуваних традиційними СЗІ (наприклад, правила фільтрації для МСЕ тощо); $Devices$ - пристрої, що встановлені на серверах віртуалізації і використовувані гостьовими операційними системами ВМ (диск, мережний контролер тощо), як об'єкт доступу; $\{ proc \}$ - множина суб'єктів впливу (шкідливий код гіпервізора, несертифіковані засоби віртуалізації і. т. п.); $Actions$ - (дії) виконання операцій суб'єктом по відношенню до об'єкту доступу (виконання команд read, write, append, create, execute…); $\{hv\}$ - середовище взаємодії процесів ВМ у гіпервізорі, що представляє собою множину компонентів $\mathrm{mod} i$; $\{vm\}$ - об'єкти впливу (множина ВМ); $Security\ Roles$ - процедури багаторівневої рольової ПБ для протидії прихованим загрозам, які реалізуються у вигляді набору міток безпеки. Набір міток являють собою значення кортежу $(s, Ord, Context\_type)$.

В рамках пропонованої моделі загроз середовище хмарних обчислень розглядається як система взаємодії гіпервізорів, встановлених на серверах віртуалізації. В рамках направленої схеми «суб'єкт-дія-об'єкт» активний характер суб'єктів і об'єктів інформаційної взаємодії передбачає ту

обставину, що вони можуть мінятися місцями. Розглянемо ситуацію, в якій зловмисник(суб'єкт) атакує сервер віртуалізації(об'єкт), модифікує компоненти гіпервізора шляхом реалізації нових загроз, приведених в таблиці

У будь-якій обчислювальній системі існують інтерфейсні рівні взаємодії між різними модулями(компонентами), що дозволяють використовувати недокументовані можливості, з одного боку, для проведення атак зловмисником, з іншої – для реалізації механізмів моніторингу з боку систем контролю і захисту ПЗ середовища хмарних обчислень.

Загроза порушення доступу до конфіденційної інформації породила необхідність розробки нових методів захисту ПЗ та предикативного алгоритму на основі розробленої моделі операцій, що допомагає систематизувати функціональні рівні, використовувані зловмисником для вбудовування до гостьової ОС і гіпервізора, і протидіяти впровадженню шкідливих кодів та загроз, які формують послідовності запитів до некоректних програмним модулів гіпервізора або використовують недекларовані можливості системного і прикладного програмного забезпечення. Різні компоненти гіпервізора розглядаються в якості потенційного джерела загроз кібербезпеці, які реалізується шляхом поширення шкідливого програмного забезпечення або ініціалізації процесів, що руйнують стан захищеності ресурсів середовища хмарних обчислень.

Література

1. Моляков А.С. KPROCESSOR_CID_TABLE факторинг – новый метод в теории компьютерного анализа вирусного кода и поиска программных закладок / А.С. Моляков // Проблемы информационной безопасности. Компьютерные системы. - СПб.: Изд-во Политех. Ун-та, 2009. - №1. - с. 17-19.

2. Козак І.В Аналіз проблем захисту інформації в середовищі хмарних обчислень / І.В. Козак, С.О. Пашков, О.В. Огнєвий // Збірник наукових праць Військового інституту Київського національного університету імені Тараса Шевченка. – К.: ВІКНУ, 2016. – Вип. № 51. – С.177-185.

3. Гладких А.А. Концептуальная модель функционирования обманной системы в условиях информационного противоборства. / А. А. Гладких, Р.Р. Зелимов // Сб. рефератов депонированных.- М: ЦВНИ МО РФ, 2004. - с. 12-15.

4. Муляр І.В. Розробка математичної моделі та методу її вирішення для підвищення ефективності використання обчислювальних ресурсів на основі технології віртуалізації / І.В. Муляр , Г.В. Гусляков , Л.В. Солодєєва // Збірник наукових праць Військового інституту Київського НУ імені Тараса Шевченка. – К.: ВІКНУ, 2016. – Вип. № 54. – С. 134-143.

Оцінка інформаційної ефективності мережевих інформаційних систем на основі кібернетичної потужності

Глушанець О.М.
Науковий керівник – к.т.н. доц. Огнєвий О.В.
Хмельницький національний університет

Оцінка інформаційної ефективності заснована на якості мережевої інформаційної системи, що враховує як швидкість, так і накопичувальні можливості системи.

Особливістю оцінки інформаційної ефективності є аналіз характеристик і показників мережевої інформаційної системи, що дає кількісне уявлення якості передачі інформаційних потоків в мережевій інформаційній системі з урахуванням обмеження на час доставки пакетів до адресатів з одного боку, з іншого – визначення такої характеристики, яка не тільки могла б оцінювати ступінь близькості системи до еталону, але і була б основою безпосереднього вирішення задачі розподілу інформаційних потоків, і як результат, забезпечувала вирішення завдання підвищення інформаційної ефективності навантаженої мережевої інформаційної системи в сенсі передачі інформації [1].

Ефективність передачі інформації в мережевій інформаційній системі, в силу випадкових вхідних потоків, оцінюється ймовірнісно-часовими характеристиками. В основному, ефективність мережевої інформаційної системи в сенсі передачі інформації, тобто інформаційна ефективність, оцінюється опосередкованими показниками: продуктивністю, тимчасовою затримкою переданих пакетів, коефіцієнтом втрат, ймовірністю доставки пакетів до адресатів з заданим тимчасовим обмеженням та іншими окремими величинами. Для проведення порівняльної оцінки декількох мереж можуть використовуватися відносні характеристики, що включають вищенаведені показники, при цьому показники однієї з систем розглядаються як еталонні або опорні. Використання сукупності оцінюваних величин дозволяє більш детально характеризувати ефективність інформаційного обміну в мережевій інформаційній системі. При необхідності визначення ступеня близькості мережевої інформаційної системи до граничних можливостей передачі інформації вводиться модель ідеальної системи [2].

Завдання оцінки інформаційної ефективності мережі вирішується на основі введеного узагальненого параметра - кіберпотужності інформаційної мережі, яка одночасно враховує як швидкісні, так і накопичувальні можливості мережевої інформаційної системи, виходячи з обмежень на тимчасову затримку інформаційних пакетів. За допомогою поняття ідеальної мережі визначено показник коефіцієнт корисної дії в сенсі передачі інформації. Він показує кількісні відмінності в ефективності роботи реальної мережевої інформаційної системи (її мат. моделі) і моделі ідеальної мережевої інформаційної системи (відповідно з поняттям примітивної мережі Г. Крона) [3].

Завдання оцінки інформаційної ефективності МІС шляхом оптимізації розподілу потоків передбачає наявність як мінімум один альтернативний шлях. У зв'язку з цим, важливим є визначення знаходження топологій з випадковою структурою і характеристиками слабкою і сильною її зв'язністю [4].

Для моделювання обрано мережу з 30 вузлів. В якості сильної зв'язності мережі з довільною топологією обрана структура, в якій 50% вузлів можуть виступати сусідами. У запропонованій моделі це буде відповідати k - зв'язності k=15. Для визначення слабкої зв'язності, за умови, що в мережі вузли можуть виходити з ладу, необхідно, як мінімум, два можливих (основний і альтернативний) шляхи. Це означає, що k - зв'язність k =2 може служити, з одного боку, характеристикою слабкої зв'язності, з іншого – забезпечувати підвищену структурну стійкість [5].

Задача моделювання полягає у визначенні ймовірності зв'язку між будь-якими двома вузлами мережі, яка б дозволила формувати мережу з заданими значеннями k - зв'язності мережі. Як МІС з випадковою топологією можуть бути обчислювальні системи з інформаційно-бездротовою системою доступу, яка використовує mesh-технології.

На рисунку 1 наведено кількісні залежності між ймовірністю існування зв'язку p і параметром k - зв'язності.

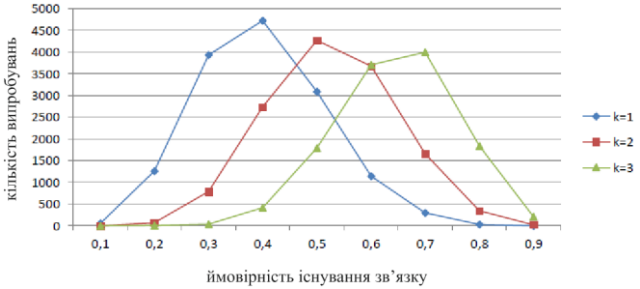

Рисунок 1 – Залежність k - зв'язності та ймовірності існування зв'язку k (низька зв'язність)

Для мереж слабкої зв'язності мінімальним значенням k - зв'язності, що забезпечують надійну передачу даних, є варіант 2. Як видно з наведеного графіка, мінімальне значення ймовірності, при якому існує значуща кількість топологій при випадковому моделюванні, які задовольняють відразу двом умовам (k і p), становить p=0,3.

Під мережею сильної (високої) зв'язності буде розумітися мережа k-зв'язність, якої прагне до $N/2$, де N – кількість вузлів в мережі. Побудова мереж більшої k - зв'язності є вкрай витратним, з точки зору ресурсів, завданням.

Для мережі з 30 вузлів висока зв'язність (рисунок 2) визначається значенням $k = N/2$. Для досягнення цього значення достатньо задати p=0,7.

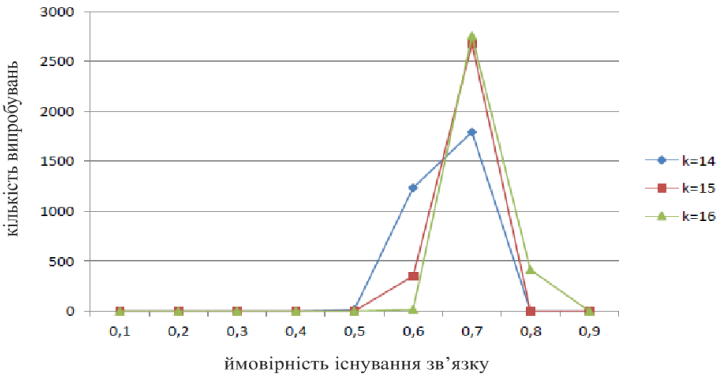

Рисунок 2 – Залежність *k* - зв'язності та ймовірності існування зв'язку *p*

У результаті порівняльного аналізу і моделювання та аналізу зв'язності випадкової топології MIC з 30 вузлів показало: для слабо зв'язних топологій, в яких для кожної пари є мінімум два шляхи передачі пакетів (2-зв'язкова CIS) значення ймовірності зв'язку p=0,3 є мінімально-необхідним; для забезпечення сильної зв'язності (15-зв'язкова MIC) – p =0.7.

Будь-який вузол комутації спільно з вихідним в напрямі адресата КЗ можна представити у вигляді послідовно з'єднаних ОС (рисунок 3).

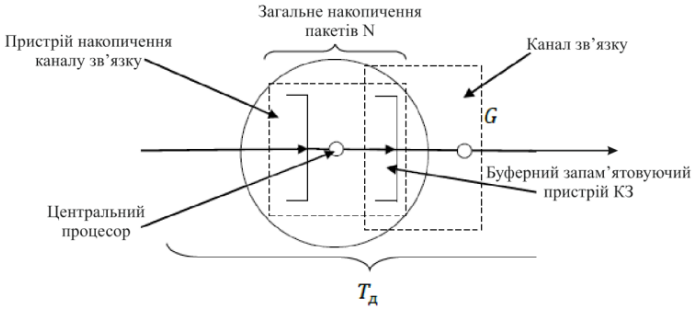

Рисунок 3 – Модель вузла комутації з вихідним КЗ на основі використання ОС

Перша ОС характеризується ПН для вхідних пакетів, а в якості обслуговуючого пристрою розглядається процесор ВК. Друга ОС відповідає окремому цифровому КЗ, тобто КЗ з пам'яттю. Так як у цифровому КЗ пристрій пам'яті використовується, по суті, для узгодження швидкості передачі інформації, тобто у вигляді буфера, то його надалі будемо називати буферний запам'ятовуючий пристрій (БЗП) каналу зв'язку. В виду того, що швидкодія цифрового процесора значно вище швидкодії КЗ, тобто його пропускної здатності, а пакет, вилучений з ПН ВК БЗП каналу зв'язку, є

одним і тим же, то загальна кількість інформації в КК можна характеризувати у вигляді значення N . Черги пакетів в ВК формуються відносно відповідних вихідних каналів зв'язку. У зв'язку з цим, сегмент МІС, наприклад, з трьох ВК, можна представити у вигляді сполучених ОС відповідно до його топології і шляховими потоками.

Кібернетична потужність ОС має вигляд:

$$P_{OC} = NG\,|_{T_Д}$$

де *N* – максимальна сукупна кількість інформаційних пакетів у системі, *G* – продуктивність системи, *ТД* – максимально допустимий час для обробки інформаційних пакетів у МІС.

Топологія МІС описується графом *G(V,E)*, який формується випадковим чином, шляхом використанням параметра *p* – ймовірність існування зв'язку між вузлами МІС. Додатковим обмеженням при остаточному виборі графа для моделювання служить параметр *k* –зв'язність. Це обмеження важливо для випадку слабкої зв'язності МІС.

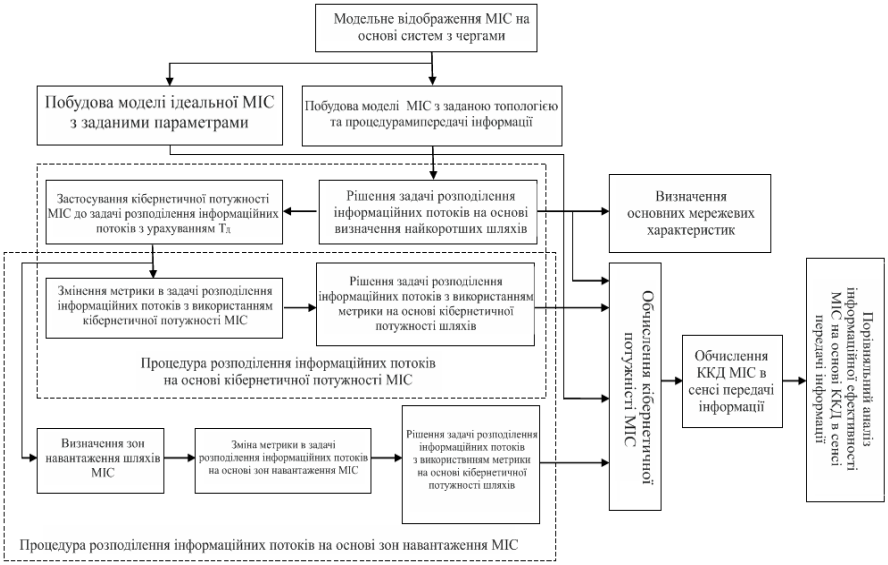

Рисунок 4 – Структура основних елементів процедурного методу МІС

Проведено аналіз задачі розподілу інформаційних потоків у МІС з метою визначення особливостей її вирішення, умов застосування та оптимізації, в результаті якого показана інформаційна ефективність задач визначення найкоротших шляхів на графі в умовах слабкого інформаційного навантаження і необхідність застосування кібернетичного параметра при рішенні задачі в умовах підвищеного інформаційного навантаження.

Література

1. Абрамов Г.В. Моделирование передачи данных по каналу конкурирующего доступа в системах реального времени / Г.В. Абрамов., А.Е. Емельянов // Вестник Воронежского государственного университета. Серия: Системный анализатор и информационные технологии, №4, 2014. – 26-31с.

2. Литвинов К.А. Оценка информационной эффективности телекоммуникационной сети со случайной топологией и разным числом узлов / К.А. Литвинов, И.И. Пасечников // Вестник Тамбовского университета. Серия: Естественные и тех. науки, Т.19, №2, 2014.–399-407с.

3. Пасечников И.И. Анализ и методы повышение информационной ефективности теллекомуникационных систем и сетей : монография. / И.И. Пасечников // – Тамбов: Изд. дом ТГУ им. Г.Р. Державина, 2010. – 118 с.

4. Пасечников И.И. О распространении строго детерминированного подхода на информационные сети / И.И. Пасечников, В.Ф. Войцеховский, Т.Я Гораздовский : сб. Математическое моделирование технологических систем. – Воронеж : ВГТА, 1999. – №3. – С. 36-43.

5. Петров М.Н. Исследование характеристик распределенных телекоммуникаций методом тензорного анализа и теории массового обслуживания: – Красноярск: КГУ,1998. – 240 с.

Управління технологічним процесом термообробки металу на основі моделювання його вхідних параметрів

Гузенко Д.В.
Науковий керівник – к.е.н., доц. Шевченко Н.Ю.
Донбаська державна машинобудівна академія

Основними факторами, що визначають технологічний режим термообробки прокату, згідно з [1] є: тиск газового середовища p; витрати газового середовища на 1 м. п. виробу ε; швидкість охолодження Δ; відносні витрати води в секціях водяного охолодження T.

Основними показниками якості процесу термічної обробки є наступні характеристики металу [1]: границя міцності σв; відносне звуження ψ; відносне подовження δ; розмір зерна d.

Для «зв'язування» вихідних показників якості з факторами, що визначають технологічний режим, будується регресійна модель.

Враховуючи, що показники якості залежать і від вихідного хімічного складу сталі, в число варійованих факторів включено відсотковий вміст основних легуючих компонентів C, Mn.

В загальному вигляді рівняння множинної лінійної регресії, що зв'язує вхідні і вихідні змінні, приймуть вигляд:

$$\sigma_в = f(T, C, Mn, p, \varepsilon, \Delta),\ \delta = f(T, C, Mn, p, \varepsilon, \Delta),$$
$$\psi = f(T, C, Mn, p, \varepsilon, \Delta),\ d = f(T, C, Mn, p, \varepsilon, \Delta). \quad (1)$$

Для побудови моделі множинної регресії доцільно використати логістичну криву, яка має вигляд (2):

$$y_i = \frac{a}{1 + b \cdot e^{f(x_i)}} \tag{2}$$

Логістична крива характеризує зростання із змінним відношенням приросту до ординати. Обрано такий вид функції, бо експоненціальні криві добре описують процеси, які мають «лавиноподібний» характер, тобто коли приріст в основному залежить від досягнутого рівня, при цьому різного роду обмеження, фактори практично не беруться до уваги.

Для чисельної реалізації логістична крива перетворюється до лінійного виду:

$$\ln\left(\frac{a - y_i}{y_i}\right) = \ln b - x_i \tag{3}$$

Заміною змінних і параметрів $\left(y^*_i = \ln\left(\frac{a - y_i}{y_i}\right);\ B = \ln b\right)$ здійснюється перехід до лінійної регресії: $y^*_i = B \cdot (-x_i)$

Для аналізу рівняння множинної лінійної регресії використовується метод найменших квадратів (МНК). Для перевірки якості побудованої моделі використовується середня помилка апроксимації, яка обчислюється як середнє відхилення розрахункових значень від фактичних результатів вимірювань. З метою формування загального уявлення про якість моделі, відносних відхилень по кожному спостереженню обчислюється середня помилка апроксимації (допустима межа помилки не повинна перевищувати 8-10%).

Розглянутий технологічний процес характеризується наявністю сукупності критеріїв якості та обмежень, які в загальному випадку знаходяться в протиріччі один з одним, коли поліпшення одного з них призводить до погіршення іншого і навпаки. Це неминуче вносить елементи якісного, суб'єктивного характеру в постановку задачі оптимізації, розв'язання компромісних питань між критеріями, про їх ранжирування і згортку в узагальнений показник якості технологічного процесу.

Критерії й обмеження є джерелом невизначеності, бо при їх формуванні використовується інформація, заснована на досвіді та інтуїції осіб, відповідальних за ведення процесу. Тому для опису критеріїв якості доцільно застосувати положення теорії нечітких множин [1].

Для формалізації критеріїв і обмежень використовуються функції приналежності трапецієвидного типу. На рис. 1-2 представлені функції належності показників якості, що відображають усереднені вимоги до технологічного процесу.

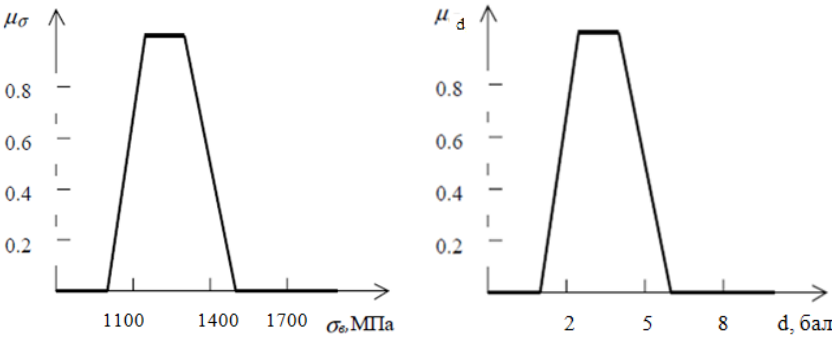

Рисунок 1 – Функції належності критеріїв

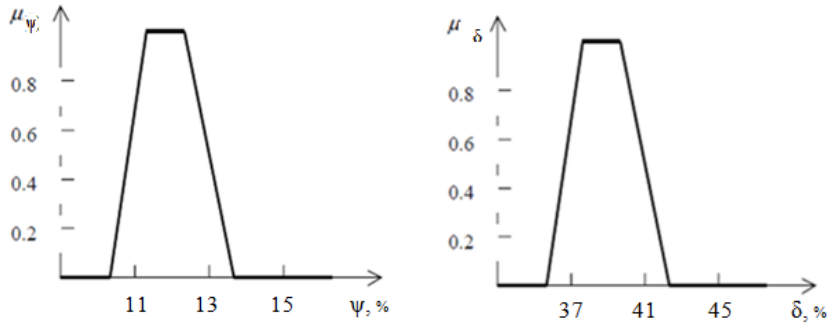

Рисунок 2 – Функції належності критеріїв

Оскільки сформульовані критерії досягаються за різних умов зміни варійованих параметрів технологічного процесу, то оптимальне рішення є компромісом суперечливих вимог. Для цього критерії агрегуються в узагальнений критерій якості процесу з урахуванням відносної важливості задоволення різних вимог.

Узагальнений критерій якості процесу має вигляд:

$$D(C,Mn,T,v,\varepsilon,\Delta) = \min(\mu_{\sigma}^{\alpha_1}(\sigma(C,Mn,T,v,\varepsilon.\Delta)),$$
$$\mu_{\psi}^{\alpha_2}(\psi(C,Mn,T,v,\varepsilon,\Delta),\mu_{d}^{\alpha 3}(d(C,Mn,T,v,\varepsilon,\Delta)), \quad (4)$$
$$\mu_{\delta}^{\alpha_2}(\delta(C,Mn,T,v,\varepsilon,\Delta)))$$

де $\alpha_1 ... \alpha_4$ – коефіцієнти відносної важливості критеріїв.

Даний критерій D приймає значення в діапазоні [0;1]. Чим ближче інтегральний показник до 1, тим більша частина параметрів якості лежать в своїх оптимальних рівнях і вище рівень якості деталі в цілому.

Введені наступні групи якісної оцінки параметра D: «Низький рівень

якості», D ∈ [0;0.6); «Прийнятний рівень якості», D ∈ [0,6;0.8); «Високий рівень якості», D ∈ [0,8;1).

З метою автоматизації наведеного процесу моделювання технологічного процесу термообробки розроблено модуль, що складається з декількох форм, шість з яких призначені для розрахунку технологічних параметрів термообробки металу і витрат ресурсів, необхідних для здійснення даного технологічного процесу (рис. 3).

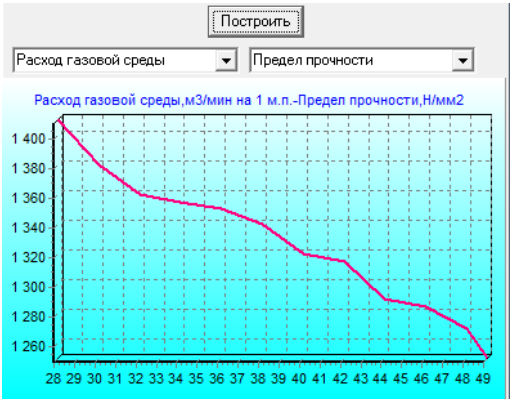

Рисунок 3 – Графік залежності параметра якості «Межа міцності» від технологічного параметра «Витрата газової середовища»

Література

1.Дилигенский Н.В. Нечеткое моделирование и многокритериальная оптимизация производственных систем в условиях неопределенности / Н.В. Дилигенский, Л.Г. Дымова, П.В. Севастьянов – М.: «Издательство Машиностроение − 1», 2004.– 397 с.

Метод реалізації генератора випадкових чисел

Демський О.О.
Науковий керівник: к.т.н. доц. Бойчук В.О.
Хмельницький національний університет

У сучасному світі майже всі володіють смартфоном і для більшості людей ця технологія стала важливою частиною їхнього життя. Смартфони мають велику кількість переваг такі як: робота в мережі мобільного зв’язку, підтримка WI-FI, Bluetooth, завдяки чому користувачі мають доступ до мережі Інтернет для завантаження та запуску сторонніх додатків. Більшість смартфонів мають вбудовані датчики: GPS, гіроскоп, акселерометр, камера, мікрофон та датчик освітленості. Для пристрою з таким числом користувачів

у всьому світі важливо, щоб він був належним чином захищений. Зростаюча популярність смартфонів роблять смартфони постійною ціллю атак хакерів. Багато користувачів смартфонів зберігають особисті дані на своїх телефонах, тому проблеми з безпекою на такій платформі, як Android, мають суттєвий вплив на суспільство та економіку.

Нажаль смартфони на операційній системі Anroid мають проблеми з безпекою даних. Зловмисники можуть перехоплювати або змінювати з'єднання зі смартфоном. Це серйозно впливає на безпеку і конфіденційність користувача смартфона.

Захищеність криптографічних систем залежить від секретних даних, які відомі авторизованим користувачам, але невідомі і непередбачувані для інших. Тобто сучасні криптографічні системи вимагають частої генерації випадкових значень. Для цього використовують генератори випадкових чисел (ГВЧ) на основі джерел ентропії для генерації випадкових бітів даних.

Криптографічні атаки, які порушують роботу або використовують слабкі сторони ГВЧ, називаються атаками генератора випадкових чисел. Створення високоякісного ГВЧ завжди потрібне для забезпечення безпеки, а відсутність якісного ГВЧ призводить до вразливості до атак. Протягом останніх років було багато прикладів атак на ГВЧ криптографічних систем. Наслідки цих атак варіюються між незначним витоком даних і серйозними проблемам для безпеки

Генератори псевдовипадкових чисел (ГПВЧ) використовують односторонні функції для генерації ентропії. Ці алгоритми потребують введення ключа на основі джерела ентропії для створення довгих випадкових послідовностей даних. Є багато прикладів ГПВЧ, такі як Fortuna, алгоритм Yarrow та конструкції, які використовують шифри, такі як AES-CTR, XSalsa20, ChaCha20[2]. Перевага ГПВЧ полягає в тому, що вони можуть генерувати практично нескінченний потік випадкових даних після того, як вони правильно налаштовані і вони не стикаються з проблемою блокування процесу оновлення ентропії. Недоліком є те, що використаний ключ в ГПВЧ є основною вразливістю і всі згенеровані дані стають передбачуваними.

Генератори справжніх випадкових чисел використовують джерело природної ентропії для генерації випадковості (наприклад, фоновий шум, атмосферний шум, шум електромагнітного апарату або космічне випромінювання). Природна ентропія, як правило, дуже непередбачувана, недетермінована і важко вимірювана зовнішнім спостерігачем. Недоліком природної ентропії є те, що ГВЧ необхідно регулярно оновлювати свою ентропію, щоб залишатися непередбачуваним. Під час збирання додаткової ентро-пії ГВЧ блокується, доки не буде достатньо ентропії для задоволення параметрів.

В операційній системі Linux спеціальний символьний псевдопристрій /dev/random є прикладом цього процесу, він блокується, поки не накопичиться достатня ентропія. При читанні даних у пристрої /dev/random

створюються тільки випадкові байти, що складаються з бітів шуму «хаотичного» пулу. У ядрі Linux «хаотичний» пул отримує ентропію з декількох джерел, в тому числі з апаратного генератора випадкових чисел процесорів Intel.

Пристрій /dev/random необхідний користувачам, наприклад, при створенні ключа шифрування, який передбачає тривале використання.

Якщо «хаотичний» пул спорожнів, то читання /dev/random блокується, поки необхідну кількість бітів в пулі не буде створено. Ця блокувальна поведінка може суттєво сповільнити роботу системи, коли виконуються дуже великі запити на випадковість.

Таким чином існує проблема отримання випадкових чисел в реальному масшабі часу на пристроях Android.

В пристроях Android є кілька можливих додаткових типів давачів , які можуть використовуватися для отримання випадкових бітів для накопичення випадкових чисел. Більшість смартфонів мають вбудовані датчики: GPS, гіроскоп, акселерометр, камера, мікрофон та датчик освітленості.

Стаціонарний акселерометр може забезпечити випадковість з високою ентропією. Деякі джерела ентропії, подібні до взаємодії користувача з сенсорним екраном та введення з клавіатури не завжди є надійним джерелом ентропії. Генерація ентропії з мікрофону та датчиків камери, які генеруюють дані з фонового шуму, потребують активної взаємодії з користувачем для забезпечення надійної ентропії. Користувачеві потрібно забезпечувати звуки для мікрофона або різні зображення для камери на, щоб забезпечити більш високий рівень випадковості.

Інші датчики, такі як термометри, датчики близькості та магнітометри, можуть забезпечити джерело випадковості, але меншої кількості ніж акселерометр

При розробці методу генерації випадкових чисел в мобільних пристроях використаємо комбінованний підхід, тобто зчитування наявних показів з декількох функціонуючих давачів пристрою, з подальшим комбінуванням отриманої послідовності з даними вбудованих генераторів випадкових чисел.

Використаємо для генерування ентропії вектори обертання, акселерометр, давач лінійного прискорення, давач гравітації, давач орієнтації, гіроскоп, давач близькості, барометр, магнітометр.

Найпопулярнішим методом виділення випадковості є криптографічна хеш-функція, така як SHA(Secure Hash Algorithm). Наприклад пристрій /dev/urandom використовує дані з ентропійного пулу як ключовий вхід. Використаємо хеш-функцію SHA-256, щоб отримати 256-бітний ключ із даних давачів, який потім буде служити як випадковий ключовий вхід для алгоритму шифрування XSalsa20 для забезпечення постійного потоку випадковості.

Отже спочатку треба автоматично визначити, які датчики доступні для

генерації даних на пристрої. Далі доступні датчики періодично генерують дані і ці дані надсилаються на вхід хеш-функції SHA-256. Коли з даних датчиків було зібрано достатню ентропію, хеш-функція перетворює дані на 256-бітний вихід. Цей 256-бітовий вихід використовується як ключ для алгоритму Xsalsa20 разом з 192-бітовим додатковим значенням nonce, тобто числом, яке може бути використано один раз. Це одноразовий код, обраний випадковим або псевдовипадковим чином. На відміну від випадкових чисел, тут не потрібно непередбачуваності числа, досить неповторюваності.

Збільшений розмір nonce в алгоритмі XSalsa20 дозволяє використовувати для його генерації криптографічно стійких псевдовипадкових чисел. Отриманий потік з потім використовується як безперервний потік випадковості, який може надаватись за запитом. Схема генерації випадкових чисел згідно данного методу показана на рисунку 1.

В отриманій програмі для платформи Android API SensorEvent використовується для читання даних датчиків, які він зберігає їх у масивах значень. Бібліотека Java MessageDigest використовується для реалізації функції SHA-256. Для реалізації алгоритму Xsalsa20 використовується API Spongy Castle, який є варіантом API Java Cryptography для роботи на пристроях Android. Користувач може взаємодіяти з програмою шляхом введення необхідної кількості бітів у випадковому числі.

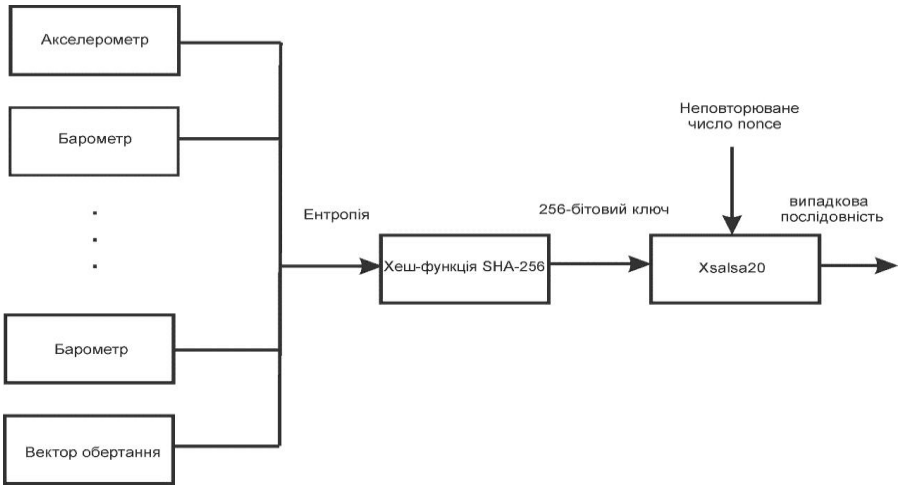

Рисунок 1 – Схема генерації випадкових чисел

Користувач отримає інформацію про час, що залишився, для заповнення пулу ентропії, якщо дані датчика все ще обробляються. Після того, як система готова отримувати запити про випадковість, результуюча випадковість відображається користувачеві у шістнадцятковому форматі.

Мета даного дослідження полягала у тому, щоб забезпечити нове джерело випадковості, яким можна доповнити існуючі джерела. Для цього було запропоновано використовувати дані датчиків Android пристрою для отримання високоентропійних даних. Результати показали, що зразки даних містять велику кількість непередбачуваності і мають рівномірний розподіл значень. 256-бітне випадкове число для отримання випадкового ряду отримується близько раз у 50 мілісекунд.

Поєднання отримання даних з високою ентропією з давачів з існуючим апаратом ентропії на основі апаратного забезпечення допоможе захистити користувачів пристроїв Android від вразливостей щодо безпеки.

Література

1. Boaz Barak and Shai Halevi. A model and architecture for pseudo-random generation with applications to /dev/random. In Proceedings of the 12th ACM conference on Computer and communications security, p. 203-212. ACM, 2005.

2. Daniel J Bernstein. The Salsa20 family of stream ciphers. In New stream cipher designs, pages 84-97. Springer, 2008.

3. Jonathan Voris, Nitesh Saxena, and Tzipora Halevi. Accelerometers and randomness: perfect together. In Proceedings of the fourth ACM conference on Wireless network security, pages 115-126. ACM, 2011.

Аналіз аномалій результатів порівняння DDoS-атак поточного стану системи з її нормальним станом

Долішний В.С.
Науковий керівник – к.т.н.,доц. Чешун В.М.
Хмельницький національний університет

Оптимальним рішенням для виявлення початку атаки і подальшого виявлення шкідливого трафіку буде рішення, засноване на проведення аналізу аномалій, в результаті якого відбувається порівняння поточного стану системи з її нормальним станом. Порівняння станів системи в контексті DDoS-атак можна проводити шляхом порівняння різних властивостей мережевої активності. До цих властивостей можуть бути віднесені: кількість запитів, тип запитів, кількість запитів певного типу або протоколу, IP адреса джерела, швидкість надходження запитів, їх час і т.д.

Нехай множина $A\left(a_1, a_2, a_3, ..., a_n\right)$ - набір всіх можливих властивостей для всіх мережевих клієнтів. Множина $B\left(b_1, b_2, b_3, ..., b_m\right)$ - множина легітивних клієнтів конкретного мережевого ресурсу. Кожен мережевий клієнт має набір індивідуальних властивостей. Наприклад, клієнт b_1 має властивості $A1\left(a_4, a_8, a_{10}, a_{14}\right)$, клієнт b_2 має властивості

$A2\,(a_3, a_8, a_{11}, a_{14})$ і т.д. Дані властивості представляють набір підмножин множини *A*. Перетин всіх цих підмножин характеризує клієнтів мережевого ресурсу, за якими вони можуть бути класифіковані. Точно так нелегітивні клієнти матимуть свій набір властивостей, за якими вони також можуть бути класифіковані.

У минулому був популярний засіб протидії DDoS-атакам типу HTTP-flood, що працює на стороні сервера, що атакується і виявляє шкідливий трафік за допомогою аналізу таких властивостей мережевої активності, як тип даних при завантаженні. Справа в тому, що браузер легітивного клієнта при завантаженні вмісту web-сторінки автоматично завантажує додаткові файли, необхідні для нормальної побудови і відображення сторінки. До цих файлів можуть відноситися: зображення, що знаходяться на сторінці, іконки, каскадні сторінки стилів, фрагменти скриптів JavaScript, винесені в окремі файли. Для зловмисника ці файли є непотрібними, його головне завдання - змусити відпрацювати скрипт, що генерує сторінку, і, тим самим, викликати додаткове навантаження на ресурси сервера. Таким чином, комп'ютери зомбі-мережі могли бути ідентифіковані з високою часткою ймовірності з аналізу тільки однієї цієї властивості. Природно, що зловмисники у відповідь на створення даного методу ускладнили алгоритм атаки, і на сьогоднішній день зомбі-комп'ютери намагаються завантажувати і всі супутні дані. На сьогоднішній день DDoS-атаки ускладнюються, і зловмисники намагаються повністю імітувати поведінку легітивних клієнтів. У цій ситуації перевагу при аналізі властивостей мережевої активності необхідно віддати тим властивостям, які не можуть бути підроблені зловмисниками. При нестачі таких властивостей необхідно вводити штучні властивості, наприклад, проходження модифікації тест Тьюринга - введення даних з картинки.

Таким чином, завдання по визначенню і виявлення шкідливих запитів в контексті даної роботи зводиться до їх класифікації на підставі властивостей мережевої активності. Оптимальним рішенням для виявлення шкідливого трафіку є використання різних класифікаторів і нейронних мереж. Складністю в реалізації даного рішення є той факт, що для нормального функціонування класифікатора потрібно мати дві актуальні навчальні вибірки, відповідно шкідливому і легітивному трафіку. Однак до моменту початку атаки отримати ці вибірки не представляється можливим. Це цілком очевидно для вибірки, що відповідає шкідливому трафіку, так як до початку атаки шкідливі запити відсутні. Але це також справедливо і для вибірки, що характеризує легітивний трафік. Так як мережева картина постійно змінюється, буде змінюватися і вміст вибірки відповідного легітивного трафіка. Таким чином, вибірка по легітивності трафіка, наприклад, місячної давності, може бути не актуальна для поточної мережевої ситуації. Крім того, є ризик, що в цій вибірці можуть виявитися дані, відповідні шкідливим запитам, що в подальшому викличе помилки в

роботі класифікатора. Дана проблема дуже актуальна, тому що зловмисник може спеціально почати підмішувати до легітивного трафіку незначне число шкідливих запитів, які не зможуть бути ідентифіковані як початок атаки, але зможуть негативно «навчити» вибірку, що характеризує легітивний трафік.

Для подолання цієї проблеми необхідно точно визначити точку початку атаки. Це дасть можливість весь попередній трафік віднести до легітивного і відкриє додаткові можливості по розділенню змішаного трафіку, який приходить після початку атаки, на легітивний і шкідливий. В цьому випадку методика виявлення шкідливого трафіку, в першому наближенні, буде зводитися до наступних кроків:

1. Визначаємо актуальні сезонні періоди.
2. З урахуванням сезонності визначаємо точку початку атаки.
3. Відносимо попередній перед початком атаки трафік до легітивного.
4. Класифікуємо змішаний трафік на легітивний і шкідливий.
5. Порівнюємо легітивний трафік виділений зі змішаного з трафіком що надійшов до початку атаки.
6. На підставі результатів, отриманих в попередньому кроці і вироблених критеріїв успішності, коригуємо вибірки.
7. Весь трафік, що надходить аналізуємо з урахуванням отриманих даних.

В рамках розробки методики виявлення DDoSатак і шкідливого трафіку розроблений оригінальний алгоритм виявлення на ранніх стадіях точки початку розподіленої атаки, спрямованої на відмову в обслуговуванні. Алгоритм враховує сезонні відхилення в навантаженні, що дає можливість виявляти точку початку атаки на ранніх стадіях і з більшою точністю. Додатково проведено дослідження, спрямоване на підтвердження існування сезонності і виявлення типових сезонних періодів. В результаті дослідження виявлені тижнева, добова і невизначена сезонність і причини її виникнення.

Розроблено методику отримання навчальних вибірок та класифікації трафіку, що надходить, на групи шкідливих і надійних запитів. Для поділу змішаного трафіку використовується алгоритм кластеризації k-means. Вибір даного алгоритму обгрунтований, проведено доказ його ефективності. Для алгоритму підібрані оптимальні характеристики і розмірність даних, вироблені критерії успішності. Розроблені алгоритми складають основу узагальненої методики виявлення DDoS-атак і шкідливого трафіку, яка в загальному вигляді може бути записана так:

1. За допомогою накопичених статистичних даних, визначаємо існуючі сезонні періоди.
2. Для кожного сезонного періоду визначаємо допустиму верхню межу кількості запитів.
3. У разі порушення границі, фіксуємо точку початку атаки.
4. Відносимо весь, що передує початку атаки, трафік до кластеру, відповідному надійному трафіку.

5. За допомогою алгоритму k-means класифікуємо змішаний трафік на надійний і шкідливий.

6. Порівнюємо трафік, що передує початку атаки, з кластером, надійного трафіку, виділеного зі змішаного трафіку.

7. На підставі результатів, отриманих на попередньому кроці, і з урахуванням вироблених критеріїв успішності, коригуємо кластери.

8. Весь трафік, що надходить, аналізуємо з урахуванням отриманих в попередньому пункті результатів.

Література

1. Алферов, А.П. Основы криптографии /А.П.Алферов, А.Ю. Зубов, А.В. Черемушкин. – М.: Гелиос АРБ, 2002. – 480 с.

2. Анин, Б.А. Защита компьютерной информации / Б.А. Анин – Спб.: БХВ-Петербург. 2000. – 384с.

3. Бабаш, А.В. Информационная безопасность. Лабораторный практикум : учеб. пособие / А. В. Бабаш, Е. К. Баранова, Ю. Н. Мельников. - 2-е изд., стер. - М. : КНОРУС, 2016. - 132 с.

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

Модель формування множини альтернативних структур енергосистеми з альтернативними джерелами енергії

Єлісєєва А.Р. , Бойко О.В.
Науковий керівник - доц. Шендрик В.В.
Сумський державний університет

У роботі розглядається типова енергосистема, у якій джерелами електроенергії є сонячні батареї (СБ) та вітрові турбіни (ВТ). Для підтримки безперервного живлення енергосистема також містить акумуляторні батареї (АБ). У даному дослідженні розглядаються саме ці елементи, так як від них, у першу чергу, залежить рівень електрозабезпечення та ціна системи. Інші елементи енергосистеми при плануванні її структури не розглядаються.

На сучасному ринку існує велика кількість установок сонячної та вітрової генерації електроенергії, що відрізняються потужністю, ціною, габаритами. Це збільшує час на прийняття правильного рішення відносно визначення можливих конфігурацій енергетичної системи з використанням альтернативних джерел енергії (АДЕ).

Важливим питанням є ефективне використання АДЕ, а саме їх комбінація для максимізації отримання генеруючої енергії при мінімізації витрат на їх установку та підтримку. Проблема планування структури таких систем на даний час все більше привертає увагу дослідників.

Мета даної роботи – створення моделі розрахунку робочих

конфігурацій енергосистеми з АДЕ для їх подальшої обробки системою підтримки прийняття рішень.

Для досягнення поставленої мети було вирішено наступні задачі:

– розроблено структуру парсингу даних із зовнішніх ресурсів;

– спроектовано модель бази даних;

– створено математичні моделі для формування загальної кількості конфігурацій енергетичної системи та визначення з них робочих альтернатив;

– розроблено відповідну інформаційну технологію для формування та збереження масиву даних.

Об'єкт дослідження – процес розробки всіх можливих конфігурацій при використанні гібридної вітро-сонячної системи електропостачання окремого об'єкта для забезпечення безперебійного електроживлення споживача та отримання якісної електричної енергії на виході із системи.

Предмет дослідження – модель процесу розробки множини конфігурацій гібридної вітро-сонячної системи електропостачання.

Ключова ідея функціонування гібридної вітро-сонячної системи електропостачання (ГВССЕ) полягає в інтегруванні та координуванні роботи всіх споживачів та джерел генерації в енергосистемі. Припускаємо, що типова ГВССЕ складається з генераторів, з невеликою виробничою потужністю: сонячних батарей (СБ) та вітрових турбін (ВТ). Для того, щоб забезпечити безперервне живлення електроенергією, як обов'язковий елемент будь-якої конфігурації системи є акумуляторні батареї. Це дозволить збалансувати роботу мережі у критичні моменти (наприклад, поломки обладнання, несприятливі погодні умови, тощо), так як акумуляторна батарея буде накопичувати надлишкову енергію, яку у подальшому можуть використовувати користувачі гібридної системи [1,2].

Для досягнення поставленої мети для перебору можливих конфігурацій енергосистеми пропонується використати елементи комбінаторного аналізу, а саме операцію сполучення. Таким чином вирішення системи рівнянь (1), визначить кількість можливих конфігурацій енергосистеми Amount [3]:

$$\begin{cases} Amo \quad unt = (PV_m^n + W_k^l + PV_m^n \cdot W_k^l) \cdot AB_o^p \\ AB_o^p \geq 1 \end{cases} \tag{1}$$

де PV_m^n, W_k^l – кількість можливих комбінацій СБ та ВТ; AB_o^p – кількість можливих варіантів АБ; n, l, p – відповідно конкретний вид СБ, ВТ, АБ; m, k, o – відповідно загальна кількість видів СБ, ВТ, АБ.

Для того щоб розрахувати кількість елементів для кожного виду СБ та ВТ, потрібно спочатку визначити кількість електроенергії, яку виробляє лише один елемент та, як наслідок, розрахувати максимальну кількість елементів у конфігураціях лише з СБ/ВТ, що будуть покривати добове споживання.

Максимальна кількість елементів СБ визначається за формулою 2.

$$\begin{cases} \min\limits_{q=1:q_{\max}} \cdot(q) \cdot P_{PV_m^n} > P \\ q \in [1 : q_{max}] \end{cases} \quad (2)$$

де $\min\limits_{q=1:q_{\max}} \cdot(q)$ – мінімальна кількість сонячних батарей в сонячній системі електропостачання, згенерована потужність яких покриває добове споживання; P_{pv} – добова потужність СБ, Р – добова енергія, спожита у господарстві.

Максимальна кількість елементів ВТ визначається за формулою 3.

$$\begin{cases} \min\limits_{q=1:q_{\max}} \cdot(w) \cdot P_{W_k^l} > P \\ qw \in [1 : w_{max}] \end{cases} \quad (3)$$

де $\min\limits_{q=1:q_{\max}} \cdot(w)$ – мінімальна кількість елементів у вітровій системі електропостачання, згенерована потужність яких покриває добове споживання; P_w – добова потужність ВТ.

Кількість робочих конфігурацій в залежності від кількості елементів СБ та ВТ наявних у енергосистемі визначається за формулою 4.

$$\begin{cases} Amount_{work} = \begin{bmatrix} (\min\limits_{q=1:q_{\max}} (q) \cdot P_{PV_m^n}) + (\min\limits_{w=1:w_{\max}} (w) \cdot P_{W_k^l}) \\ + (u_q^a \cdot P_{PV_m^n} \cdot v_w^b \cdot P_{w_k^l}) \\ a \in [1; q],\ b \in [1; w] \end{bmatrix} \cdot AB_o^p > P \end{cases} \quad (4)$$

де u_q^a та v_w^b – кількість можливих конфігурацій елементів СБ та ВТ в одній конфігурації; *a, b* – відповідно чітко визначене число елементів СБ та ВТ; *q, w* – відповідно загальна кількість елементів СБ, ВТ, які можуть використовуватися в конфігураціях.

Таким чином у роботі було проведено комплекс досліджень, який дозволив створити модель визначення можливих конфігурацій системи електропостачання з використанням альтернативних джерел енергії, вироблена потужність яких покриває добову кількість споживання електроенергії людиною в господарстві.

Література

1. N.Alon, J. H. Spencer, The Probabilistic Method (Wiley: 2016).
2. T. Ritchey. General Morphological Analysis (GMA). Wicked Problems – Social Messes. Risk, Governance and Society (Berlin: Springer: 2011).
3. O.Shulyma, V.Shendryk, et al., Communications in Computer and Information Science 756, 213-225 (2017).

Аналіз проблем людського фактору в задачах забезпечення кібербезпеки

Кіншаков Е., Щербань Т.
Науковий керівник – професор Лавров Е.А.
Сумський державний університет, Суми, Україна

Проблеми кібербезпеки, набули надзвичайної актуальності.Інформаційна безпека (ІБ) складається з цілого комплексу різних заходів і дій. Це, перш за все, контроль дій різного роду суб'єктів - рядових співробітників компанії, привілейованих користувачів, IT-аутсорсерів, контрагентів. Крім того, це чітке розмежування прав доступу всередині компанії, використання резервного копіювання даних, а також наявність простої, зрозумілої і доведеної до відома працівників політики безпеки. У поточних реаліях захист повинен бути досить гнучким, щоб забезпечити і достатній рівень захищеності, і виконання бізнес-цілей. Згідно з інформацією, яка міститься в дослідженні Lloyd's of London і Cyence, фінансові втрати від масштабної кібератаки можуть коштувати світовій економіці від 15,6 млрд до 121 млрд доларів. Якщо розглядати найбільш песимістичний сценарій розвитку подій, то втрати від кібератак можуть перевищити економічний збиток від урагану «Катріна», який став найбільш руйнівним в історії Сполучених Штатів. Втрати від нього склали 108 млрд доларів. У доповіді вказуються два потенційних сценарію розвитку глобальної кібератаки: злом провайдерів хмарних сховищ або використання можливих вразливостей в операційних системах.

У першому сценарії хакери модифікують «гіпервизор», керуючу систему хмарних сховищ, в результаті чого всі зберігаються файли виявляються загубленими. У другому варіанті розглядається гіпотетичний випадок, коли кібераналітік випадково забуває в поїзді сумку, в якій зберігається доповідь про уразливість всіх версій операційної системи, встановленої на 45% всіх світових пристроїв. Ця доповідь згодом продається кримінальним групам. Мінімальний збиток при першому сценарії складе від 4,6 млрд до 53,1 млрд доларів. При другому сценарії втрати складуть від 9,7 млрд до 28,7 млрд доларів.

Людський фактор. Саме проблема «надійних рук» або, кажучи іншими словами, кваліфікованих кадрів є однією з найбільш нагальних. Вона має особливу актуальність протягом усіх останніх років, тому що на сьогоднішній день людина залишається найбільш уразливим ланкою в IT-інфраструктурі.Найслабша ланка в інформаційній безпеці банку - це співробітник компанії. Якщо співробітники не дотримуються правил безпеки, то технології не зможуть допомогти захиститися.

Так, при використанні соціальної інженерії зловмисники можуть змусити співробітника організації здійснити якусь дію, яке спростить проведення атаки, пояснює експерт. «Часто, щоб підібрати пароль до аккаунту, зловмиснику не обов'язково його« зламувати »- вся інформація про

пароль є в профілі соціальних мереж або поруч з робочим столом. Навіть співробітники на керівних позиціях виробляють маніпуляції, спровоковані зловмисниками. Окремим рядком можна привести небажання працівників слідувати політиці і вимогам по ІБ заради спрощення своєї роботи». Щоб мінімізувати вплив людського фактора, потрібно постійно підвищувати обізнаність співробітників в області ІБ, а також впроваджувати систему контролів і моніторингу дотримання політик і вимог в області ІБ. Серед основних способів мінімізації загрози ІБ -підвищення обізнаності персоналу в питаннях ІБ, проведення тестів, ділових ігор, кібернавчань.

У зв'язку з проблемою ризиків, які несе людський фактор, цікаво згадати дослідження антивірусної компанії ESET, опубліковане в липні 2017 року. Чотири компанії з п'яти недооцінюють ризики ІБ, пов'язані з людським фактором. Такий висновок зробили співробітники ESET після опитування інтернет-користувачів з СНД. Респондентам запропонували вибрати відповідь на питання «Чи проходили ви на роботі тренінг з інформаційної безпеки?». Негативна відповідь лідирує з великим відривом. 69% респондентів ніколи не проходили навчання основам кібербезпеки в своїх компаніях. Ще 15% учасників опитування повідомили, що їх роботодавці обмежилися мінімальним обсягом інформації. Навчання не виходило за рамки «в разі неполадок перезавантажте комп'ютер», правила кібербезпеки не зачіпалися. Тільки 16% респондентів пройшли якісні тренінги з докладною розповіддю про інформаційну безпеку. Для порівняння: більше 60% учасників аналогічного опитування в США повідомили, що їх роботодавці організували для них навчання з кібербезпеки.

Аналіз основних визначень і підходів до організації обробки персональних даних

Ковальчук Я.В.
Науковий керівник – к.т.н., доц. Джулій В.М.
Хмельницький національний університет

Інформація, яка містить відомості про фізичних осіб (громадян) - персональні дані, використовуються в різних системах обробки інформації все частіше, що обумовлено постійним розширенням сфери застосування інформаційних технологій для обслуговування населення. Специфіка роботи з персональними даними заснована на потенційній можливості їх використання для заподіяння шкоди суб'єктам, до яких відносяться дані - власникам персональних даних. Особлива увага приділяється питанням захисту персональних даних (ПД) в автоматизованих інформаційних системах ПД – (ІСПД). Вимоги до захисту в ІСПД, відповідно до низки документів, враховують категорію і кількість ПД, специфіку вирішуваних завдань і ряд інших показників. Виконання цих вимог, як правило, пов'язане з

істотними матеріальними і фінансовими витратами.

У зв'язку з цим представляють інтерес дослідження спрямовані на розробку і аналіз методів обробки ПД, що дозволяють знизити витрати на забезпечення безпеки в ІСПД.

Для реалізації цього підходу потрібно розробити методи знеособлення і де-знеособлення ПД, що будуть включати в себе правила роботи з знеособленими даними, теоретичне обґрунтування методів знеособлення і де-знеособлення ПД, що дозволяють забезпечити їх конфіденційність, а також правил організації обробки знеособлених даних.

База персональних даних – іменована сукупність упорядкованих персональних даних в електронній формі та/або у формі картотек персональних даних. З огляду на це база персональних даних є упорядкованою сукупністю логічно пов'язаних даних про фізичних осіб: що зберігаються та обробляються відповідним програмним забезпеченням, є базою персональних даних в електронній формі; що зберігаються та обробляються на паперових носіях інформації, є базою персональних даних у формі картотек.

Картотекою персональних даних є будь-який структурований масив персональних даних, що є доступним з визначеними критеріями, незалежно від того, чи є такий масив централізованим, децентралізованим або розділеним на функціональних або географічних засадах. Такі дані мають бути структуровані за визначеними критеріями, що стосуються осіб, щоб забезпечити легкий доступ до відповідних персональних даних. Варто зазначити, персональні дані одночасно можуть бути упорядковані і в електронній формі, і в формі картотек. Фізичні особи-підприємці та самозайняті особи самостійно визначають чи володіють вони базами персональних даних у сенсі Закону.

Законодавець поширив дію Закону на всі види діяльності, пов'язані зі створенням баз персональних даних та обробкою персональних даних у цих базах, за винятком такої діяльності, яка здійснюється: фізичною особою - виключно для непрофесійних особистих чи побутових потреб; журналістом - у зв'язку з виконанням ним службових чи професійних обов'язків; професійним творчим працівником - для здійснення творчої діяльності.

Так, під час здійснення своєї професійної діяльності на адвокатів законодавством не покладено обов'язок ведення баз персональних даних клієнтів. Але, якщо адвокати формують справи на своїх клієнтів, які вони постійно оновлюють та підтримують в актуальному стані, такі справи є базою персональних даних та підлягають державній реєстрації. Нотаріуси можуть обробляти персональні дані своїх найманих працівників, клієнтів у базах персональних даних, однак, документи нотаріального діловодства та архів нотаріуса, визначені у статті 14 Закону України «Про нотаріат» не є базою персональних даних у сенсі Закону України «Про захист персональних даних» та не підлягають державній реєстрації. Крім того, у

випадку, якщо фізичні особи – підприємці, укладаються договори виконання робіт або надання послуг з фізичними особами, такі договори також не є базою персональних даних та не підлягають державній реєстрації.

Власником бази персональних даних згідно з абзацом третім статті 2 Закону є фізична або юридична особа, якій законом або за згодою суб'єкта персональних даних надано право на обробку цих даних, яка затверджує мету обробки персональних даних та процедуру їх обробки, якщо інше не визначене законом. Так, якщо персональні дані обробляються юридичною особою, то власником бази персональних даних є юридична особа. Розпорядником бази персональних даних згідно з абзацом дев'ятим статті 2 Закону може бути фізична чи юридична особа, якій власником бази персональних даних або законом надано право обробляти ці дані.

Практичним прикладом можуть бути відносини між юридичними особами та їх представництвами, філіями, відділеннями тощо. Так, у сенсі Закону ці представництва, філії, відділення виступатимуть розпорядниками баз персональних даних, власником яких є юридична особа. Згідно зі статтею 4 Закону власником чи розпорядником бази персональних даних можуть бути підприємства, установи і організації усіх форм власності, органи державної влади чи органи місцевого самоврядування, які обробляють персональні дані відповідно до закону. Але якщо власником бази персональних даних є орган державної влади чи орган місцевого самоврядування, то розпорядником бази персональних даних, крім цих органів, може бути лише підприємство державної або комунальної форми власності, що належить до сфери управління цього органу.

Порядок обробки персональних даних. Обробка персональних даних включає в себе такі дії, як збирання, реєстрацію, накопичення, зберігання, адаптування, зміну, поновлення, використання і поширення (розповсюдження, реалізацію, передачу), знеособлення, знищення персональних даних. Обробка персональних даних може бути здійснена як неавтоматичними засобами з носіїв (у тому числі паперових), що становлять будь-який структурований масив персональних даних, який є доступним за визначеними критеріями, так і з використанням інформаційних (автоматизованих) систем.

До персональних даних можна віднести будь-які відомості, за якими ідентифікується або може бути ідентифікована фізична особа, зокрема: прізвище, ім'я, по батькові, адреса, телефони, паспортні дані, національність, освіта, сімейний стан, релігійні та світоглядні переконання, стан здоров'я, матеріальний стан, дата і місце народження, місце проживання та перебування тощо, дані про особисті майнові та немайнові відносини цієї особи з іншими особами, зокрема членами сім'ї, а також відомості про події та явища, що відбувалися або відбуваються у побутовому, інтимному, товариському, професійному, діловому та інших сферах життя особи (за винятком даних стосовно виконання повноважень особою, яка займає посаду,

пов’язану із здійсненням функцій держави або органу місцевого самоврядування) тощо. Вказаний перелік не є вичерпним. Така інформація про фізичну особу та членів її сім’ї є конфіденційною і може оброблятися в тому числі поширюватись тільки за їх згодою, крім випадків, визначених законом, і лише в інтересах національної безпеки, економічного добробуту та прав людини.

Вимоги до обробки персональних даних. Власник визначає: мету та підстави обробки персональних даних; категорії суб’єктів персональних даних; склад персональних даних; порядок обробки персональних даних (спосіб збору, накопичення персональних даних; строк та умови зберігання персональних даних; умови та процедуру зміни, видалення або знищення персональних даних; умови та процедуру передачі персональних даних та перелік третіх осіб, яким можуть передаватися персональні дані; порядок доступу до персональних даних осіб, які здійснюють обробку, а також суб’єктів персональних даних; заходи забезпечення захисту персональних даних; процедуру збереження інформації про операції, пов’язані з обробкою персональних даних та доступом до них). Також, у деяких випадках, власник визначає обов’язки та права осіб, відповідальних за організацію роботи, пов’язаної із захистом персональних даних під час їх обробки. Обробка даних повинна здійснюватися тільки за згодою суб’єкта. Якщо за результатами розгляду такої вимоги виявлено, що персональні дані суб’єкта (їх частина) обробляються незаконно власник повинен припинити обробку персональних даних суб’єкта (їх частини) та проінформувати про це.

Видалення та знищення персональних даних здійснюється у спосіб, що виключає подальшу можливість поновлення таких персональних даних. Також власник , розпорядник персональних даних повинен вжити заходів щодо забезпечення захисту персональних даних на всіх етапах їх обробки, у тому числі за допомогою організаційних та технічних заходів.

Розглянемо автоматизовані системи обробки персональних даних.

Автоматизована система обробки ПД – це організаційно-технічна система, що являє собою сукупність наступних взаємопов'язаних компонентів: технічних засобів обробки і передачі даних (засобів обчислювальної техніки і зв'язку); методів і алгоритмів обробки у вигляді відповідного програмного забезпечення; інформації (масивів, наборів, баз даних) на різних носіях; персоналу і користувачів системи, об'єднаного за організаційно-структурними, тематичними, технологічними або іншими ознаками для виконання автоматизованої обробки інформації (даних) з метою задоволення інформаційних потреб суб'єктів інформаційних відносин.

Аналіз методів організації обробки і захисту ПД, показав, що пропоновані методи і створювані на їх основі системи захисту вимагають значних ресурсів для реалізації, мають сильну залежністю від типу даних і високою надмірністю при практичному застосуванні для роботи з масивами даних невеликої розмірності. Тому в ряді випадків доцільно застосовувати

методи, що знімають вимоги до конфіденційності ПД, що значно скорочує витрати на захист. Одним з ефективних і перспективних підходів до захисту ПД в інформаційних системах є знеособлення.

У зв'язку з цим доцільно провести дослідження в наступних напрямках: аналіз основних визначень і підходів до організації обробки персональних даних, створення математичної моделі персональних даних; розробка досить універсальних і мало витратних методів і засобів перетворення персональних даних, що забезпечують зниження вимог до захисту; серед таких методів можна виділити знеособлення персональних даних; розробка методики застосування знеособлення при організації захисту персональних даних; розробка методів оцінки якості захисту персональних даних при знеособлення; розробка інформаційної системи для реєстрації операторів персональних даних та контролю за їх діяльністю.

Література

1. Шаньгин В.Ф. Комплексная защита информации в корпоративных компьютерных системах: Учебн. пособие. В.Ф. Шаньгин - М.: ИД «ФОРУМ»: ИНФРА-М, 2010. - 592с.

2. Щербаков А.Ю.. Современная компьютерная безопасность. Практические аспекты. /А.Ю. Щербаков. - М.: Книжный мир, 2009. - 352с.

3. Сабанов А.Г. Защита персональных данных в организациях здравоохранения. // А.Г. Сабанов, В.Д. Зыков. - Москва, Горячая линия Телеком, 2012. - 206с.

Діагностування схем оперативної пам’яті з довільним доступом

Кравчук Р.В., Гаврилюк Р.Л., Ференс В.О.
Науковий керівник – к.т.н.,доц. Чешун В.М.
Хмельницький національний університет

Жорстка конкуренція на ринку електронних компонентів, пристроїв та систем стала рушійною силою стрімкого прогресу цифрових технологій, майже кожного дня з’являються новини про появу нових технологій, компонентів або виробів цифрової електроніки. При цьому терміни актуальності наявних розробок швидко зменшуються через появу нових поколінь і моральне старіння існуючих. З точки зору технічної діагностики дискретних пристроїв і систем це призводить появу нових класів і видів об’єктів діагностування, що приходять на зміну раніше існувавшим поколінням прототипів, і зумовлює потребу постійного вдосконалення існуючих методів діагностування та створення нових.

Дослідження тенденцій розвитку схем оперативної пам’яті показує, що зміна і вдосконалення технологій їх виробництва дозволяють постійно збільшувати ступінь інтеграції таких компонентів та нарощувати обсяги

пам’яті на кристалі. На сьогоднішній день на єдиному кристалі розміщуються елементи пам’яті на гігабайти двійкового коду, що зовсім нещодавно здавалося недосяжним.

Отримані позитивні результати у створенні надвеликих схем оперативної пам’яті з довільним доступом мають негативні наслідки для технічної діагностики:

– велика кількість комірок пам’яті, що розміщуються на одному кристалі і підлягають діагностуванню, зумовлюють потребу у реалізації надзвичайно великої кількості елементарних тестових перевірок для покриття всього поля діагностованих елементів;

– збільшення внутрішньої складності (нарощування ємності/розрядності) схем оперативної пам’яті з довільним доступом висуває підвищенні вимоги до застосовуваних засобів діагностування як за обсягами використовуваних і оброблюваних діагностичних даних, так і за можливостями контролю і керування діагностованим об’єктом (зокрема, щодо управління адресним простором) ;

– збільшення ступеня інтеграції мікросхем пам’яті, в першому варіанті, досягається за рахунок зменшення відстаней між елементами пам’яті на кристалі (зростання щільності розміщення запам’ятовуючих елементів), що підвищує ймовірність утворення паразитних з’єднань між елементами пам’яті, через які можуть спотворюватись біти даних взаємно пов’язаних елементів;

– збільшення ступеня інтеграції мікросхем пам’яті, в другому варіанті, досягається за рахунок зменшення самих елементів пам’яті на кристалі, що зменшує їх надійність і збільшує імовірність їх нестабільності та руйнування з різних причин;

– використання надвисоких робочих частот (в тому числі з застосуванням методів помноженої синхронізації) в сукупності зі зменшенням відстаней між елементами пам’яті на кристалі через збільшення ступеня інтеграції мікросхем пам’яті підвищує ймовірність утворення паразитних електромагнітних впливів між зазначеними елементами, що також може призводити до втрати цілісності збережуваних даних.

Проблеми нестабільності, взаємовпливів та саморуйнування елементів схем оперативної пам’яті з довільним доступом загострюються по мірі їх експлуатаційного старіння.

До позитивних властивостей схем оперативної пам’яті як об’єктів діагностування слід віднести однорідність і регулярність їх внутрішньої структурної організації.

Стосовно схем оперативної пам’яті з довільним доступом в їх структурі можна виділити схему вибірки-керування і поле комірок зберігання даних. З формальної точки зору, поле комірок зберігання даних мікросхеми оперативної пам’яті з довільним доступом – це масив адресованих елементів пам’яті, що може бути відображений матричною структурою розмірності $m \times n$.

Однорідність і регулярність їх внутрішньої структурної організації схем оперативної пам’яті зумовлює застосування типових варіантів тестів для зазначеного класу об’єктів діагностування. Тести для пам'яті виділяються в окремий клас, проте простота тестування пам'яті є умовною, що зумовило появу великої кількості варіантів тестів з різними підходами до організації діагностичних випробувань.

Тест "всі нулі" (аналогічно і "всі одиниці") передбачає запис до всіх комірок дігностованої схеми оперативної пам’яті нулів (одиниць), після чого виробляється послідовне зчитування і перевірка цілісності збережуваних даних. Тести "всі нулі" і "всі одиниці" дозволяють виявити константні несправності елементів пам’яті (незмінність значень одиниці або нуля).

Тест "адресний" передбачає запис до всіх комірок кодів їх власних адрес, після чого виробляється послідовне зчитування і перевірка цілісності збережуваних даних. Адресний тест забезпечує найкраще перевірку адресних дешифраторів схем оперативної пам’яті.

Тест "шаховий" передбачає запис до всіх комірок кодів, що є чередуванням нулів і одиниць, при чому в сусідні комірки записуються взаємно інвертовані значення цих кодів. Таким чином, до поля матриці запам’ятовуючих елементів записується набір даних, що має шаховий розподіл нулів і одиниць. Тобто елемент пам'яті, встановлений в одиницю, в полі справної матриці завжди оточений бітами, встановленими в нуль, і навпаки. При несправності елемента пам'яті або наявності паразитних зв’язків створюється велика ймовірність "стікання" заряду (або 1 в 0, або навпаки). Потім проводиться послідовне зчитування і перевірка цілісності записаних даних. Шаховий тест, найчастіше, використовується для перевірки взаємовпливу сусідніх комірок, що містять інформацію, записану в зворотному коді.

Тест "сканування" передбачає запис до всіх комірок кодів одного значення, а потім їх перезапис на протилежне значення. Тобто, спочатку у всі комірки записуються нулі (одиниці), які потім послідовно зчитуються і реалізується перевірка записаних даних, а після цього в усі комірки схеми пам’яті записуються одиниці (нулі), які потім також послідовно зчитуються і реалізується перевірка якість зберігання записаних даних. Тест "сканування" використовується для перевірки схем оперативної пам’яті через створення умов максимальної статичної перешкоди, викликаної сумарним струмом витоку (заряду) всіх комірок , які перебувають в одному стані і змінюють його на протязі мінімально можливого часу.

Тест "чергування рядків 0/1" передбачає запис до всіх комірок кодів з однотипних значень в межах комірки, але при цьому в сусідні комірки записуються взаємно інвертовані значення цих кодів (наприклад, в комірки з непарними номерами всі нулі, а в в комірки з парними номерами всі одиниці). Таким чином, на матричному полі утворюються горизонтальні рядки нулів і одиниць, що чередуються. Потім проводиться послідовне

зчитування і перевірка цілісності записаних кодів. Цей тест використовується для перевірки взаємовпливу адресних шин за рядками.

Тест "чергування стовпців 0/1" передбачає запис до всіх комірок однакових значень кодів, що утворюються чередуванням нулів і одиниць. Таким чином, на матричному полі утворюються вертикальні стовпчики (колонки) нулів і одиниць, що чередуються. Потім проводиться послідовне зчитування і перевірка цілісності записаних кодів. Цей тест використовується для перевірки взаємовпливу адресних шин за стовпцями.

Тест " запис, запис/зчитування вперед і назад" передбачає запис до всіх комірок значень нулів, після чого проводиться послідовне зчитування і перевірка записаних значень. Одразу по завершенню перевірки кожної чергової комірки в неї записується значення в оберненому коді (одиниці). Після перевірки останньої комірки і запису в неї одиниць процедура повторюється від старшого значення адреси до молодшої з читанням одиниць, перевіркою записаних значень і заміною на нулі одразу по завершенню перевірки кожної чергової комірки. Цей тест використовується для перевірки взаємовпливу сусідніх комірок при зміні в них даних.

Тест "маршовий" передбачає запис до всіх комірок значень одиниць, після чого проводиться послідовне зчитування записаних даних з перевіркою і заміною на нулі. Після звернення до останньої комірки процедура повторюється з даними в оберненому коді, тобто реалізується послідовне зчитування нулів, починаючи з першої комірки, з перевіркою і заміною на одиниці. Після досягнення останньої комірки процедура знову повторюється з даними в зворотному коді, тобто, з нулями, але вже в зворотному напрямку – від останньої комірки до першої. Після повернення до першої комірки процедура повторюється – зчитуються нулі і на їх місце записуються одиниці. Після досягнення останньої комірки виконується читання з всіх комірок від першої до останньої перевіркою записаних даних (одиниць). Цей тест є модифікацією попереднього тесту "запис і запис / зчитування вперед і назад".

Тест "додаткова адресація" передбачає запис до всіх комірок фонового набору одиниць (нулів), після чого виконується послідовне зчитування інформації, починаючи з першої комірки, з перевіркою і заміною її на нулі (одиниці). Кожне друге звернення виконується за адресою, код якої є доповненням до попередньої. Тест призначений для перевірки адресних ланцюгів, інформація яких в цьому тесті піддається максимальній зміні.

Тест "довбання" передбачає запис до всіх комірок певних тестових кодів (різноманітність варіантів тестових кодів дає модифікації цього тесту), після чого проводиться багаторазове зчитування за кожною адресою з подальшою перевіркою збереження цілісності зчитуваних значень. Процедура повторюється з замною інформації в кожній комірці на інформацію в оберненому коді. Цей тест призначений для перевірки здатності комірок витримувати багаторазові звернення зі зчитуваннями.

Тест "хрест" передбачає запис до комірок схеми оперативної пам'яті тестових і фонових кодів. При цьому у обрану комірку записується тестовий код (набір одиниць і нулів), а в кожну з чотирьох сусідніх комірок - фонове слово. Потім інформація в сусідніх комірках змінюється і перевіряється вплив цієї зміни на обрану контрольовану комірку. За допомогою цього тесту перевіряється чутливість комірки до змін станів хрестоподібно розташованих сусідніх комірок.

Тест "руйнування зчитуванням" передбачає запис до першої (нульової) комірки тестового слова (всі одиниці), яке записується, одразу зчитується і перевіряється. Виконується приріст адреси та тестове слово записується в наступну комірку. Після цього інформація з обох комірок зчитується і перевіряється. Процедура повторюється до тих пір, поки в усі комірки не буде записано тестове слово. Зчитування всіх попередніх комірок повторюється після кожного нового запису. Таким чином, до нульової комірки проводиться звернень n, до першої – (n-1), …, до останньої - одне. Цей тест призначений для перевірки взаємовпливу комірок пам'яті в ході запису в них однакових значень.

Тест "біг" (або "переміщення") передбачає запис до першої (нульової) комірки тестового слова з одиниць (або нулів), а до всіх інших - фонових нулів (одиниць). Потім всі значення послідовно зчитуються з перевіркою; останньою зчитується перша комірку з подальшим записом в неї інвертованого значення (нулів замість одиниць або навпаки). Послідовність операцій повторюється для другої комірки, третьої і так далі, аж до останньої. Цей тест призначений для виявлення збоїв в схемах оперативної пам'яті, викликаних перехідними процесами в розрядних ланцюгах (переміщення одиниць на фоні нулів, як і протилежне переміщення, створює найгірші умови для підсилювачів зчитування).

Тест "пінг-понг" передбачає запис до першої (нульової) комірки тестового слова з одиниць, а до всіх інших - фонових нулів. Потім послідовно зчитуються і перевіряються комірки 2, 1; потім 3,1; 4,1 і так далі, поки всі пари переходів, що включають комірку 1, не будуть перевірені. Після цього в комірку 1 записуються нулі, а в другу - одиниці. У тій же послідовності операції повторюються для комірки 2 і так далі. Цикл повторюється для інверсної інформації. За допомогою цього тесту перевіряється функціонування накопичувальної частини схем оперативної пам'яті, дешифратора, а також вплив операцій запису на зберігання даних.

Тест "галоп" також передбачає запис до першої (нульової) комірки тестового слова з одиниць, а до всіх інших - фонових нулів. Потім послідовно зчитуються і перевіряються комірки 2, 1, 2; потім 3, 1, 3 і так далі, поки всі пари переходів, що включають комірку 1, не будуть перевірені потрійними зчитуваннями з перевірками. Після цього до першої (нульової) комірки записуються нулі і знову виконується перевірка потрійними зчитуваннями з перевірками. У тій же послідовності операції повторюються для наступних

комірок, аж до останньої. За ефективністю тест "галоп" еквівалентний тесту "пінг-понг".

Як показує аналіз наведених методів, діагностування схеми вибірки-керування оперативної пам'яті з довільним доступом полягає в перевірці правильності реакції схеми на сигнали керування, забезпеченні можливості доступу до всіх комірок пам'яті в режимах запису-зчитування, а також виконання додаткових функцій (переведення виходів у стан високого імпедансу тощо). Для перевірки схеми вибірки-керування достатньо виконати запис масиву даних з мінімальними дублюваннями в комірки пам'яті з подальшим його зчитуванням і контролем (адресний тест), а також провести активізацію і контроль додаткових функцій. Тестування поля комірок передбачає перевірку їх статичних і динамічних властивостей. Статичні властивості характеризують загальну здатність комірок пам'яті зберігати будь-які набори даних без їх пошкодження (через несправності розрядів пам'яті, утворення паразитних зв'язків між ними тощо). Для перевірки статичних властивостей використовуються тести «всі нулі», «всі одиниці», «сканування», «шаховий», «маршовий» та інші. Динамічні властивості характеризують здатність елементів пам'яті зберігати дані в часі, їх перевіряють, зокрема, тестом «довбання» (запис тестового масиву в комірки пам'яті з багатократним періодичним повторним зчитуванням і контролем).

Література

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

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

3. Дрозд А.В. Вероятностный подход к функциональному диагностированию вычислительных устройств для обработки приближенных данных / А.В. Дрозд // Радіоелектроніка і інформатика. – Харків. : ХНУРЕ, 2004. – № 1. – С. 101-102. Ярмолик В. Н. Обзор методов неразрушающего тестирования ОЗУ // В. Н. Ярмолик , А. П. Занкович / Доклады БГУИР, 2005. – No 4 (12) – С.62-72.

4. Ярмолик В. Н. Тестовое диагностирование аппаратного и программного обеспечения вычислительных систем // В. Н. Ярмолик , А.А. Иванюк / Доклады БГУИР, 2014. – No 2 (80) – С.127-142.

5. Малиновский М. Л. HDL-модель памяти RAM со встроенной схемой генерации неразрушающих тестов //М. Л.Малиновский, Д. А.Аленин, Барсов В. И./ Вісник Харківського національного технічного університету імені Петра Василенка, 2014 – Випуск №117. – С.45-51.

6. Кордовер К. А . Универсальный блок управления массивом запоминающих устройств наземного отладочного комплекса // К. А . Кордовер , А. А. Жданов, А. М.Данилов / Труды МАИ. – Выпуск № 65, 2016. – С.61-70

7. Ярмолик В. Н. Псевдоисчерпывающее тестирование запоминающих устройств на базе многократных маршевых тестов // В. Н. Ярмолик , И . Мрозек, В. А. Леванцевич / Информатика, 2018. – Т. 15, No 1. – С.110-121

8. Ярмолик В.Н. Симметричное неразрушающее тестирование ОЗУ //А.П. Занкович,В.Н. Ярмолик / Доклады белорусского государственного университета информатики и радиоэлектроники. – 2005. – С.57-62.

9. Michael L. Essentials of electronic testing for digi- tal, memory and mixed-signal VLSI circuits / L. Michael, D. Vishwani. – Kluwer: Academic Publishers, 2002. – 671 c.

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

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

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

Адаптація процесів захисту доступу користувачів до соціальних систем

Купратий В.О.
Науковий керівник: к.т.н. доц. Красильников С.Р.
Хмельницький національний університет

Захист інформаційних соціальних систем є обов'язковою складовою та передумовою їх функціонування. Оскільки соціальна система орієнтована на обслуговування користувачів, то захист доступу до системи є особливо актуальним. Складність задачі захисту соціальних систем зумовлена цілим рядом факторів, до яких належать: відсутність підготовки користувачів до використання інформаційних систем; необхідність у відповідних рівнях захисту для різних даних, що перебувають у системі; введення нових даних, для яких необхідно визначити потрібний рівень захисту та інші фактори, що відображають багатогранність використання інформаційних систем у суспільстві.

У сучасних умовах розвитку соціальних систем інформатизація процесу їх функціонування постійно збільшується. Це зумовлено такими

причинами: інформатизація суспільства допомагає окремим його членам суттєво скоротити час розв'язання задач, пов'язаних із використанням державних установ, відповідно зменшується необхідна кількість взаємних контактів громадян з працівниками державних органів і т. д. Оскільки органи державного управління повинні гарантувати захист даних про громадян від несанкціонованого їх використання особами, які можуть мати ті чи інші негативні наміри, то в міру поширення інформаційних соціальних систем має підвищуватися рівень їх захищеності. Зростання міри захищеності не може являти собою процес, який стосується всіх даних, тим більше всіх соціальних систем однаковою мірою. Рівень захищеності соціальних систем і даних, що в них знаходяться, повинен бути керованим залежно від типу даних та рівня повноважень і користувачів.

Захист даних у соціальних інформаційних системах через їх розподіленість та наявність великої кількості користувачів забезпечується через захист доступу до системи. Для випадку соціальних систем захист доступу не може реалізовуватися тільки на рівні використання стандартних засобів, що обмежуються застосуванням ідентифікаторів та паролів.

Задачі захисту даних та управління і контролю доступу за своєю суттю доволі складні, оскільки будь-яка небезпека для реалізації ініційованої нею атаки насамперед уможливлює доступ до системи.

Незважаючи на значну кількість праць, присвячених захисту інформаційних систем, в рамках зазначеної проблематики не розв'язані задачі адаптації захисту системи з врахуванням категорій таємності даних та рівня повноважень користувачів.

Адаптація системи захисту доступу (СЗД) до соціальної системи (СС) є необхідною характеристикою системи типу (СС) і має цілий ряд особливостей, які відрізняють її від інформаційних систем інших типів, що орієнтовані на розв'язання інших задач.

Особливості системи СС:

– параметри міри захищеності даних системи є розподіленими в межах системи і можуть набувати в середовищі СС різних значень;

– значення параметра захищеності даних може змінюватися динамічно, і такі зміни зумовлюються, на рівні з іншими факторами, також зовнішніми факторами;

– система доступу до СС як і засоби захисту доступу до СС є неоднорідною та розподіленою в просторі;

– система засобів захисту являє собою окрему розподілену мережу, в якій, крім зв'язків з системою СС, існують зв'язки між окремими засобами захисту;

– мережа засобів системи доступу, крім контролю доступу до СС, розв'язує задачі контролю видачі даних та контролю послуг, що надаються в результаті звернення користувачів до системи.

З наведеного вище можна прийняти, що мережа засобів захисту є деяким бар'єром, який відділяє СС від зовнішнього оточення. Оскільки зовнішнє середовище являє собою певним чином організованих користувачів системи, то необхідність у реалізації адаптивних можливостей такої системи зумовлюється такими факторами:

– користувачі системи СС є досить неоднорідні з погляду своєї підготовленості до використання системи;

– неоднорідність користувачів стосується також і їх намірів або цілей, з якими вони звертаються до системи;

– у процесі експлуатації системи окремі елементи системи доступу можуть виходити з ладу або з інших причин відмовляти користувачеві у доступі, але, незважаючи на це, доступ до системи має бути забезпечений;

– зміни, що відбуваються в середовищі СС, особливо ті, що стосуються рівнів захисту даних, можуть впливати на алгоритми, які реалізуються в середовищі мережі системи доступу.

Оскільки до складу системи СС входить загальна система управління, необхідність в якій зумовлюється такими факторами:

– технічним обслуговуваннями системи;

– необхідністю управління зв'язками між системами, які можуть бути системами різних типів;

– необхідністю управління мережею доступу.

Методи реалізації адаптації СЗД тісно пов'язані з причинами, які її зумовлюють.

Підготовка типового споживача до використання тієї чи іншої інформаційної системи визначається мірою масовості поширення і, відповідно, використанням інформаційної системи СС та типами інших електронних інформаційно- комунікаційних засобів, що набули широкого розповсюдження у суспільстві. Одним із таких засобів є мобільний засіб зв'язку. Більшість мобільних телефонів, що масово використовуються, мають широкі функціональні можливості.Тому використання мобільних телефонів для зв'язку з системою СС є доцільним, оскільки в цьому випадку розв'язується задача реалізації фізичного доступу користувачів до системи доступу СС. Канали мобільного зв'язку є достатньо захищеними та відповідають прийнятим стандартам міри захищеності приватних даних. У такому випадку регіональний пункт системи доступу до СС представляється у вигляді деякого абонента мобільної мережі зв'язку. Очевидно, що використання засобів безпеки каналів мобільного зв'язку в їх стандартному вигляді для мереж мобільного зв'язку не забезпечує необхідного захисту інформаційної системи. Особливість використання мобільного зв'язку для доступу до СС полягає у тому, що активізація такого зв'язку - це комунікація зі сторони користувача. Вона реалізується голосовими повідомленнями і у крайньому випадку - текстовими повідомленнями, що надходять зі сторони

користувача у вигляді SMS - повідомлень. Комунікація зі сторони СД до СС також реалізується голосовим способом. Як і у випадку з користувачем голосовий спосіб може замінятися на комунікацію з допомогою SMS повідомлень.

Користувач, крім мобільних каналів зв'язку з СС повинен мати можливість комунікуватися із системою СС із стаціонарних пунктів доступу, розміщених у відповідному регіоні. Передавання даних з СС до користувача може здійснюватися такими способами:

– дані можуть передаватися на адресу електронної пошти користувача;

– на записуючі пристрої регіонального пункту доступу до системи, яка відповідає регіону перебування користувача в момент запиту;

– на друкарські пристрої регіональних пунктів доступу до СС.

Отже, пр. адаптації процесів захисту доступу користувачів до соціальних систем використовуються профілі користувачів, на основі аналізу яких є можливим розрізнити непідготовленого користувача від несанкціонованого користувача. Використання уявлень про різні категорії даних дає можливість адаптувати механізми доступу до даних окремих користувачів.

Література

1. Дурняк Б. В. Засоби захисту даних в інформаційних системах / Б. В. Дурняк, Т. М. Хомета // Квалілогія книги. - 2015. - № 1 (27). - С 32-40.

2. Хомета Т. М. Організація процесу доступу користувачів до соціальної інформаційної системи / Т. М. Хомета // XXTTT Міжнародна науково- практична конференція. Український науково-дослідний інститут спеціальних видів друку. - К., 2016. - С. 145-147.

3. Хомета Т. М. Адаптація засобів захисту до заданого рівня захисту / Т. М. Хомета // XXII Міжнародна науково-практична конференція. Український НДІ спеціальних видів друку. - К., 2016. - С. 106-108.

Система визначення об'єктів з використанням методу дерева квадрантів

Курай В.І.

Науковий керівник – к.т.н., лоц., Киричек Г.Г.

Запорізький національний технічний університет

Задачі аналізу зображень, розпізнавання образів та порівняння зображень поширені в наш час. Існує багато інструментів, які дозволяють вести дослідження у цьому напрямку. Найбільш поширеними є: перцептивні хеш-алгоритми [1] і штучна нейронна мережа [2]. Найвідомішою бібліотекою для роботи з зображеннями наразі є OpenCV, яка надає засоби для обробки і аналізу вмісту зображень, у тому числі розпізнавання об'єктів [3].

Хоча наведені рішення це добре відточені алгоритми і бібліотеки для роботи з зображеннями, все ж таки інколи розгортання подібної системи є невиправданим з точки зору швидкодії, задіяння ресурсів обладнання та складності роботи з ним. Тому в даній роботі досліджено та запропоновано свій підхід до вирішення цієї проблеми, шляхом впровадження системи визначення об'єктів з використанням методу дерева квадрантів, для порівняння зображень, як можливе рішення, коли компактне представлення зображення і високоефективний доступ до елементів є ключовими вимогами.

На вибір Python, у якості мови програмування вплинули: підтримка Python об'єктно-орієнтованого підходу, простота синтаксису, а також наявність великої кількості вбудованих функцій і структур даних, які прискорили швидкість розробки програмного забезпечення [4].

До популярних методів сегментації зображення можна віднести метод дерева квадрантів (Q-дерево). Основна ідея квадро-дерева - комбінування однакових або схожих елементів даних і кодування великих однорідних сукупностей даних малою кількістю бітів. Причини вибору цього методу наступні: компактне представлення зображення та високоефективний доступ до елементів, які досягаються за рахунок деревовидної структури даних [5].

Виходячи з основних завдань роботи наведено метод реалізації дерева квадрантів при розбитті зображення. Перший крок розбиття зображення - отримання зображення у вигляді двомірного масиву пікселів, шляхом створення методу get_pixels_from_image(), що повертає двомірний масив, елементами якого є масив довжиною 3. Використовуючи доступ до елементів через індекс, отримано значення кольору, у відповідній точці зображення.

Змінюючи мінімальний розмір вузла регулюємо якість вихідного зображення. Завдяки мінімальному розміру вузла обмежуємо подальше створення нащадків. Так як, великі області одного кольору представляться у вигляді одного великого прямокутника, то на їх подальше розбиття зміна мінімального розміру вузла не впливає. Натомість, області зображення, на яких знаходиться декілька кольорів, породжують більшу кількість вузлів. У цих випадках за допомогою зменшення мінімального розміру вузла можна завадити подальшому розбиттю зображення.

Для визначення середнього значення кольору створено метод, який приймає на вхід вузол дерева. Далі у вкладеному циклі, через індекси проводиться доступ до пікселів зображення, що знаходяться в області вузла дерева. Проводиться підрахунок трьох складових кольору та кожна складова кольору ділиться на кількість пікселів у заданому вузлі, для знаходження середнього значення. Таким чином метод повертає три складових кольору. В тілі методу calc_color_distance_in_rect(self,rect), який приймає в якості аргументу прямокутник (вузол) проводиться підрахунок різниці кольорів.

Наступним кроком є знаходження різниці за модулем між середнім значенням кольору в квадранті та значенням кольору пікселя. Для цього у вкладеному циклі знаходим різницю між складовою кольору кожного пікселя

та середнім значенням кольору. Розрахунок проводиться три рази, для кожної складової кольору. Результати підсумовуються.

Після завершення підрахунків метод повертає значення різниці кольору поділене на площу прямокутника та помножену на три. Множення площі прямокутника на три, відбувається через те, що для тієї ж самої площі розрахунок проводився три рази, для кожної складової кольору. Код, який відповідає за отримання пікселів із зображення, знаходження середнього кольору та дистанції кольорів в вузлів дерева винесено у клас ImageSplitter.

Результатом розбиття зображення виступає масив квадрантів вузлів дерева. Кожний такий вузол містить в собі об'єкт типу прямокутника та значення середнього кольору та відстані кольорів. Маючи такий набір характеристик для кожного вузла дерева, можна провести порівняння двох зображень через порівняння їх дерев квадрантів.

Для коректного порівняння елементів дерева квадрантів треба для класу QuadTreeRect змінити метод, що відповідає за порівняння об'єктів. Відповідно до нової реалізації метод __eq__() поверне значення True, у тому разі якщо співпаде середній колір квадрантів, а також координати їхньої верхньої лівої та нижньої правої точок. Наступним кроком є знаходження таких елементів дерева квадрантів зображення 1, яких не містить зображення 2. Тобто необхідно знайти різницю множин дерев квадрантів. Для реалізації даної операції розроблено метод compare_images_rects(self, list1, list2).

Алгоритм порівняння зображень такий: провести розбиття зображень, задавши значення мінімального розміру вузла; зробити порівняння отриманих вузлів дерева двох зображень; для знайдених квадрантів, повторювати порівняння, поки досягнемо заданого значення мінімального розміру квадранта; після отримання результатів, створити новий графічний файл, та на основі отриманого дерева квадрантів отримати зображення.

В результаті проведення досліджень та визначення основних етапів роботи системи, при порівнянні зображень, розроблено консольну утиліту, яка використовує клас ArgumentParser. Користувач (або система), при цьому, має можливість передати в програму наступні аргументи: шлях до зображень; мінімальний розмір вузла дерева; мінімальну різницю кольорів та шлях до міста збереження результуючого зображення. У лістингу 1 наведено код, що відповідає за передачу аргументів з консолі та запуск роботи програми.

Лістинг 1 - Передача аргументів з консолі та запуск роботи програми

```
def get_args():
parser = ArgumentParser()
parser.add_argument('--image1',required=True,type=str)
parser.add_argument('--image2',required=True, type=str)
parser.add_argument('--save_to',required=True, type=str)
parser.add_argument('--min_color_diff', required=False,type=int, default=5)
parser.add_argument('--min_quad_size',required=False,type=int, default=4)
```

```
args = parser.parse_args()
return args
if __name__ == '__main__':
args = get_args()
image_comparator = ImageComparator(
    args.image1,
    args.image2,
    args.save_to,
    args.min_color_diff,
    args.min_quad_size)
image_comparator.compare_images()
```

В роботі розглянуто приклад роботи програми при порівняні зображень супутникових знімків аеродрому з використанням послідовності кроків розбивки зображень, поки не буде досягнута задана точність, та порівнянь їх дерев квадрантів. Також наведена залежність часу виконання роботи програми, в залежності від площі мінімального вузла. В результаті порівняння зображень отримано третє результуюче з вмістом отриманих результатів. Цей метод дозволяє порівнюючи карти місцевості, отримані в результаті періодичної зйомки, виявляти об'єкти що з'являються або зникають у місцях проведення контролю місцевості, без участі людини.

Література

1. «Выглядит похоже». Как работает перцептивный хеш. – [Електронний ресурс]. – Режим доступу: https://habrahabr.ru/post/120562/.

2. Лукін В.Є. Аналіз використання технології штучних нейронних мереж в якості нового підходу до обробки сигналів // В.С. Лукін / Телекомунікаційні та інформаційні технології. – №3. – Київ, 2014. – С. 81-88.

3. Петин В. Микрокомпьютеры Raspberry Pi: Практическое руководство // В. Петин. – СПб: Питер, 2015. – 240 с.

4. Прохоренок Н.А. Python 3 и PyQt 5. Разработка приложений // Н.А. Прохоренок, В.А. Дронов. – СПб: БХВ-Петербург, 2016. – 812 с.

5. Квадродеревья и октодеревья – [Електронний ресурс] Режим доступу: http://loi.sscc.ru/gis/QuadTree/QuadTree.html.

Дослідження та класифікація основних типів загрозливих програм

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

Однією з ключових сучасних проблем забезпечення комп'ютерної безпеки є необхідність ефективної протидії загрозливим програмам. При цьому необхідно враховувати, що це можуть бути, як самостійні програми,

покликані здійснювати відповідні несанкціоновані дії, так і цілком легальні, санкціоновано використовувані додатки, що наділяються в процесі роботи загрозливими властивостями. У загальному випадку атаки подібних програм можуть бути націлені, як на розкрадання даних, так і на виведення з ладу комп'ютерних ресурсів, як наслідок, об'єктами захисту, стосовно до даних загроз, повинні бути, як інформаційні, так і системні комп'ютерні ресурси. Існуюча статистика зростання загрозливих програм дозволяє припустити про низьку ефективність методів вирішення найбільш актуальних сучасних завдань захисту інформації. Незалежно від типу, загрозливі програми здатні завдавати значної шкоди, реалізуючи будь-які загрози інформації - порушення цілісності, конфіденційності, доступності.

Загрозливі програми прийнято ділити на класи за такими основними ознаками: місце існування; обсяг завданої шкоди; особливості алгоритму роботи; операційна система.

Загрозливі програми (ЗП) по природному середовищі можна розділити на наступні типи: макро; завантажувальні; мережеві; файлові; скриптові. За обсягом заподіяної шкоди ЗП діляться на: безпечні - в результаті свого поширення обмежуються зменшенням вільної пам'яті на диску; небезпечні - можуть привести до серйозних збоїв в роботі комп'ютера або ОС; дуже небезпечні - можуть привести до втрати програм, конфіденційних даних, системних файлів і інших критичних файлів. При цьому не можна з повною впевненістю назвати програму незагрозливою, якщо в її коді, не знайдено команд, що завдають шкоди системі, так як її проникнення в комп'ютер-жертву може викликати непередбачувані наслідки. За алгоритмом роботи ЗП діляться на наступні типи: з використанням стелс-алгоритмів; з самошифруванням і поліморфічністю; з використанням нестандартних прийомів; резидентні; наділення санкціонованих програм шкідливими властивостями.

Використання стелс-алгоритмів дозволяє ЗП повністю або частково приховати себе в системі. Найбільш поширеним стелс-алгоритмом є перехоплення запитів операційної системи на читання/запис заражених об'єктів. Стелс ЗП при цьому заміняють собою незаражені ділянки інформації. Резидентна ЗП при інфікуванні комп'ютера-жертви залишає в оперативній пам'яті свою резидентну частину, яка після зараження перехоплює звернення операційної системи до об'єктів зараження і впроваджується в них. Резидентні ЗП знаходяться в пам'яті і є активними до перезавантаження операційної системи або вимкнення комп'ютера.

За способом зараження ЗП діляться на кілька груп: перезаписуючі; паразитичні; «компаньйон»; інші способи зараження. Перший спосіб зараження є найбільш простим: ЗП записує свій код замість коду в файли, які заражаються, знищуючи його вміст. Як правило перезаписуючі (overwriting) - програми швидше за все виявляються, так як рано чи пізно система починає працювати не коректно або повністю втрачає працездатність. Паразитичні програми (parasitic) додають свій код в заражений файл, після чого він

залишається повністю або частково працездатним. До категорії «компаньйони» відносяться програми, що не змінюють файли, які заражаються. Алгоритм роботи полягає в тому, що для файла, що заражається створюється файл-двійник і при запуску зараженого файла управління отримує саме цей двійник. До інших способів зараження відносяться загрозливі програми, які не пов'язують свою присутність з яким-небудь виконуваним файлом. При зараженні вони копіюють свій код або файл цілком в будь-які каталоги дисків, нові копії будуть коли-небудь запущені користувачем або прописуються в автозапуск.

На основі проведених досліджень всіх типів загрозливих програм пропонується провести їх класифікацію за способами виконання загрозливих файлів. У загальному вигляді загрозливі програми слід ділити на виконувані і макро-програми, в свою чергу виконувані діляться на бінарні, мережні загрозливі програми, класичні комп'ютерні віруси, троянські програми, комп'ютерні черв'яки, хакерські утиліти, потенційно небажане програмне забезпечення, і скриптові загрозливі програми.

Запропоновано загальний підхід до захисту від загрозливих програм, заснований на контролі доступу до ресурсів по розширенням і типам файлів. Дослідження актуальності захисту від загрозливих програм і ефективності існуючих методів захисту, показало, що навіть при такому підході до оцінювання можна зробити висновок, що завдання захисту від загрозливих програм актуальне, а ефективність існуючих засобів захисту низька.

В результаті проведених досліджень виникає необхідність кількісної оцінки актуальності завдання захисту від загрозливих програм і ефективності існуючих засобів захисту. Без вирішення цього завдання неможливо оцінити реальний стан справ в даній області.

Перш за все слід розглянути наскільки завдання захисту від загрозливих програм актуальне для інформаційної системи в цілому, з урахуванням безлічі інших загроз, так чи інакше експлуатованих атаками. Для оцінки актуальності загрози в інформаційній системі, в тому числі розглядаються загрози впровадження та запуску загрозливої програми, використаємо математичну модель, засновану на представленні атаки у вигляді реалізації послідовності загроз на орієнтованому графі (рис. 1). В результаті успішної атаки буде розкрадання (крадіжка) або модифікація інформації, відмова в доступі до інформаційної системи або відмови операційної системи. На рис. 1: *З* - зловмисник; *Зі*, де *і* від 1 до *n* - реалізація загрози; *Ц* - цілі: розкрадання (крадіжка інформації), модифікація інформації, відмова в доступі.

Отримуємо наступну розрахункову формулу побудованої моделі, що дозволяє оцінити ймовірність здійснення атаки:

$$p_{0здійсн} = \prod_{i=1}^{n} (1 - p_{0i}) \quad (1)$$

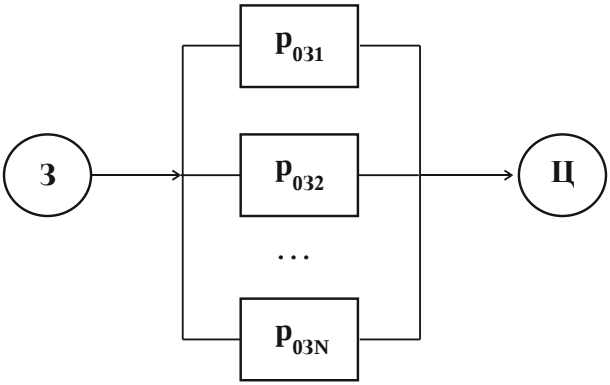

Рисунок 1 - Схема паралельного резервування

Проведено дослідження основних типів загрозливих програм, на підставі якого запропоновано класифікацію загрозливих програм за способом їх виконання. На підставі існуючої статистики зроблено висновок, що найбільш актуальними для захисту є виконувані бінарні і скриптові файли. Проведено дослідження способів впровадження загрозливих програм, в результаті якого дійшли висновку - класи загрозливих програм, що розглядаються передбачають обов'язкове збереження файлу на жорсткому диску перед виконанням (читанням). Для захисту від найбільш актуальних загрозливих програм потенційно може бути реалізований контроль доступу (розмежувальна політика доступу) до файлових об'єктів.

Література

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

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

3. Михайлов А. В. Компьютерные вирусы и борьба с ними. / А.В. Михайлов. – М.: Диалог-МИФИ, 2012. – 148 с.

4. Касперский Е. В. «Компьютерное зловредство» / Е. В. Касперский. – Санкт-петербург: Питер, 2009. – 208 с.

Адаптивний метод передачі інформації по каналах зв'язку з врахуванням завадостійкого кодування

Литвиненко Р.С.
Науковий керівник: к.т.н. доц. Красильников С.Р.
Хмельницький національний університет

Незважаючи на те, що в теорії інформації були побудовані певні рішення для цього завдання, їх не можна визнати задовільними з практичної точки зору. Причиною цього є оптимізаційний критерій, використовуваний в подібних теоретичних дослідженнях, а саме максимізація сумарної пропускної спроможності всіх користувачів системи. Це не дозволяє врахувати обмежень, пов'язаних як з неможливістю досягнення пропускної здатності каналу системи зв'язку за допомогою існуючих методів передачі інформації, так і з необхідністю підтримки певної якості обслуговування окремих користувачів системи [1]. Тому виникає необхідність розробки алгоритмів адаптивної передачі, враховуючи вищенаведені обмеження. При цьому використання високочастотного методу передачі, отримало широке поширення в останні роки, воно дозволяє суттєво спростити реалізацію відповідних оптимізаційних алгоритмів, і також допускає використання при аналізі системи досить простих математичних моделей.

Побудова адаптивної системи передачі даних потребує наявності методу кодування і модуляції, що забезпечує ступінь захисту передачі та прийому даних від перешкод. При цьому особливу важливість має ефективна реалізація методу обробки інформації, зокрема кодування і декодування коригувальних кодів. Алгоритми кодування та декодування багатьох сучасних кодів включають в себе класичні обчислювальні методи, такі як циклічна згортка, пошук кореня многочлена, дискретне перетворення Фур'є тощо.

Незважаючи на те, що відомі швидкі алгоритми вирішення вище зазначених завдань, у багатьох випадках їх використання при реалізації алгоритмів кодування і декодування виявляється вкрай неефективним як в силу специфіки обчислень в кінцевих полях, так і в силу обмежень, накладених структурою алгоритмів кодування і декодування. У зв'язку з цим виникає завдання ефективної реалізації відповідного обчислювального алгоритму.

Зробивши аналіз характеристик алгоритмів модуляції а також деяких систем кодування даних, сигналів та символів, можна впевнено заявляти що самі по собі ці системи модуляції ефективні. Але їхня ефективність обмежена певними чинниками, своєрідністю та способами обчислення. Однак якщо ці системи модуляції/демодуляції інформації об'єднати для утворення алгоритму модуляції, вийде об'єднаний метод, кращий, він буде включати в себе сильні сторони наведених вище алгоритмів і виключати їхні вади [2]. Вже на базі цих алгоритмів можна проводити обрахунки та будувати наш

метод оптимізації. А для цього нам потрібно:

- проаналізувати характеристики ліній зв'язку;
- детально розглянути поширені моделі каналів;
- зробити аналіз позитивних та негативних сторін моделей каналів;
- після отриманих даних зробити висновки про їхню ефективність;
- провести обчислення за найвідомішим методом коригувальних кодів за для отримання математичної моделі;
- на основі отриманих математичних моделей каналів та оптимізованого методу, змоделювати систему для отримання даних що до успішності запропонованого методу;
- на основі зібраних даних провести корегування параметрів мережі.

Розглянемо задачу обчислення коефіцієнтів поділу під каналів в багато користувацькій системі, описану в контексті OFDM-системи [3]. Відомо, що в багато частотній системі прийнятий сигнал для кожного з *K* користувачів і *N* під каналів може бути представлений як (1):

$$r_{ki}^{j} = \mu_{ki}^{j} s_{i}^{j} + \eta_{ki}^{j}, i = 0..N-1, k = 0..K-1 \quad (1)$$

де μ_{ki}^{j} - передавальний коефіцієнт *i*-го під каналу, спостережуваний *k*-м користувачем в момент часу *j*, s_i^j - сигнал, переданий по *i*-му підканалу в *j*-й момент часу, η_{ki}^{j} - адитивний Гаусівський шум.

Припустимо, що повністю відсутня між користувацька інтерференція, що може бути досягнуто при ідеальній синхронізації базової станції і користувацьких пристроїв. У разі кодового розділення для цього додатково потрібно незмінність каналу в часі і ортогональність використовуваних розширюючих послідовностей.

Як було показано вище, для ефективного здійснення передачі даних в системах з одним і багатьма користувачами необхідна адаптація використовуваної схеми передачі до мінливих умов каналу. У даній роботі розглядається задача управління параметрами багато частотного каналу зв'язку з метою мінімізації потужності передавача, необхідної для досягнення заданої швидкості передачі. Практична реалізація цього правила призводить до необхідності використання дискретних швидкостей передачі. Таким чином, можна очікувати підвищення ефективності адаптивної системи у разі використання набору схем кодування/модуляції з малим кроком швидкостей, що може бути реалізовано на основі концепції багаторівневого кодування.

Використання коригувальних кодів у свою чергу, вимагає ефективної реалізації відповідних обчислювальних алгоритмів. При цьому особливий інтерес представляє зниження обчислювальної складності без зниження якості декодування. У тому випадку, коли обчислення проводяться над кінцевими полями, як при декодуванні кодів Ріда-Соломона, застосування деяких їх спеціальних властивостей може призвести.

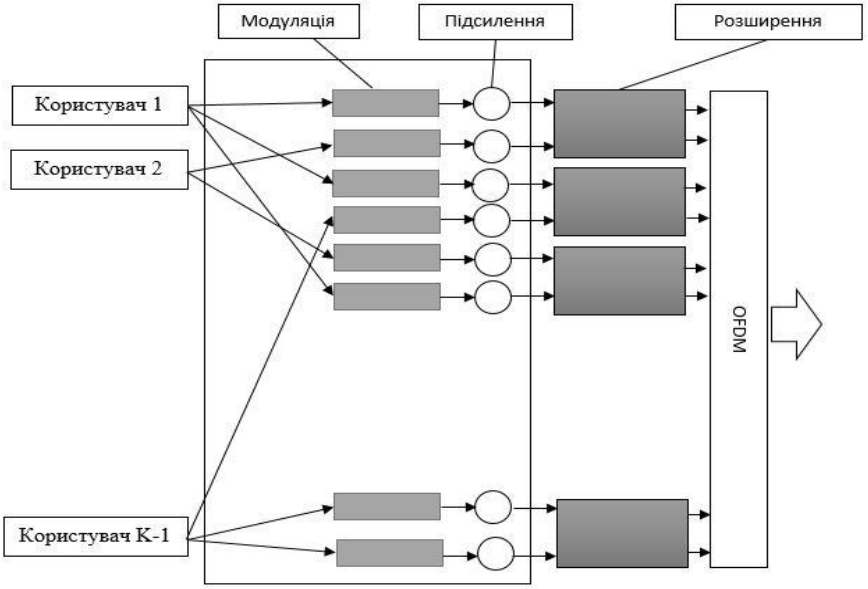

Рисунок 1 - Архітектура системи для оптимізації

Точний теоретичний аналіз адаптивної багатокористувацької системи виявляється досить складним завданням, у порівнянні з аналогічною системою в одно користувацькому випадку. З огляду на те, що рішення оптимізаційної задачі залежить не тільки від передавальних коефіцієнтів, а й від їх співвідношення для різних користувачів, тут вже не можна безпосередньо скористатися апаратом порядкових статистик. Крім того, передавальні коефіцієнти каналів для різних користувачів незалежні. Тоді безліч їх найкращих каналів також будуть незалежними. Отже, ймовірність збігу каналів, використовуваних різними користувачами, досить мала.

Література

1. Vasic B. Combinatorial constructions of low-density parity-check codes for iterative decoding / B. Vasic, Milenkovic O. // IEEE Transactions on Information Theory. – 2004.–June.–Vol. 50, no. 6.

2. Васильев Ф. П. Методы оптимизации/ Ф. П. Васильев.— М.: Факториал, 2012.– 824 с.

3. Mestdagh D. J. G. A method to reduce the probability of clippingin DMT-based transceivers / D. J. G. Mestdagh, P. M. P. Spruyt // IEEE Transactions on Signal Processing.– 1996. –October.– Vol. 44, no. 10.– Pp. 1234–1238.

Гра «Шерлок Холмс» - розв’язання загадки Ейнштейна

Лукін О.Ю.
Науковий керівник – к.т.н., доц. Костюкова Н.С.
ДВНЗ «Донецький національний технічний університет», м.Покровськ

Майже всі, хто мають справу з комп'ютером, стикалися з комп'ютерними іграми. Історія комп'ютерних ігор налічує вже більше 50 років, і за цей час з'явилися культові, інноваційні і дуже красиві ігри [1].

Багато людей намагаються поєднувати їх з чимось більш корисним, наприклад, грати в ігри які розвівають пам'ять, логічне мислення, інтелект. Однією з таких ігор є "Sherlock the logic game", перший поширений екземпляр якої був створений американським інді–розробником Евереттом Кейзером у 1991 році. Ця гра грунтуєтьсяна класичній задачі на логіку, авторство якої приписують інколи Альберту Ейнштейну, інколи - Льюїсу Керролу [2].

Автором створено гру, що базується на загадці Ейнштейна, засобами ігрового движка Unity [3], з використанням алгоритму розв’язання логічної загадки Ейнштейна. Задачу проектування ігрового додатку сформульовано на основі аналізу найпопулярніших з існуючих аналогів. Проектування ігрового додатку здійснено за допомогою мови UML [4].

На рисунку 1 можна побачити текст загадки.

Оригінальний текст загадки Ейнштейна (переклад з англ.

1. На вулиці стоять п'ять будинків.
2. Англієць живе в червоному будинку.
3. У іспанця є собака.
4. У зеленому будинку п'ють каву.
5. Українець п'є чай.
6. Зелений будинок стоїть відразу праворуч від білого дому.
7. Той, хто курить Old Gold, розводить равликів.
8. У жовтому будинку курять Kool.
9. У центральному будинку п'ють молоко.
10. Норвежець живе в першому будинку.
11. Сусід того, хто курить Chesterfield, тримає лисицю.
12. У будинку по сусідству з тим, в якому тримають коня, курять Kool.
13. Той, хто курить Lucky Strike, п'є апельсиновий сік.
14. Японець курить Parliament.
15. Норвежець живе поруч з синім будинком.

Питання: Хто п'є воду, а хто тримає зебру?

Рисунок 1 – Текст загадки Ейнштейна

Рішення цієї загадки полягає у тому, щоб вписати відомі співвідношення в таблицю, послідовно виключаючи неможливі варіанти [5].

На рисунку 2 наведено алгоритм вирішення загадки Ейнштейна.

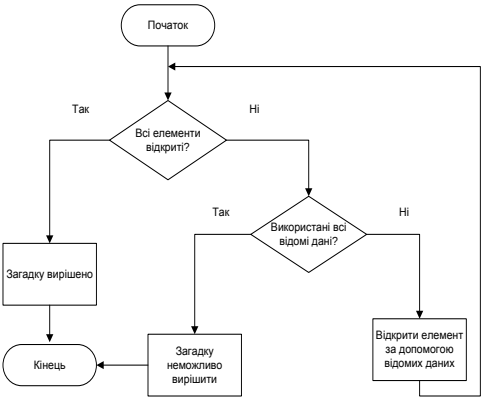

Рисунок 2 – Алгоритм вирішення загадки Ейнштейна

Діаграма класів програмної системи містить 8 класів. MonoBehaviour – це базовий клас для роботи з Unity, від якого успадковуються скрипти. Клас MenuScript відповідає за роботу головного меню, клас PauseScript відповідає за роботу паузи у грі, клас TimerScript відповідає за роботу таймера, клас OptionScript відповідає за роботу меню опцій, клас SetupPuzzle відповідає за роботу гри. Клас Puzzle та Clues є допоміжними класами для класу SetupPuzzle. Гра поділяється на декілька сцен, у кожній сцені є головний клас. У сцени з меню – це MenuScript, у сцени з грою – це SetupPuzzle.

На рисунку 3 наведено діаграму варіантів використання для сцени “Меню”. Гравець, обираючи кожен з пунктів, може потрапляти в нову сцену або переходити по пунктам меню.

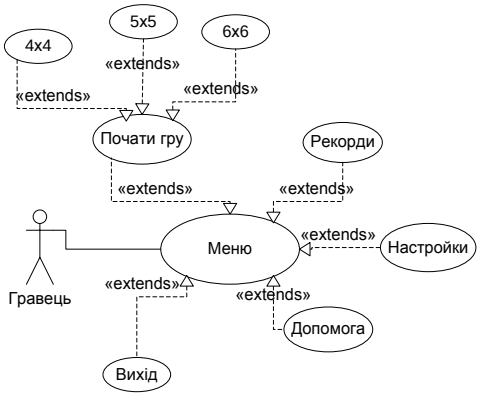

Рисунок 3 – Діаграма варіантів використання для сцени "Меню"

З "Меню", натиснувши на "Почати гру" гравець переходить в підміню,

яке представляє собою вибір складності гри. Після вибору складності гравця перенаправляє на сцену гри, яке представляє собою вхідну таблицю з підказками. Як тільки буде заповнена таблиця – гра завершиться, і з'явиться повідомлення про перемогу, якщо буде встановлено рекорд гравцем, рекорд буде внесено в статистику.

Інтерфейс програми розроблено з дотриманням принципум KISS (keep it short and simple) та принципу п'яти (не більше п'яти кнопок в кожному меню). На рисунку 4 можна побачити приклад меню.

Рисунок 4 – Приклад головного меню

Структура ігрового додатку складається з наступних модулів: модуль меню та модуль для роботи гри.

Модуль меню виконує всю роботу по функціонуванню меню та підменю. Модуль меню включає відображення меню та підменю; обробку натискань елементів меню; обробку зміни налаштувань в підменю; обробку скролінга в підменю; обробку паузи в грі. Модуль для роботи гри виконує головну функцію додатку, він забезпечує роботу гри. Модуль для роботи гри включає відображення ігрового поля; відображення інформаційної панелі; обробку натискання на елементи ігрового поля; роботу таймера гри; розрахунок балів гри; формування підказок.

Література

1. Паркин С. Самые знаменитые компьютерные игры / С. Паркин ; [пер. з англ. М. А. Райтмана]. - Москва : Эксмо, 2015. - 255 с.

2. Sherlock the logic game [Електронний ресурс]. – Режим доступу: http://www.kaser.com

3. Хокинг Д. Unity в действии. Мультиплатформенная разработка на C#/Д. Хокинг – Санкт-Петербург: Питер, 2016. – 336 с.

4. The Unified Modeling Language [Електронний ресурс]. – Режим доступу: http://www.uml-diagrams.org/

5. Stangroom J. Einstein's Riddle: Riddles, Paradoxes, and Conundrums to Stretch Your Mind / J. Stangroom - USA: Bloomsbury, 2009.-144 с.

Метод псевдо-ймовірного блочного шифрування

Мурава В.М.
Науковий керівник - д.т.н., професор Мясіщев О. А.
Хмельницький національний університет

У відомому методі для імовірнісного блочного шифрування використовується b-бітова функція шифрування E, а зашифроване повідомлення розділяється на v-розрядні блоки даних (v <b). Для перетворення блоку відкритого повідомлення M генерується u-бітовий випадковий блок $R(u = b - v)$, за яким слідує складання b-бітового вхідного блоку даних $B = R \| B$, де знак || позначає операцію конкатенації двох довічних векторів R і M і обчислює блок зашифрованого повідомлення $C = E_K(B)$, де K - ключ шифрування. Раціональність практичного застосування алгоритму імовірнісного шифрування відноситься до наступних елементів:

1) він забезпечує більший захист від атак з використанням бекдор в використовуваних блок-шифрах;

2) він потенційно запобігає атаки з використанням деяких непередбачених вразливостей алгоритму блочного шифрування.

Слід зазначити, що в реальних пристроях шифрування використовується генератор випадкових чисел (ГВЧ) повинен бути впроваджений в якості внутрішньої частини, наприклад, в електронному циклі, що реалізує алгоритм блочного шифрування E. Таким чином, підвищення безпеки забезпечується тільки в тому випадку, коли потенційний супротивник не в стані модифікувати ГВЧ або його вихідні дані.

При використанні різних значень відносини b / v для деякої даної функції шифрування E, можна вибрати необхідний компроміс між безпекою та швидкістю шифрування. Чим більше це відношення, тим більше підвищується рівень безпеки і тим нижче швидкість шифрування даних. Останнє можна грубо оцінити за допомогою формули $s = s_0(b-u)/b$, де s_0 - швидкість шифрування. Загальна схема ймовірного блочного шифрування показана на рисунку 1.

Схема імовірнісного блочного шифрування може бути легко перетворена в схему псевдо-ймовірного блочного шифрування, яка може бути використана для одночасного шифрування двох незалежних повідомлень, фіктивних і секретних, з використанням двох різних ключів K і Q відповідно. Для цієї мети можна замінити ГВЧ деякої блокової функцією шифрування E' з блоком u-бітних вхідних даних. Замість генерації v-бітного випадкового числа R, це зашифрований v-бітний блок T секретного повідомлення (рисунок 2). При використанні блокового алгоритму шифрування E' для перетворення блоку даних T з ключем Q, отриманий

проміжний блок шифр-тексту $C_T = E_Q(T)$, буде обчислювано відрізнятись від рівномірно випадкового v-бітового двійкового вектора.

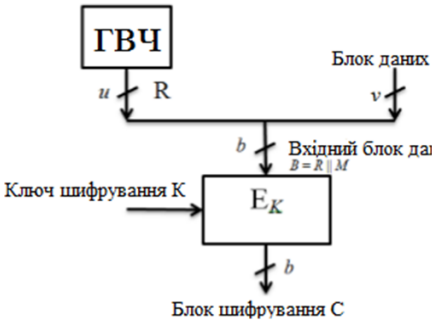

Рисунок 1 - Узагальнена схема ймовірного блочного шифрування

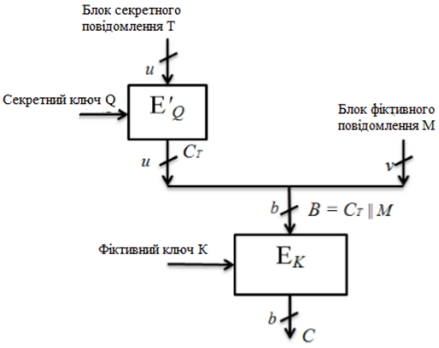

Рисунок 2- Узагальнена схема псевдо-імовірнісного блочного шифрування

Потім блок СТ об'єднується з блоком фіктивного повідомлення М і перетворюється у вихідний блок зашифрованого повідомлення:

$$C = E_K(C_T \| M).$$

При примусі відправник і одержувач повідомлення можуть відкривати фіктивне повідомлення М і підроблений ключ К і оголошувати про використання блочного імовірнісного алгоритму шифрування. Таким чином, під час примушує атаки обидві сторони мають можливість без підозр використовувати фіктивні дані.

Пропонований псевдо-імовірнісний метод шифрування забезпечує заперечність. У монографії представлені кілька методів імовірнісного шифрування з деякою детермінованою функцією блочного шифрування. Для кожного з цих методів можна запропонувати відповідну псевдо-вірогідну схему блочного шифрування, яка безпечна для атак з примусом.

Такі псевдо-ймовірні схеми шифрування відносяться до запланованих алгоритмам і протоколам заперечного шифрування. Вони забезпечують двосторонню захист до тих пір, поки противник не зможе перевірити час розшифрування, необхідне для розкриття секретного повідомлення. Якщо у нього є така можливість, то він зможе встановити, що час дешифрування підробленого повідомлення менше часу дешифрування секретного повідомлення. Крім того, можна припустити, що в деяких випадках противник порівнює функції кодування, які використовуються для дешифрування фіктивних і секретних повідомлень.

Щоб забезпечити заперечення під час атак, скоєних противником, які мають зазначені можливості, можна запропонувати наступний додатковий

критерій алгоритму шифрування: фіктивні і секретні повідомлення повинні бути розшифровані одним і тим же алгоритмом дешифрування. Цей критерій може бути виконаний за допомогою додавання додаткового перетворення фіктивного блоку повідомлення M з блочною функцією шифрування E' і установкою значень v = u = b / 2, як показано на рисунку 3, де блок $Transp^{(e)}$ виконує керовану перестановку двох блоків даних u – біт СТ і СМ: якщо e = 1, то $Transp^{(e)}(C_T \parallel C_M) = C_M \parallel C_T$, якщо e = 0, то $Transp^{(e)}(C_T \parallel C_M) = C_T \parallel C_M$.

Передбачається, що ключі K і Q задовольняють умові $(K \bmod 2) \oplus (Q \bmod 2) = 1$, де і значення e залежить від ключа наступним чином (ключі K і Q генеруються так, що вони мають різну парність): e = K mod 2 і e = Q mod 2. у цьому випадку наступний алгоритм розкриває фіктивні або секретні повідомлення в залежності від використовуваного ключа K і Q.

Імовірнісний алгоритм шифрування, який може бути пов'язаний з псевдо-імовірнісним алгоритмом шифрування (рис. 3), показаний на рис. 4.

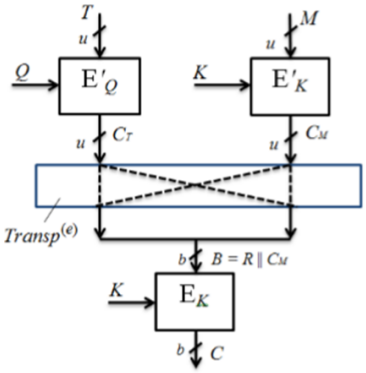

Рисунок 3 - Узагальнена схема псевдо-імовірнісного блочного шифрування

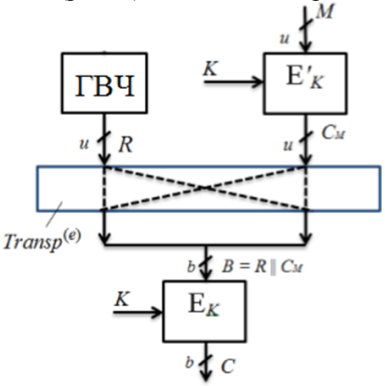

Рисунок 4 - Узагальнена схема псевдо-імовірнісного блочного шифрування

Легко бачити, що зашифроване повідомлення, створене першим алгоритмом під час одночасного шифрування секретного повідомлення T з секретним ключем Q і фіктивним повідомленням M з фіктивним ключем K, потенційно може бути отриманий другим алгоритмом, використовуваним для шифрування фіктивного повідомлення по фіктивному ключу. Щоб відрізнити псевдо-розподіл усіх шифрування від ймовірного, вимагається розкриття секретного повідомлення T. При використанні функцій блочного шифрування E' і E, наприклад TripleDES, з блоком вхідних даних, які мають розмір u = 64 і AES, з блоком вхідних даних, які мають розмір b = 128,

обчислювано важко відрізнити заперечується схему шифрування (рисунок 3) від ймовірнісної схеми (рисунок 4).

Алгоритм розшифрування, який відповідає як заперечує, так і імовірнісним схемами шифрування, виглядає наступним чином:

1. Встановити ключ K * = (K, K'), де K' = K (для розкриття фіктивного повідомлення) і K' = Q (для розкриття секретного повідомлення).

2. Обчислити біт e = K' mod 2.

3. Розшифрувати блок зашифрованого повідомлення C: B = $(B_1 \parallel B_2)$ = $E_k^{-1}(C)$, де проміжний блок B зашифрованого повідомлення представлений як поєднання u-бітних під блоків даних B_1 і B_2.

4. Виконати операцію транспонування

$$Transp^{(e)}(B_1 \parallel B_2) : (B_1^{'} \parallel B_2^{'}) = Transp^{(e)}(B_1 \parallel B_2)$$

5. Обчислити *u*-бітний блок відкритого повідомлення $M' : M' = E'^{-1}_k(B_2^{'})$.

Висновки: представлений метод псевдо-ймовірного блочного шифрування, який задовольняють додаткової вимоги до схем заперечного шифрування, яке забезпечує безпеку примусових атак з вимірюванням часу дешифрування. Додаткова вимога формується як розшифрування як секретного повідомлення, так і підробленого повідомлення одним і тим же алгоритмом розшифрування.

Література

1. Березин А. Н., Биричевский А. Р., Молдовян Н. А., Рыжков А. В. Способ отрицаемого шифрования // Вопросы защиты информации. № 2. 2013. – С. 18-21.

2. Алексеев Л.Е., Молдовян А.А., Молдовян Н.А. Алгоритмы защиты информации в СЗИ НСД ―СПЕКТР-Z‖ // Вопросы защиты информации, 2000. №3. – С. 63–68

3. ГОСТ 28147–89.Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования.

4. Емельянов Г.В. Криптография и защита информации. – Информационное общество. № 2.2005.– С. 32 -36.

5. Венбо Мао. Современная криптография. Теория и практика. – М., СПб, Киев. Издательский дом «Вильямс», 2005. – С. 76

6. C. Peikert, B. Waters. An attack on deniability?// Personal communication 1 May 2011.

Розробка лабораторного макету системи контролю та управління тепличним господарством

Перепелиця М.В.
Науковий керівник – д.т.н., доц. Прохоров О.В.
Національний аерокосмічний університет ім.. М.Є. Жуковського «ХАІ»

Тепличне господарство зараз є невід'ємною частиною існування у будь-якій розвиненій країні. Це дозволяє забезпечувати продуктами харчування та іншою рослинністю з виключенням залежності від пори року та погодних умов, а також від клімату місцевості взагалі.

Теплиця дозволяє повністю відтворити умови, які є комфортними для вирощування будь-якого сорту та виду рослин, таким чином забезпечуючи населення свіжими продуктами, які навіть можуть бути не характерними для місцевості.

У своїй більшості автоматизовані системи управління, пов'язані з тепличним господарством, розбиті на підсистеми через те, що таким чином з окремих частин складається загальне уявлення про ефективність та доцільність використання саме цієї системи, та можливості автоматизованого управління описуються саме через наявність керуючих технічних засобів елементів підсистем та можливості моніторингу у реальному часі [1].

Розроблення систем тепличного господарства є важливим завданням з автоматизації вирощування різного роду рослин, що може забезпечити стабільне джерело продукції як для ринку, так і для персональних потреб. Сучасні технічні засоби та можливості програмного забезпечення дають можливість створити систему управління тепличним господарством, яка може бути використана і в приватному будинку, і в промислових масштабах [2].

Аналізуючи існуючі системи тепличного господарства можна виділити декілька підсистем, які керують та нормалізують певні аспекти ведення аграрного господарства, а саме:

- система вентиляції;
- система полива;
- система освітлення;
- система обігріву.

Система вентиляції складається з двох керованих вентиляторів, датчика температури і вологості повітря та реле. Показання датчиків порівнюються з уставками після чого реле перемикається в нормально відкритий або нормально закритий режим. Доцільним є наявність додаткового елементу вентилювання у вигляді механічно керованого вікна.

Система поливу представлена основною ємністю та буферною ємністю для зберігання води для крапельного поливу, датчиків рівня рідини і вологості ґрунту, герметичним датчиком температури.

З основної ємності рідина за допомогою насоса перекачується в буферну ємність, щоб уникнути переливу води у резервуарах чи сухого ходу

помпи, ємності обладнані датчиками рівня рідини. З буферної ємності під дією сили тяжіння вода по поливальним трубка рівномірно розподіляється між двома окремими лотками з землею. Після чого по трубках повертається в основну ємність.

Система освітлення складається з двох світодіодних ламп, датчика освітленості і реле. Показання датчика освітленості порівнюються з уставками після чого замикається або розмикається контакт реле.

Система підігріву ґрунту являє собою кабель з самонагріванням та герметичний датчик температури. В залежності від показань герметичного датчика температури замикається або розмикається контакт реле.

Регулювання температури відбувається на стороні автоматизованої системи, бо кабель не має окремого контролера для регуляції температури.

Усі системи повинні бути розміщені у каркасі теплиці з урахуванням особливостей їх впливу на навколишнє середовище. Призначення елементів макету приведені у таблиці 1.

Для можливості віддаленого управління теплицею з використанням мережі WiFi спроектований та програмно реалізований простий веб-сервер, що дозволяє управляти пристроями в нашій локальній мережі з телефона або комп'ютера. Все, що потрібно це просто підключити реле до виходів контролера МЕГА та налаштувати обмін даними між ним та модулем wi-fi. Макет веб-сторінки, яку ми отримуємо в результаті наведений на рисунку 1.

Таблиця 1 – Перелік елементів макету

№	Елемент макету теплиці	Належність до підсистеми
1	Розподільна коробка з усіма керуючими елементами та батареєю	Підсистема управління
2	Ємність для подачі води у бак, об'єм – 5 літрів	Підсистема поливу
3	Помповий насос для перекачування води	Підсистема поливу
4	Вентилятори	Підсистема вентиляції
5	Бак, що є буферною ємністю для крапельного поливу	Підсистема поливу
6	Світлодіодні лампи	Підсистема освітлення
7	Трубки для крапельного поливу	Підсистема поливу
8	Нагрівальний кабель для теплої підлоги	Підсистема обігріву
9	Датчики верхнього та нижнього аварійного рівня рідини у баку	Підсистема поливу
10	Герметичний датчик температури	Підсистема обігріву
11	Датчик температури і вологості повітря	Підсистема вентиляції
12	Датчик вологості ґрунту	Підсистема поливу
13	Датчик освітленості	Підсистема освітлення

Рисунок 1 – Макет веб-сторінки локального веб-сервісу

При неможливості здійснення ручного управління чи контролю показників візуально на дисплеї реалізована можливість спостереження за показниками параметрів в режимі реального часу з використанням сервісу ThingSpeak. ThingSpeak - це платформа для проектів, побудованих на концепції "Інтернет речей".

Основу платформи складають канали, в які і надсилаються дані для зберігання і візуалізації.

Кожен канал включає в себе 8 полів для будь-якого типу даних, 3 поля для розташування (широта, довгота, висота), і 1 поле стану.

Для відсилання даних до сервісу ThingSpeak був створений окремий аккаунт, в якому було створено один канал з шістьома полями для відображення даних про температуру, вологість, наявність світла та рівня води.

Приклад відображення даних у сервісі наведений на рисунку 2.

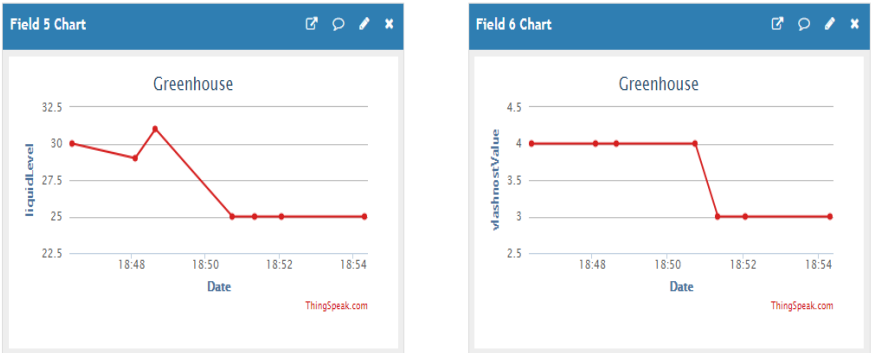

Рисунок 2 – Приклад відображення даних у сервісі ThingSpeak

Література
1. Еременко В.В. Информационное пространство отношений элементов информационной системы управления сложным техническим комплексом / В.В. Еременко // Труды международного симпозиума Надежность и качество. 2012. Т. 1. С. 35-36
2. Виноградов А.Н. Применение информационных технологий в управлении процесами потребления тепловой энергии объектами ЖКХ / А.Н. Виноградов // Труды международного симпозиума Надежность и качество. 2013. Т. 1. С. 263-265.

Архітектурні особливості обчислювальних систем з програмованою структурою

Присяжнюк В.В.
Науковий керівник – к.т.н., доц.. Огнєвий О.В.
Хмельницький національний університет

Сучасні розподілені ОС є мультиархітектурними. Залежно від рівня розгляду їх функціональних структур, вони можуть виглядати і як MISD, і як SIMD, і як MIMD системи (рис.1). Для таких систем характерні ієрархічна організація і різні пропускні спроможності каналів зв'язку між їхніми ресурсами (обчислювальними вузлами, ЕМ, процесорами і їх ядрами).

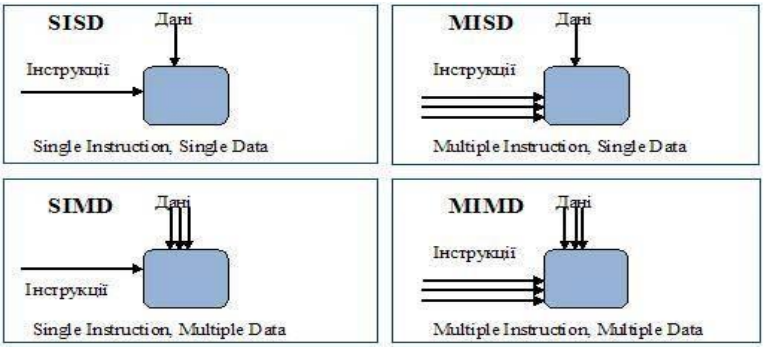

Рисунок 1 - Класифікація комп’ютерних систем по Флінну

Час виконання паралельних програм на розподілених ОС істотно залежить від того наскільки вони ефективно вкладені в систему. Під ефективним вкладенням розуміється такий розподіл гілок паралельної програми між ЕМ системи, при якому досягаються мінімуми накладних витрат на міжмашинний обмін інформацією і дисбаланс завантаження ЕМ.

При організації ефективного функціонування розподілених ОС попереду стоїть завдання розробки моделей і алгоритмів вкладення па

паралельно програм, які враховують архітектурні особливості сучасних систем.

Обчислювальні системи з програмованою структурою - це розподілені засоби обробки інформації з архітектурою MIMD (Multiple Instruction Multiple Data)- системи із множинним потоком команд і множинним потоком даних. До цього класу належать як векторні суперЕОМ, так і всі багатопроцесорні системи обробки даних (СОД). Загальна структурна схема таких систем показана на рис.2. Ця архітектура включає всі рівні паралелізму, від конвеєра операцій до незалежних операцій і команд. Вживаючи термін MIMD треба мати на увазі не тільки роботу багатьох процесорів, але і безліч обчислювальних процесів, що виконуються одночасно в ОС.

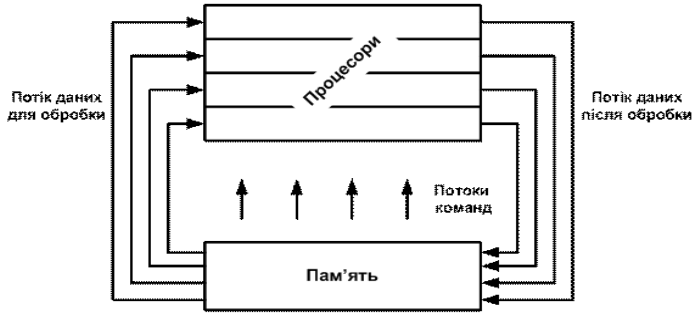

Рисунок 2 -. Загальна структура системи класу MIMD

Особливість систем в тому, що в них закладена можливість програмного переналаштування архітектури MIMD в архітектури MISD або SIMD. Основна функціонально-структурна одиниця обчислювальних ресурсів в системах розглянутого класу - це елементарна машина, яка є композицією з обчислювального модуля і системного пристрою.

Обчислювальний модуль (ОМ) служить як для обробки та зберігання інформації, так і для виконання функцій з управління системою в цілому.

Системний пристрій (СП) - це та апаратурна частина ЕМ, яка призначається тільки для забезпечення взаємодії даного ЕМ з найближчими сусідніми машинами (точніше, з системними пристроями, з якими є безпосередній зв'язок).

Допускається конфігурація ОС з довільним числом ЕМ. Отже, ОС з програмованою структурою відносяться до засобів обробки інформації що допускають формування конфігурацій з масовим паралелізмом (Scalable Massively Parallel Architecture Computing Systems).

Взаємодія між ЕМ здійснюється через програмно налаштовану мережу зв'язку. Структура ОС описується графом *G=(C,E)*, множина вершин *C* якого - кількість елементарних машини (або системних пристроїв, або локальних комутаторів), а множина ребер *E*- лінії міжмашинного зв'язку.

До структур сучасних ОС висувається ряд вимог [1].

1) Простота вкладення паралельного алгоритму розв'язання складної задачі в структуру ОС. Структура ОС повинна бути адекватна досить широкому класу вирішуваних задач; настройка проблемно-орієнтованих віртуальних змін не повинна бути пов'язана зі значними накладними витратами.

2) Зручність адресації елементарних машин і «перенесення» підсистем в межах обчислювальної системи. Обчислювальна система повинна надавати можливість користувачам створювати паралельні програми з віртуальними адресами ЕМ.

3) Здійсненність принципу близькодії і мінімуму затримок при міжмашинних передачах інформації в ОС. Принцип близькодії перевизначає реалізацію обмінів інформацією між «віддаленими» один від одного ЕМ через проміжні машини системи. Отже, в умовах обмеженості числа зв'язків, структура ОС повинна забезпечувати мінімум затримок при «транзитних» передачах інформації.

4) Масштабованість і великомасштабні структури ОС. Для формування конфігурацій ОС із заданою ефективністю потрібно, щоб структура була здатна до нарощування і скорочення числа вершин (машин). Зміна числа ЕМ в ОС не повинна призводити до корінних перекомутацій між машинами і (або) до необхідності зміни числа зав’язків для будь-яких ЕМ.

Для досягнення високої продуктивності ОС при існуючих можливостях мікропроцесорної техніки потрібно число ЕМ близько 10-106. Для підтримки великомасштабних ОС (масового паралелізму) необхідно, щоб структура ОС була здатна ефективно здійснювати міжмашинний обмін інформацією в умовах неможливості реалізації зв’язків повного графу (наприклад, через обмеженість числа виводів в корпусах ВІС).

5) Комутація структури ОС. Обчислювальна система повинна бути пристосована до реалізації групових міжмашинних обмінів інформацією. Отже, структура ОС повинна мати здатність здійснювати вказану кількість одночасних непересічних взаємодій між елементарними машинами.

6) Живучість структури ОС. Важливою вимогою до ОС в цілому являється забезпечення працездатності при відмові її компонентів або навіть підсистем. Основою функціональної цілісності ОС, як колективу елементарних машин, є живучість структури. Під останньою розуміють здатність структури ОС забезпечити зв'язність необхідного числа робочих ЕМ в системі при ненадійних лініях міжмашинних зв’язків.

7) Технологічність структур ОС. Структура мережі міжмашинних зв'язків ОС не повинна пред'являти особливих вимог до елементної бази, до технології виготовлення мікропроцесорних ВІС. Системи повинні бути сприйнятливі до масової технології, їх «обчислювальне ядро» має формуватися з масових мікропроцесорних ВІС. Останнє дозволить досягти прийнятних значень техніко-економічних показників ОС.

Література
1. Абрамов В.О. Архітектура електронно-обчислювальних машин / В.О. Абрамов.: навч. посіб.- К.: КМПУ імені Б.Д. Грінченка, 2007. 84 с.
2. Ясько М.М. Паралельні та розподілені обчислення / М.М. Ясько. - Д.: РВВ ДНУ, 2010. 111с.

Аналіз та дослідження форматних стеганоалгоритмів на основі графічних контейнерів

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

Форматні методи, по загальновизнаному висновку більшості фахівців, не відносяться до цифрової стеганографії в чистому вигляді, оскільки не використовують методи цифрової обробки сигналів. Вони засновані на надмірності форматів комп'ютерних даних, наприклад, структурі файлів, ір-пакетів. Хоча цифрові зображення теж є сигналами, але вони мають «застиглий» характер. Тому при роботі з будь-яким алгоритмом, який додає в кінець файлу JPEG байти файлу RAR, не можна строго говорити про цифрову стеганографію зображень, оскільки це є використанням форматного методу в комп'ютерній стеганографії. Тому пропонується наступне визначення форматного методу в цифровій стеганографії зображень: метод перетворення зображення, причому впроваджувана інформація не ініціює візуалізацію артефактів вбудовування інформації, але обов'язково враховується і використовується при декодуванні файлу зображення спираючись на специфікацію формату JPEG.

Розроблений стеганографічний алгоритм використовує запропоноване визначення форматного підходу для впровадження інформації в цифрове зображення. Відмітимо, що впровадження інформації відбувається не з потоком повідомлення, а із службовою і сигнальною інформацією, що не несе смислового навантаження.

Стеганоалгоритми, які використовуються в просторовій області, вбудовують інформацію в область самого зображення. Перевага в тому, що для впровадження інформації не потрібно проводити обчислювально-трудомісткі лінійні перетворення зображень. Інформація вбудовується маніпуляціями колірними складовими $(r(x,y), b(x,y), g(x,y))$ або яскравістю $l(x,y) \in \{1,\ldots,L\}$. Ці алгоритми були розроблені, коли широко застосовувався графічний формат BMP, в якому закладений механізм зберігання інформації про колірні складові кожного пікселя в зображенні. Різновид BMP-форматів полягав в кодуванні колірних складових, тобто в кількості кольорів, півтонів і відтінків в конкретному BMP-зображенні. Тут для кодування кольору кожного пікселя виділяється три байти. Кожен байт потенційно дозволяє

закодувати 28-256 відтінків кольору. А три байти можуть представити один з 2563 = 16777216 мільйонів відтінків кольору.

Серед всіх лінійних ортогональних перетворень найбільшу популярність в стеганографії отримали вейвлет-перетворення і ДКП, що частково пояснюється їх успішним використанням при стискуванні зображень. Крім того, бажано застосовувати для приховування даних те ж перетворення зображення, як і те, якому воно піддасться при можливому подальшому стискуванні. Стеганоалгоритм може бути вельми робастним до подальшої компресії зображення, якщо він враховуватиме особливості алгоритму стискування. При цьому, звичайно стеганоалгоритм, використовуючий ДКП, зовсім не обов'язково буде робастним по відношенню до вейвлетному алгоритму стискуванню. Стеганоалгоритм, використовуючий вейвлети, може бути неробастним до стискування із застосуванням ДКП. Ще більші труднощі з вибором перетворення при приховуванні даних у відеопослідовності. Причина полягає в тому, що при стискуванні відео основну роль грає кодування векторів компенсації руху, а не лише нерухомого кадру. Робастний стеганоалгоритм повинен якимсь чином враховувати це.

Відомо багато моделей для оцінки пропускної спроможності каналу приховування даних. Розлянемо наступну модель: Нехай *S0* - початкове зображення (контейнер), *W*- вкладення. Тоді модифіковане зображення *SW=S0+W*. Модифіковане зображення візуально не відрізняється від початкового і може бути піддане стискуванню з втратами: *SW=C(SW)*, де *C()* - оператор компресії. Біти вкладення *W* мають витягуватись з $\tilde{S}_W$.

Блок-діаграма даного стеганоканала представлена на рисунку 1.

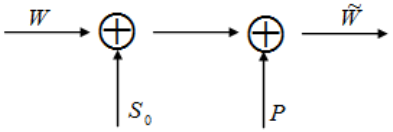

Рисунок 1 - Блок-діаграма стеганоканала

Повідомлення *W* передається по каналу. Канал має два джерела «шуму»: *S0* - зображення-контейнер і *P* - «шум», що виникає при компресії/декомпресії. $\tilde{W}$ - можливо спотворене повідомлення.

Структурна схема схема стеганосистеми приведена на рис. 2. Зображення декомпозується на *L* субсмуг. До кожної субсмуги «підмішується» прихована інформація. Після зворотного перетворення виходить модифіковане зображення *SW*. Після компресії/декомпресії

виходить зображення $\tilde{S}_W$. Воно піддається прямому перетворенню, і з кожної L субсмуг незалежно витягується приховане повідомлення.

Реальні зображення не є випадковим процесом з рівномірно розподіленими значеннями величин. Добре відомо, і це використовується в алгоритмах стискування, що велика частина енергії зображень зосереджена в низькочастотній частині спектру. Звідси і потреба в здійсненні декомпозиції зображення на субсмуги. Стеганоповідомлення додається до субсмуг зображення. Низькочастотні субсмуги містять більшу частину енергії зображення і, отже, носять шумовий характер. Високочастотні субсмуги найбільш схильні до дії з боку різних алгоритмів обробки, будь то стискування або НЧ фільтрація. Таким чином, для вкладення повідомлення найбільш відповідними кандидатами є середньочастотні субсмуги спектру зображення.

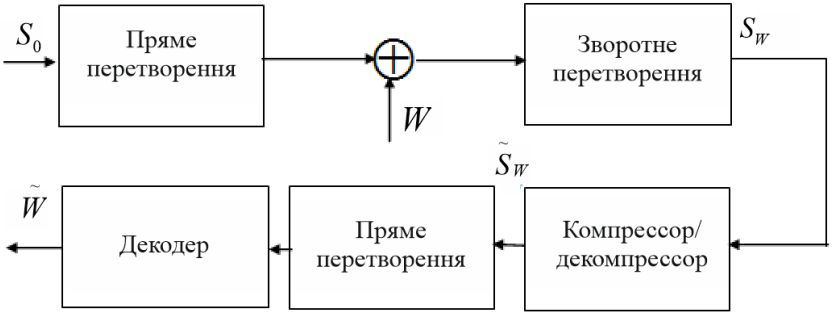

Рисунок 2 - Структурна схема схема стеганосистеми

Найбільший виграш дає перетворення Карунену-Лоева (ПКЛ) – розкладання по базису одиничного імпульсу (тобто відсутність перетворення). Перетворення, що мають високі значення виграшу від кодування, такі як ДКП, вейвлет-перетворення, характеризуються різко нерівномірним розподілом дисперсій коефіцієнтів субсмуг. Високочастотні субсмуги не підходять для вкладення через великий шум обробки, а низькочастотні – через великий шум зображення. Тому доводиться обмежуватися середньочастотними смугами, в яких шум зображення приблизно дорівнює шуму обробки. Оскільки таких смуг небагато, то пропускна спроможність стеганоканала невелика. В разі використання перетворення з нижчим виграшем від кодування, наприклад, Адамара або Фур'є, є більше блоків, в яких шум зображення приблизно дорівнює шуму обробки. Отже, і пропускна спроможність вища. Для підвищення пропускної спроможності стеганографічного каналу краще застосовувати перетворення з меншими виграшами від кодування, які погано підходять для стиску сигналів.

Ефективність використання вейвлет-перетворення і ДКП для

стискування зображень пояснюється тим, що вони добре моделюють процес обробки зображення в СЧЗ, відокремлюють «значимі» деталі від «незначимих». Отже, їх доцільніше застосовувати в разі активного порушника. Насправді, модифікація значимих коефіцієнтів може привести до неприйнятного спотворення зображення. При використанні перетворення з низькими значеннями виграшу від кодування існує небезпека порушення вкладення, оскільки коефіцієнти перетворення менш чутливі до модифікацій.

ДКП застосовують як до всього зображення в цілому, так і до окремих блоків точок зображення. Зазвичай контейнер розбивають на блоки розміром 8x8 пікселів. Потім до кожного блоку застосовують ДКП. Отримані матриці коефіцієнтів ДКП мають розмір 8x8.

Дослідження показали, що область перетворення погано підходить для впровадження великих об'ємів даних. Проте область перетворення добре підходить для впровадження ЦВЗ, що являють собою невелику послідовність з байтів. Принциповою вимогою є також і безліч обмежень, що пред'являються до контейнера.Алгоритми, які застосовуються до просторової області зображення, базуються на візуальній надмірності сприйманої інформації і тому в даний час є не такими популярними, як алгоритми області перетворення. Це пов'язано з великим поширенням формату JPEG, де колірні і яскраві складові пікселей приховані за областю перетворення.

Запропонована метрика оцінки спотворень зображень, що забезпечує об'єктивність порівняльного аналізу стійкості різних стеганоалгоритмів ДВП області вбудовування. Область перетворення, за рахунок невеликого числа потенційних місць для впровадження, більше личить для впровадження невеликої кількості інформації, наприклад ЦВЗ.

За результатами виконаного порівняльного аналізу стійкості ЦВЗ, які були вбудовані різними стеганографічними алгоритмами ДКП області вбудовування, зроблений висновок, що стійкість ЦВЗ не більше *Kjpeg-2000=60* для цифрових водяних знаків, представлених рядком символів в 8-мі бітовому кодуванню, і *Kjpeg-2000=50* для цифрових водяних знаків, представлених бітами. Контейнер з *Kjpeg-2000=50* можливо застосовувати для комерційного використання.

Література

1. Грибунин В.Г. Цифровая стеганография. /В.Г.Грибунин, И.Н.,Оков И.В.,Туринцев // М.: СОЛОН-Пресс; 2002. - 261 с.

2. Конахович Т.Ф. Компьютерная стеганография / Т.Ф Конахович, А.Ю Пузыренко // Теория и практика. Киев: МК-Пресс, 2006. -288с.

3. Мамаев М. Технологии защиты информации:в Интернете/ Мамаев М., Петренко С. //: Специальный;справочник..СИб::Иитер 2002. -848 с.

4. Аграновский А.В. Стеганография, цифровые водяные знаки и стеганоанализ. /А.В. Аграновский, А.В. Балакин, В.Г. Грибунин - М.: Вузовская книга 2009. - 220 с.

Технології надання доступу до сервісів розподіленої хмарної системи

Сівак А.С.
Науковий керівник: к.т.н., доц. Муляр І.В.
Хмельницький національний університет

На сучасному етапі розвитку української держави актуальною є проблема надання доступу до сервісів розподіленої хмарної системи [1]. Актуальність дослідження зумовлена необхідністю підвищення доступності хмарних сервісів на базі клієнт-серверної й однорангової розподіленої хмарних архітектур. У контексті даного дослідження необхідним є пошук алгоритмів і засобів використання хмарних систем і мережних додатків, провайдерами хмарних послуг, а також підприємствами, які розгортають власні хмарні інформаційні сервіси.

Так принцип розподілу ресурсів між вузлами хмарної системи з децентралізованою структурою з метою підвищення оперативності відповіді на запит покладено в основу технології надання доступу до сервісів однорангової розподіленої хмарної системи.

Варто зазначити, що в локальній мережі або в мережі Інтернет розподіл реалізується за допомогою протоколу Kademlia [2] і містить процеси публікації, реплікації і надання доступу до ресурсів. Інформація, яка зберігається на дисковому просторі робочої станції, у вигляді відповіді на запит розподіленої хмарної системи, надається в спільне користування іншим вузлам-учасникам. Шляхом модифікації ресурсних записів процес розподілення ресурсів координується за допомогою DNS-сервера. На робочих станціях всіх вузлів-учасників одногранної розподіленої хмарної системи встановлено спеціалізоване програмне забезпечення, яке організовує взаємодію вузлів. З урахуванням завантаженості каналів зв'язку, доступ до ресурсів для вузлів, які не беруть участі в процесі, надається за допомогою їх

Розглянемо взаємодію основних компонентів для отримання доступу до веб-ресурсу http://domain-name.com/resource-name. Перш за все вузол-учасник виконує стандартний запит до ресурсу http://domain-name.com/resource-name через веб-браузер на своїй робочій станції, далі через прикладне ПЗ виконується запит до додаткового ресурсного запису DNS-сервера ідентифікатора ресурсу ID_res та до ідентифікаторів базових вузлів реплікації. Це необхідно для запуску стандартного процесу пошуку DHT-lookup «FIND_NODE» найближчого за XOR-метрикою вузла реплікації ресурсу з ідентифікатором ID_res на наступному етапі. Наступним кроком буде виконання запиту «FIND_VALUE» на надання доступу до ресурсу до вузла з відповідним ідентифікатором. На прикінцевому етапі через веб-браузер і збереження відповіді на запит відбувається завантаження контенту для запитувача вузла на дисковому просторі цього вузла.

Розглянемо основні типи ідентифікаторів: ідентифікатор вузла (ID_node), глобальний ідентифікатор ресурсу (доменне ім'я) й унікальний

ідентифікатор ресурсу (ID_res) в одноранговій мережі.

Незалежно від ролі вузла, при додаванні кожного нового в систему, для нього генерується унікальний ідентифікатор вузла ID_node і дві пари публічних і приватних (відкритих і закритих) ключів: Kpub1, Kpriv1, Kpub2, Kpriv2 на основі алгоритму RSA-256 або RSA-128. Перша ключова пара використовується для стандартних процесів комунікації, друга – для процесу валідації ідентифікатора вузла з метою захисту від підміни особистості.

Ідентифікатор вузла генерується у такий спосіб [3]:

$$ID\_node = Hash\ (Kpub1 + Sig_{Kpriv2(Kpub1)})$$

де *ID_node* – ідентифікатор вузла; *Kpub1* – відкритий ключ 1; *Kpriv2* – закритий ключ 2; *Hash (Data)* – оператор обчислення хеш-функції даних Data; $Sig_{Kpriv2(Kpub1)}$- цифровий підпис даних *Kpub1*,який отримано за допомогою ключа *Kpriv2*.

Унікальний ідентифікатор ресурсу – хеш-функція його вмісту, що обчислюється за допомогою SHA (2)-256:

$$ID\_res = Hash\ (Content\ (res)),$$

де *ID_res* – ідентифікатор ресурсу; *Hash (Data)* – оператор обчислення хеш-функції даних Data; *Content (res)* – вміст ресурсу res.

Публікація ресурсу може здійснюватися тільки власником цього ресурсу. Даний процес полягає у відкритті доступу до інформації, що публікується, додаванні DNS-запису відповідності адреси ресурсу і доменного імені. Після отримання доступу безпосередньо до ресурсу або його частини, інформація зберігається на дисковому просторі вузла-учасника у вигляді відповіді на запит, і такий вузол може надати цей ресурс за запитом іншим вузлам мережі за допомогою протоколу Kademlia, тобто виступити в ролі сервера. Для цього його ідентифікатор додається в глобальну розподілену хеш-таблицю (DHT) ідентифікаторів ресурсів [2].

У разі, якщо інший вузол-учасник однорангової розподіленої хмарної системи запитує той самий ресурс, він отримує доступ до нього за ідентифікатором найближчого за XOR-метрикою [2] вузла реплікації, що володіє ресурсом частково або повністю, , що містяться в ресурсних записах DNS для даного ресурсу. Якщо, на дисковому просторі вузол-учасник запитує всі частини ресурсу його робочої станції, то буде закешовано весь ресурс повністю. Тоді цей вузол буде мати повну репліку ресурсу, а його адреса буде додана до основного ресурсного запису (A-запису) DNS-сервера. Отже, значна кількість вузлів-учасників однорангової розподіленої хмарної системи, зі збільшенням популярності ресурсу, може ним поділитися. Для виконання серверних функцій кожен вузол-учасник має надати в спільне користування частину власних апаратних ресурсів – дискового простору й потужності центрального процесора. Якщо вузли не є учасниками системи, вони можуть отримати доступ до ресурсу завдяки способу взаємодії з сервером власника ресурсу або вузлами, що володіють повною реплікою

ресурсу, за їхніми адресами з урахуванням завантаженості каналів зв'язку і станції власника ресурсу або вузлів.

Перевірка цілісності ресурсу та ідентичність реплік проводиться шляхом обчислення хеш-функції отриманого ресурсу і зіставлення зі значенням ідентифікатора цього ресурсу. Крім того, на проміжному етапі перед завантаженням контенту виробляється процес валідації вузла, що надає ресурс. Вузол, що надає ресурс, для підтвердження свого ідентифікатора відправляє повідомлення такого формату [2]:

ID_node	
Kpub1	SigKpriv2(Kpub1)
Kpub2	SigKpriv2(Kpub2)

Data

На першому кроці валідації ідентифікатора вузла виконується перевірка автентичності переданих відкритих ключів Kpub1 і Kpub2 за рахунок механізму цифрового підпису. Якщо перевірка проходить успішно, то далі перевіряється істинність співвідношення [3]:

$$ID\_node = Hash\ (Kpub1 + Sig_{Kpriv2}(Kpub1))$$

де *ID_node* – ідентифікатор вузла, переданий в повідомленні; *Kpub1* – відкритий ключ 1; *Hash (Data)* – оператор обчислення хеш-функції даних *Data*; *SigKpriv2(Kpub1)* – цифровий підпис даних *Kpub1*, отриманий за допомогою ключа *Kpriv2*.

Якщо рівність виконується, то збір інформації проходить успішно і вузол-відправник вважається успішно ідентифікованим і підтвердженим.

Незалежно від ролі, будь-який вузол-учасник однорангової розподіленої хмарної системи отримує високошвидкісний доступ до всіх ресурсів системи, завдяки можливості отримати ресурс від найближчого вузла-учасника, який має копію цього ресурсу. Описані процеси додавання нового вузла, збору інформації, публікації ресурсу й отримання доступу до ресурсу подано у вигляді поетапної послідовності дій з надання доступу до сервісів розподіленої хмарної системи за допомогою інформаційних потоків, взаємодії процесів оброблення інформації та об'єктів, які є частиною цих процесів.

Таким чином, архітектура реалізує новий підхід до розподілу ресурсів в хмарної мережі з децентралізованою структурою, об'єднуючи переваги технологій GRID, хмарних обчислень і пірінгових мереж. Її особливістю є самоорганізація процесу реплікації ресурсів засобами робочих станцій вузлів-учасників системи: даний процес не вимагає втручання адміністратора або сторонніх механізмів.

Література

1. Технологии Web, Grid, Cloud для гарантоспособных ИТ-инфраструктур[Текст] : монография / В. С. Харченко и др; Харьков. нац.

аэрокосм. ун-т им. Н. Е. Жуковского «ХАИ», 2013. 868 с.

2. Zhou, S. Liu, and G. Huang, Kad-D: An Improved Model Based on Kademlia, Multimedia Information Networking and Security (MINES), 2011 Third International Conference on, 2011. P.123-127.

3. Основы зеленой ИТ-инженерии. Моделирование облачных систем. Практикум / Харченко В.С., Дрозд А.В., Поночовный Ю.Л., Яновская О.В., Яновский М.Э., Кривцов А.Ю., Иванченко О.В. / Под ред. Харченко В.С. Министерство образования и науки Украины, Нац. аэрокосмический ун-т им. Н.Е. Жуковского «ХАИ». 2016. 168 с.

Аналіз мережевого трафіку за допомогою сніфер-програм

Сташков Д.В.
Науковий керівник – к.т.н. доц. Бойчук В.О.
Хмельницький національний університет

Аналізатор трафіку, сніфер – це програма, яка перехоплює мережевий трафік який проходить через мережеву карту. Програма призначена для діагностики мережі, завдяки чому вона часто використовується системними адміністраторами. З іншого боку, сніфери також використовуються для несанкціонованого доступу до даних і перехоплення паролів. Ці системні програмні засоби працюють на рівні мережевого адаптера NIC (Network Interface Card).

Перехоплення трафіку може здійснюватися:

– звичайним «прослуховуванням» мережевого інтерфейсу (метод ефективний при використанні в сегменті концентраторів (хабів) замість комутаторів (світчів), інакше метод малоефективний, оскільки на сніфер потрапляють лише окремі фрейми);

– підключенням сніфера в розрив каналу;

– відгалуженням (програмним або апаратним) трафіку і спрямуванням його копії на сніфер;

– через атаку на канальному (MAC-spoofing) або мережевому рівні (IP-spoofing), що приводить до перенаправлення трафіку або всього трафіку сегменту на сніфер з подальшим поверненням трафіку в належну адресу.

За замовчуванням мережева плата комп'ютера бачить тільки то, що призначене саме для неї. Однак сніфери встановлюють її в режим прийому всіх пакетів (promiscuous mode) та змушують мережеву плату приймати всі пакети, незалежно від того, кому вони адресовані. Перехоплений трафік передається декодеру пакетів, який ідентифікує та розкладає пакети за відповідними рівнями. Залежно від можливостей конкретного сніферу, перехоплена інформація про пакети може пізніше аналізуватися та фільтруватися.

Аналіз трафіку, що пройшов через сніфер, дозволяє:

– виявити паразитний, вірусний і закільцований трафік, наявність якого збільшує завантаження мережного устаткування і каналів зв'язку;

– виявити в мережі шкідливе і несанкціоноване ПЗ, наприклад, мережеві сканери, флудери, троянські програми, клієнти пірінгових мереж тощо;

– перехопити будь-який незашифрований (а інколи і зашифрований) призначений для користувача трафік з метою отримання паролів і іншої інформації;

– локалізувати несправність мережі або помилку конфігурації мережних агентів.

На рис. 1 показано процес узгодження ключів сеансів у випадках RSA, а також інформацію, яку бачить сніфер в перехопленому SSL трафіку.

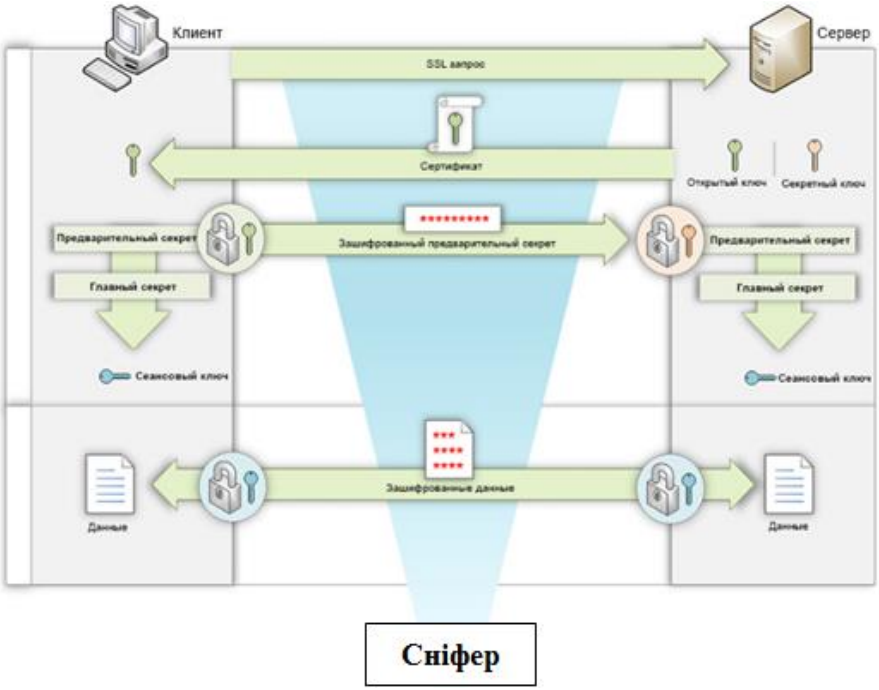

Рисунок 1 – схема передачі інформації між клієнтом та сервером

Сніфер для операційних систем Windows, для захвату пакетів використовує функції socket-функції recv(під Microsoft Visual Studio) - recv(RS_SSocket, (char*)&RS_Buffer[0], sizeof(RS_Buffer), 0), в Unix-системах використовується утиліта tcpdump.

Клієнт і сервер формують сеансовий ключ для подальшого розшифрування даних які вони будуть отримувати один від одного. Сніфер використовуючи відкритий ключ, який він перехоплює під час передачі сертифікату сервером до клієнта, на основі якого створюється попередній та головний секрет (генерація випадкових чисел чи рядків) має змогу отримати дані і про сеансовий ключ. У випадку RSA використовується ключова пара, відкритий та закритий ключ:

Ці ключі зв'язані між собою певним математичним чином, тому знаючи сеансовий ключ клієнта є змога отримати і сеансів ключ сервера. Далі на основі цих ключів і відбувається розбір зашифрованих даних.

Отже, використання сніфер-програм дозволяє здійснити аналіз трафіку, який проходить через мережеву карту. Також програми використовуються адміністраторами для запобігання різного роду проблем які виникають при проходженні даних по мережі. Для збереження своєї інформації потрібно використовувати комутатори та захищені протоколи передачі даних.

Література

1. Пахомов С. Анализаторы сетевых пакетов :[Електронний ресурс] / С. Пахомов. – Режим доступу : http://compress.ru/Archive/CP/2006/4/48.

2. Побегайло А.П. Системное программирование в Windows / А.П. Побегайло. – СПб.: БХВ-Петербург, 2006. – 1056 с.

3. Принцип работы сниффера. Анализаторы трафика : [Електронний ресурс]. – Режим доступу: http://www1.hut.ru/aneksniff/sniffer2.html.

Підхід до функціонального діагностування цифрових процесорів зі скороченою системою команд

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

Вибір способу саме функціонального діагностування [1] стосовно цифрових процесорів зі скороченою системою команд (ЦПССК) є повністю обґрунтованим з урахуванням внутрішньої складності як ЦПССК об'єктів діагностування, що унеможливлює їх повноцінне тестове детерміноване діагностування та робить малоефективними методи імовірнісного діагностування.

Оскільки основним призначенням ЦПССК є обробка даних, яка виконується програмно-керовано із застосуванням команд з системи команд процесора, то в функціональному діагностуванні процесора можна виділити два основних завдання:

– перевірка здатності діагностуваного ЦПССК коректно виконувати

всі команди з системи команд процесора;

– перевірка здатності діагностуваного ЦПССК коректно виконувати операції з даними, що передбачені кожною застосовуваною командою.

Зрозуміло, що ідеальним варіантом вирішення цих завдань з точки зору повноти діагностичного покриття несправностей ЦПССК є організація його діагностування шляхом перевірки коректності виконання всіх команд з системи команд процесора в різних варіантах їх комбінування (для виявлення можливих взаємозв'язків та взаємовпливів між командами) та з усіма можливими наборами даних.

При зовнішній привабливості такого підходу виникає ряд суттєвих проблем стосовно його застосування:

– загальна кількість елементарних командних перевірок ЦПССК при виконанні всіх його команд в різних комбінаціях та з усіма можливими наборами даних визначається надзвичайно великими комбінаторними числами (навіть з урахуванням того факту, що система команд процесора ідентифікується як скорочена і відрізняється простотою виконання);

– час діагностичних випробувань ЦПССК при реалізації отримуваної кількість елементарних командних перевірок стає також надзвичайно великим і несумісним з допустимими обмеженнями на реалізацію діагностичного експерименту (навіть з урахуванням того факту, що робочі частоти діагностованих процесорів є досить високими і команди характеризуються операційною простотою і, через це, малим часом виконання);

– детально спланувати і проаналізувати хід такого діагностичного експерименту фахівцям майже неможливо навіть із застосуванням для розв'язування типових задач засобів автоматизації з елементами штучного інтелекту.

Зазначені ускладнення "лобової атаки" призводять до потреби розробки оптимізованих методів діагностування ЦПССК [2-5], до числа яких слід віднести методи функціонального діагностування.

При аналізі пропонованого підходу до функціонального діагностування ЦПССК будемо оперувати наступними математичними описами, виконаними із застосуванням інструментарію теорії множин:

– W:{$w1$,$w2$,…,wj,…,wn} – множина функціональних вузлів ЦПССК при його розгляді як об'єкта діагностування;

– K:{$k1$,$k2$,…,kg,…,ks} – множина команд ЦПССК;

– V:{$v1$,$v2$,…, vq,…,vt} – множина елементарних процесорних операцій ЦПССК.

Аналізуючи поставлені завдання перевірки здатності діагностуваного ЦПССК коректно виконувати всі команди $kg \in K$ і, як супутній наслідок, коректно опрацьовувати дані при їх виконанні, та враховуючи архітектурні і функціональні особливості досліджуваних процесорів як об'єктів

діагностування, можна уточнити основні задачі:

– перевірка функціональних вузлів об'єкта діагностування $wj \in W$ на справність;

– перевірка типових елементарних внутрішніх операцій процесора $vt \in V$ на виконуваність.

За суттю, ці два варіанти перевірок є дуже тісно пов'язаними, оскільки перевірка вузлів процесора $wj \in W$ на справність передбачає і реалізується через перевірку типових елементарних внутрішніх операцій процесора $vt \in V$ на виконуваність зазначеними вузлами, але такий розподіл є повністю виправданий з точки зору програмно-керованого (точніше, командно-керованого) функціонального діагностування ЦПССК. Фактично, при командно-керованому функціональному діагностуванні ЦПССК кожна виконувана команда $kg \in K$ залучає в роботу ряд функціональних вузлів процесора $wj \in W$. При цьому команда $kg \in K$ активізує кожен вузол $wj \in W$ на виконання характерної для $kg \in K$ операції $vt \in V$ (сукупності операцій $vt \in V$), що не забезпечує перевірку командою $kg \in K$ всіх операцій $vt \in V$ кожного задіяного вузла $wj \in W$ (наприклад, команда паралельного запису в регістр не перевіряє здатність регістра виконувати операції зсуву). Таким чином, команди $kg \in K$ виконують лише часткові перевірки вузлів $wj \in W$, а для повної їх перевірки необхідно використати певний набір команд з множини K. Реалізація цілеспрямованої перевірки окремого функціонального вузла об'єкта діагностування $wj \in W$ на здатність виконувати всі характерні для нього типові елементарні процесорні операцій $vjt \in V$ дозволить встановити справність саме цього вузла $wj \in W$, але задіюваний для цього набір команд $kg \in K$ одночасно буде задіювати інші вузли процесора $wh \in W$, кожен з яких буде виконувати властиві йому типові елементарні процесорні операції з числа $vhd \in V$. При цьому, практично, абсолютно гарантованим фактом є те, що деякі або всі задіяні вузли $wh \in W$ будуть виконувати не всі характерні для них типові елементарні процесорні операції $vhd \in V$, що не дозволяє говорити про повноцінність виконання перевірок функціонального вузла об'єкта діагностування $wh \in W$ при комплексному виконанні перевірок функціонального вузла $wj \in W$ ($j \neq h$). Оскільки супутні результати, отримувані при виконанні повного комплексу перевірок коректності виконання типових елементарних процесорних операцій $vjt \in V$ функціональним вузлом $wj \in W$, які можна використати для оцінки коректності виконання іншими функціональними вузлами $wh \in W$ ($j \neq h$) хочаб частини характерних для них типових елементарних процесорних операцій $vhd \in V$, ігнорувати недопустимо, виникає потреба відокремленого обліку перевірок функціональних вузлів ЦПССК $wj \in W$ та виконуваних цими вузлами типових елементарних процесорних операцій $vjt \in V$. Це дозволить накопичувати дані про результати перевірок командами $kg \in K$ виконання типових елементарних

процесорних операцій $vjt \in V$ функціональним вузлом $wj \in W$ навіть за умови, що його перевірки будуть виконуватись не систематизовано в часі діагностичного експерименту або як супутні при перевірці інших функціональних вузлів об'єкта діагностування, та фіксувати момент завершення перевірки вузла $wj \in W$ як накопичувальний результат перевірок всіх його типових операцій $vjt \in V$.

Спостережуваність результатів елементарних перевірок при організації функціонального діагностування ЦПССК досягається через відображення характеру впливу реалізовуваних функціональними вузлами $wj \in W$ при виконанні команд $kg \in K$ типових елементарних процесорних операцій $vjt \in V$ над певними заданими наборами даних. При цьому здатність сукупності задіяних функціональних вузлів процесора $wj \in W$ (виконавчих, керуючих і допоміжних) виконувати більшість операцій $vjt \in V$ може бути визначена лише на одному наборі даних (наприклад, для перевірки здатності регістра виконувати операцію зсуву достатньо виконати її на одному двійковому слові даних), а гарантована перевірка коректності виконання функціональним вузлом процесора $wj \in W$ операцій $vjt \in V$ може бути виконана лише при при дослідженні його поведінки на всіх можливих наборах даних або спеціально підібраних оптимізованих наборах даних. Оскільки дослідження поведінки функціональних вузлів об'єкта діагностування на всіх можливих наборах даних є нераціональною, розроблено багато методів формування оптимізованих тестових наборів даних для всіх видів вузлів цифрової електроніки, на більш детальному аналізі розгляді яких зупинятися не будемо.

При визначенні загальних принципів реалізації функціонального діагностування ЦПССК будемо використовувати положення, що перевірка функціональних вузлів $wj \in W$ при реалізації команд $kg \in K$ на здатність виконувати типові процесорні операції $vjt \in V$ зводиться до виконання цих операцій над заданими мінімально-необхідними контрольними наборами даних, а не до виконання повної перевірки для різних наборів даних. Тобто, перевіряється лише здатність виконувати досліджувані операції, а не здатність виконувати ці операції над всіма наборами даних, хоча повна або оптимізована мінімально-достатньо-повна перевірка залишається допустимою за потреби та з використанням інших існуючих методів організації розширених (поглиблених) діагностичних перевірок обраного типу вузлів цифрового процесора.

Визначившись з загальними принципами організації перевірок ЦПССК, слід звернути увагу на порядок організації елементарних командних перевірок об'єкта діагностування (тобто, на принципах визначення пріоритетів щодо використання команд $kg \in K$ в діагностичному експерименті).

Для визначення порядку організації діагностичного експерименту

введемо функцію пріоритету (ваги) команди в діагностичних випробуваннях ЦПССК.

Функція ваги команди $kg \in K$ в діагностичних випробуваннях ЦПССК повинна пов'язуватись зі здатністю команди перевіряти функціональні вузли об'єкта діагностування $wj \in W$ і його типові операції $vt \in V$. Дані щодо здатності команд $kg \in K$ перевіряти функціональні вузли об'єкта діагностування $wj \in W$ і типові операції $vt \in V$ систематизовано в матрицях Pw і Pv відповідно, тому для розрахунку значень функцій пріоритету команд $kg \in K$ використаємо дані цих матриць:

$$F_{k_g \in K} = \alpha_w \sum_{j=1}^{|W|} p(w)_{gj} + \alpha_v \sum_{q=1}^{|V|} p(v)_{gq} , \qquad (1)$$

де $F_{k_g \in K}$ – значення функції пріоритету команди $kg \in K$; αw і αv – вагові коефіцієнти важливості перевірок функціональних вузлів $wj \in W$ і типових елементарних мікрооперацій $vt \in V$ процесора відповідно (задаються апріорно перед початком діагностичного експерименту); $p(w)_{gj}$ і $p(v)_{gq}$ – елементи матриць Pw і Pv відповідно.

Коефіцієнти важливості перевірок вузлів $wj \in W$ і мікрооперацій $vt \in V$ процесора αw і αv дозволяють регулювати надання переваги перевірці саме вузлів $wj \in W$ (збільшуючи значення αw і зменшуючи значення αv) або виконуваних цими вузлами операцій $vt \in V$ (зменшуючи значення αw і збільшуючи значення αv).

За потреби, функцію (1) можна реорганізувати для урахування важливості перевірки кожного функціонального вузла об'єкта діагностування $wj \in W$ і кожної з типових операцій $vt \in V$

Слід зазначити, що можливі два варіанти упорядкування організації діагностичного експерименту на основі значень $F_{k_g \in K}$ функції приоритету команд $kg \in K$.

Перший варіант – з наданням пріоритету при виборі командам $kg \in K$ з мінімальними значеннями $F_{k_g \in K}$ функції приоритету:

$$F_{k_g \in K} \to \min. \qquad (2)$$

При такому підході для організації діагностичного експерименту на основі значень $F_{k_g \in K}$ першочергово будуть обиратися найпростіші команди $kg \in K$, що перевіряють мінімальну кількість функціональних вузлів $wj \in W$ і

типових елементарних мікрооперацій $vt \in V$ процесора, що спростить уточнення місця виникнення несправності при виявленні помилки виконання команди $kg \in K$.

Другий варіант – з наданням пріоритету при виборі командам $kg \in K$ з максимальними значенями $F_{k_g \in K}$ функції приоритету:

$$F_{k_g \in K} \to \max. \tag{3}$$

При такому підході для організації діагностичного експерименту на основі значень $F_{k_g \in K}$ першочергово будуть обиратися найпотужніші команди $kg \in K$, що задіюють максимальну кількість функціональних вузлів $wj \in W$ і типових елементарних мікрооперацій $vt \in V$ процесора. Це прискорить виявлення факту виникнення несправності при виявленні помилки виконання команди $kg \in K$, але ускладнить уточнення місця виникнення несправності (для локалізації місця виникнення несправності знадобиться більше уточнюючих команд).

Виходячи з попередньо проведеного аналізу і зроблених в попередніх розділах висновків можна сформулювати основне положення щодо призначення функціонального діагностування ЦПССК, а також інші базові положення досліджуваного підходу:

1. Мета – перевірити на контрольних наборах даних здатність виконання діагностованим цифровим процесором команд, що утворюють його систему команд, та коректність спрацьовування всіх його вузлів при виконанні характерних для застосовуваних команд елементарних внутрішніх процесорних операцій (мікрооперацій).

2. Підхід базується на принципах командно-керованого функціонального діагностування.

3. Відповідно до двох попередніх положень, основним інструментом організації діагностичних випробувань є команди $kg \in K$, що використовуються для перевірки правильності виконання вузлами процесора $wj \in W$ типових мікрооперацій $vt \in V$ з відображенням результатів діагностичних перевірок на використовуваних контрольних наборах даних.

4. За потреби розширеної перевірки роботи вузлів процесора застосовуються типові процедури тестування вузлів зазначеного класу (в тому числі методи тестового детермінованого або ймовірнісного діагностування [4-8], методи випадкового пошуку [3] тощо).

5. Відбір команд $kg \in K$ для першочергового застосування при перевірці процесора виконується за значенням вагової функції $F_{k_g \in K}$ та у відповідності із заданим критерієм відбору (максимальне або мінімальне

значення $F_{k_g \in K}$ – формули (2) або (3)). Значення функції $F_{k_g \in K}$ обчислюється для команд $kg \in K$ на основі матриць перевірок Pw і Pv за формулою (1).

Література

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

14. Поморова О. В. Метод представлення знань у багатокомпонентних інтелектуальних системах діагностування мікропроцесорних пристроїв / О. В. Поморова, О. Я. Олар // Радіоелектронні і комп. системи. - 2006. - № 6. - С. 110-114.

15. Метод случайного поиска [Електронний ресурс] / Портал «life-prog.ru». – Режим доступу: https://life-prog.ru/1_6778_metod-sluchaynogo-poiska.html (дата звернення 30.10.2018). – Назва з екрана.

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

17. Глущак Сергій Валеріанович. Метод і засоби тестового діагностування цифрових та мікропроцесорних пристроїв з компонентами, побудованими за КМДН-технологією: Дис. канд. техн. наук: 05.13.05 / Технологічний ун-т Поділля. - Хмельницький, 2002. - 215арк.

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

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

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

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

22. Дрозд А.В. Вероятностный подход к функциональному диагностированию вычислительных устройств для обработки приближенных данных / А.В. Дрозд // Радіоелектроніка і інформатика. – Харків. : ХНУРЕ, 2004. – № 1. – С. 101-102.

Вдосконалення алгоритму ранжування та індексації сайтів

Судома І.В.
Науковий керівник – д.т.н., проф. Мясіщев О.А.
Хмельницький національний університет

Алгоритмом для пошуку та видачі результатів Google є PageRank (метод для виміру важливості сторінки). Google використовує його для того, щоб відкорегувати результати так – більш важливіші сайти піднімуться вверх на сторінці результатів пошуку. Суть виміру важливості сайту визначається наступним чином: якщо сайт А посилається на сайт В, то сайт А вважає що сайт В є важливішим.

Для виявлення недоліків видачі результатів запиту на сторінку користувача, потрібно розглянути як саме побудований PageRank.

Порядок ранжування сайтів в Google працює наступним чином:

1) Знайти всі сайти які відповідають ключовим словам пошуку
2) Відранжувати відповідно до «сторінкових факторів», таких як ключові слова
3) Взяти до уваги текст посилань на сайти
4) Відкорегувати результати даними PageRank

Для отримання необхідних корисних результатів пошуку в мережі та отримання реальних оцінок сайту, PageRank буде модифіковано ще одним коефіцієнтом який відповідатиме за популярність сайту у соціальних мережах та його рекомендування на форумах.

Тобто у формулу ранжування PageRank буде додано змінну s – коефіцієнт який буде рівний 0,15, буде множитись на кількість посилань з форумів та соціальних мереж – $T(Cs)$.

Перейдемо до математичного рейтингу сайту та розглянемо як у простій мережі функціонує алгоритм ранжування.

Наприклад веб-сайт С має більш високий рейтинг, ніж сайт Е, хоча є менше посилань на С, ніж на Е, але одна з посилань на С виходить з важливіших сайтів і, отже, має більш високе значення. Якщо умовно вважати, що веб-користувач, який знаходиться на випадковому сайті, має 85% ймовірність вибору випадкового посилання на поточному сайті, і 15% переходу на будь-який інший сайт, то ймовірності переходу на сторінку Е з інших посилань дорівнює 8,1% часу. (15% ймовірності переходу до довільного сайту відповідає коефіцієнту загасання 85%.) Без загасання веб-користувачі в кінцевому підсумку потрапляють на сайти А, В або С, і всі інші сайти будуть мати PageRank, рівний нулю. При наявності загасання сайт А ефективно пов'язує майже всі посилання на сайти в цій мережі, навіть якщо він не має своїх власних вихідних посилань.

Метод підвищення інформаційної безпеки комп'ютерних мереж

Щерба В.І.
Науковий керівник – к.т.н. доц. Красильников С.Р.
Хмельницький національний університет

Інформаційна безпека - стан захищеності життєво важливих інтересів людини, суспільства і держави, при якому запобігається нанесення шкоди через: неповноту, невчасність та невірогідність інформації, що використовується; негативний інформаційний вплив; негативні наслідки застосування інформаційних технологій; несанкціоноване розповсюдження, використання і порушення цілісності, конфіденційності та доступності інформації. Знання в цій сфері дозволять студенту зрозуміти природу інформації та її властивостей, усвідомити сутність інформаційної небезпеки і шляхів її запобігання та усунення [1].

Захищеність комп'ютерної мережі визначається як ступінь адекватності реалізованих в ній механізмів захисту інформації (такі, як ідентифікація і аутентифікація, управління доступом, протоколювання і аудит, криптографія, екранування) існуючим в даному середовищі функціонування ризиків, пов'язаних із здійсненням загроз безпеки інформації, тобто здатність механізмів захисту забезпечити конфіденційність, цілісність і доступність інформації. Захищеність може надавати і часто надає вирішальний вплив на показники ефективності функціонування комп'ютерних мереж. Під загрозою розуміється сукупність умов і факторів, що визначають потенційну або реально існуючу небезпеку виникнення інциденту, який може привести до нанесення збитку функціонуванню комп'ютерної мережі. Загрози можуть класифікуватися за різними ознаками. Зокрема, за характером походження загрози поділяються на дві групи: навмисні і природні. Основними умисними небезпеками вважаються: підключення порушника до каналів зв'язку; несанкціонований доступ; розкрадання носіїв інформації.

Необхідність аналізу захищеності комп'ютерних мереж на етапах проектування і експлуатації диктується також неприпустимо низьким рівнем ефективності існуючих засобів забезпечення інформаційної безпеки. Так, наприклад, по статистичним даним Національного відділу ФБР США по комп'ютерним злочинам, «величина ймовірності запобігання несанкціонованого проникнення в інформаційні системи складає в середньому близько 0,12» [18]. В той час «в багатьох прикладних областях, де забезпеченню безпеки процесів і об'єктів приділяють серйозну увагу, норми безпеки, викладені у відповідних документах. Мають порядок 0,9» [2].

На основі вище сказаного актуальною є розробка підходу, який об'єднає множину моделей і методику, для реалізації детального аналізу захищеності комп'ютерних мереж на етапах експлуатації і проектування, який базується на імітації дій порушника, побудові і аналізу графу загроз.

Досягнення даної мети гостро необхідно, так як використання засобів автоматизації детального аналізу захищеності буде сприяти створенню і експлуатації більш безпечних комп'ютерних мереж.

При постановці завдання дослідження скористаємося представленням системи аналізу захищеності (рис. 1).

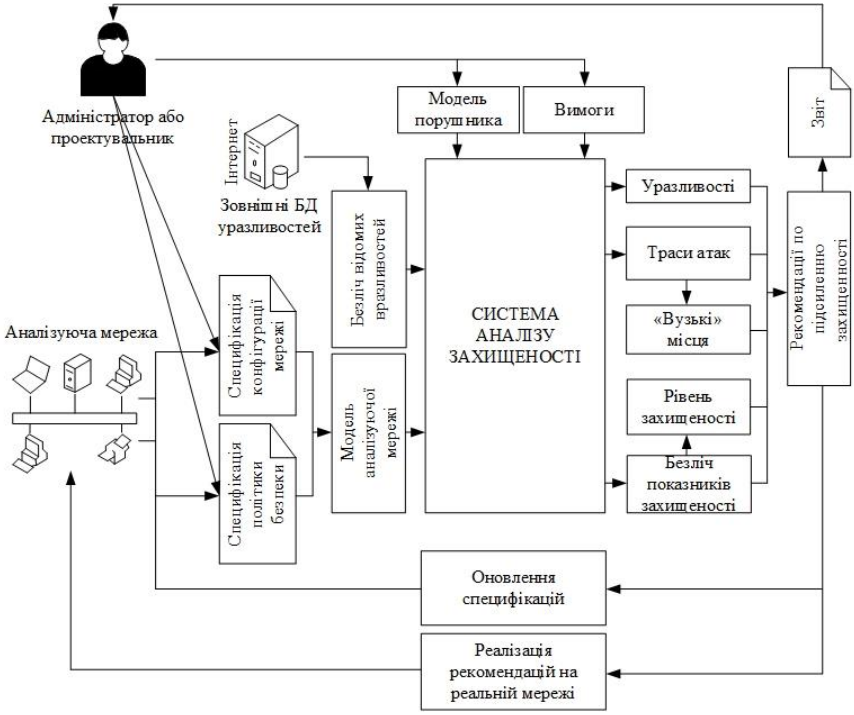

Рисунок 1 – Представлення системи аналізу захищеності

САЗ повинна реалізовувати аналіз захищеності комп'ютерних мереж на етапі проектування і експлуатації. Для задоволення цієї вимоги передбачається використати підхід, при якому аналізується модель комп'ютерної мережі. Ця модель будується на базі специфікацій, що описують конфігурацію мережі(топологію, склад ПЗ і АЗ) і політику безпеки, що реалізовується в ній.

Специфікації описуються на спеціалізованих мовах, ґрунтованих на XML. На етапі проектування комп'ютерної мережі специфікації формуються проектувальником, на етапі експлуатації – в автоматичному режимі за допомогою програмних агентів, що функціонують на хостах.

Під час роботи САЗ повинна формувати сценарії (траси) комп'ютерних

атак, враховувати модель порушника (первинне положення, рівень знань і умінь, первинні знання про аналізовану мережу), робити розрахунок множини показників, що характеризують захищеність комп'ютерної мережі в цілому і її окремих компонентів, враховувати топологію аналізованої мережі, склад програмного і апаратного забезпечення, політику безпеки, що реалізовується. Результатами роботи САЗ являються множина виявлених вразливостей, траси атак, «вузькі» місця в захищеності комп'ютерної мережі (найбільш критичні компоненти комп'ютерної мережі, вірогідність атаки яких найвища), множина показників захищеності, рекомендації по підвищенню рівня захищеності аналізованої мережі. Отримані результати гарантують вироблення обґрунтованих рекомендацій по усуненню виявлених «вузьких» місць і посиленню захищеності комп'ютерної мережі в цілому.

На змістовному рівні наукове завдання даного дослідження можна сформулювати таким чином: розробити методику аналізу захищеності комп'ютерних мереж на етапах проектування і експлуатації, що базується на побудові дерева атак і розрахунку множини показників, що характеризують рівень захищеності комп'ютерної мережі в цілому і окремих її компонентів. Реалізація цієї методики системами аналізу захищеності повинна дозволяти не лише оцінювати рівень захищеності мережі, але і досягати його необхідного значення шляхом зміни конфігурації аналізованої мережі і політики безпеки, що реалізовується в ній.

Для реалізації аналізу захищеності комп'ютерних мереж на етапах проектування і експлуатації необхідно розробити моделі комп'ютерних атак і порушника, аналізованої комп'ютерної мережі, побудови дерева атак і оцінки рівня захищеності.

Позначимо *G* – множина рекомендацій, що формуються системою аналізу захищеності і спрямованих на підвищення захищеності аналізованої мережі Network (наприклад, «Встановити пакет програмних корекцій Windows XP Service Pack 2»), NetworkG – початкова комп'ютерна мережа з реалізованою в ній множиною рекомендацій G, *SecurityLevel* (Network) – функція, результатом якої є рівень захищеності мережі Network [3].

Тоді цільовою функцією методики аналізу захищеності є підвищення загального рівня захищеності комп'ютерної мережі *SecurityLevel* (NetworkG)→max (в окремому випадку цільова функція може бути задана у вигляді *SecurityLevel* (NetworkG)→ *SLПОРТ*, *SLПОРТ*– потрібний рівень захищеності), при дотриманні вимог до інших властивостей САЗ:

1. до своєчасності $P_{СВ}\left(t \le T^{ДОП}\right) \ge P_{СВ}^{ДОП}$, де $P_{РЕС}^{ДОП} = 0{,}99$ і допустимому часі проведення аналізу $T_{ПР}^{ДОП} = T_{ПР}$, де при аналізі мереж розміром до 40 хостів на етапі проектування *ТПР*=45хв. і на етапі експлуатації *ТЕК*=25хв. ($T_{ПР}^{ДОП} \lhd T_{ПР}$,оскільки на етапі експлуатації можлива часткова автоматизація підготовчого етапу методики);

2. до обґрунтованості: $N_C \geq \max_{s \in S} N_C^S$, $N_Y \geq \max_{s \in S} N_Y^S$, $N_П \geq \max_{s \in S} N_П^S$ $N_Y \geq \max_{s \in S} N_Y^S, N_П \geq \max_{s \in S} N_П^S$, де NC, NУ, NП - кількість аналізованих сценаріїв атак, число виявлених вразливостей і число параметрів, що враховуються, розробленим прототипом САЗ, S – множина існуючих САЗ, N_C^S, $N_У^S$ і $N_П^S$ –кількість аналізованих сценаріїв атак, виявлених вразливостей і параметрів, що враховуються, існуючою САЗ s відповідно. Множина параметрів, що враховуються, складається з наступних елементів:(а) облік конфігурації аналізованої мережі (хости, концентратори, різні ОС, мережеві сервіси, топологія); (б) облік політики безпеки(правил фільтрації, аутентифікації, авторизації); (в) облік параметрів порушника (первинне місце розташування, первинні знання, рівень знань і умінь); (г) загальні параметри(оновлення БД вразливостей, сценаріїв тих, що атакують дії, можливість завдання множини аналізованих хостів); до ресурсоспоживання:, де $P_{РЕС}\left(r \leq R_{ДОП}\right) \geq P_{РЕС}^{ДОП}$, де $P_{РЕС}^{ДОП} = 0{,}99$, $R_{ДОП} = 0{,}15$ (15% від загального ресурсу, доступного для вирішення призначених для користувача завдань) для наступних ресурсів: оперативна пам'ять, жорсткий магнітний диск, мережевий інтерфейс. На час побудови і аналізу дерева атак допускається максимальна завантаженість центрального процесора спеціально виділеного для виконання процесу аналізу захищеності комп'ютера

Література

1. Астахов, А. Анализ защищенности корпоративных автоматизированных систем [Электронный ресурс] / А. Астахов. – Электрон, текстовые дан. и граф. дан. – [Б. м.: б. и.]. – Режим доступа: http: // www.jetinfo.ru / 2002 / 7 / 1 / article 1.7.2002.html (по состоянию на 01.01.2017).

2. Котенко, И. В. Интеллектуальная система анализа защищенности компьютерных сетей на различных этапах жизненного цикла [Текст] / И. В. Котенко, М. В. Степашкин // Труды Международных научно-технических конференций «Интеллектуальные системы (AIS–05)» и «Интеллектуальные САПР (CAD-2005)». – М.: Физматлит, 2005. –Т. 1. –С. 231-237.

3. Основы информационной безопасности. Вводный курс по специальности 075500 [Электронный ресурс] / Электрон, текстовые дан. и граф. дан. – [Б. м.: б. и.]. – Режим доступа: http: // intsys.msu.ru / staff / vnosov / infosecurity.ppt (по состоянию на 01.01.2017).

Наукове видання

«Інтелектуальний потенціал – 2018» - збірник наукових праць молодих науковців і студентів з нагоди 30-річчя підготовки ІТ- фахівців в ХНУ/ Колектив авторів – Хмельницький: ПВНЗ УЕП, 2018. – Ч.3: Кібербезпека та актуальні проблеми комп'ютерних систем і мереж – 108 с.

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

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

Здано до складання 5.11.18. Підписано до друку 10.11.18. Формат 60х84/16. Папір друкарський. Тираж 50 прим. Умовних друкованих аркушів – 6,75. Обліково-видавничих аркушів –1,5.

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

ББК 74.480.278
С.88