МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

кафедра

інформаційних систем

та мереж

Контрольна розрахункова робота

з дисципліни

«Технології захисту інформації»

Варіант 24

Виконала

студентка

групи ІТ-31

*Щербак Л.В.*

Прийняв

*доцент кафедри ІСМ*

*Басюк Т.М.*

Львів-2022

**ЗМІСТ**

[ВСТУП](#_vbbxs0970zlm) **3**

[РОЗДІЛ 1. ТЕОРЕТИЧНІ ПОЛОЖЕННЯ ЗГІДНО ІЗ ТЕМОЮ](#_wlblom2g925t) **4**

[РОЗДІЛ 2. ПОРІВНЯЛЬНИЙ АНАЛІЗ ТА ФОРМУВАННЯ ВИМОГ](#_z1k3bt9m9v4y) **12**

[**РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЇ**](#_be00xtlk5vxr) **22**

[ВИСНОВКИ](#_qdtl2mudolhb) **26**

[**СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ**](#_iqhwgd6rgf0p) **27**

# **ВСТУП**

Постійний розвиток і широке використання Інтернету приносить користь багатьом користувачам мережі з багатьох аспектів. Тим часом безпека мережі стає набагато важливішою з широким використанням мережі. Безпека мережі тісно пов’язана з комп’ютерами, мережами, програмами, різними даними тощо, де метою захисту є запобігання неавторизованому доступу та модифікації. Проте зростаюча кількість підключених до Інтернету систем у фінансах, електронній комерції та військових сферах робить їх об’єктами мережевих атак, що призводить до великого ризику та шкоди. По суті, необхідно забезпечити ефективні стратегії для виявлення та захисту атак і підтримки безпеки мережі. Крім того, різні типи атак зазвичай потрібно обробляти різними способами. Таким чином, як ідентифікувати різні типи мережевих атак, це стає головним завданням у сфері мережевої безпеки, яке необхідно вирішити, особливо тих атак, яких раніше не було. Сьогодні існує велика потреба у забезпеченні шифрування інформації в режимі реального часу, але існуючі стандарти алгоритмів шифрування є дуже повільними для вирішення цієї задачі. Тому постає гостра необхідність у розробці алгоритмів шифрування, які могли це здійснювати на порядок швидше. В даній роботі я буду досліджувати системи виявлення атак на комп'ютерну мережу та шифрування/розшифрування в режимі AES. Ми живемо в глобальному інформаційному просторі, який забезпечує ефективну взаємодію людей, їх доступ до світових інформаційних ресурсів, продуктів і послуг. У зв’язку з цим виникає проблема захисту інформації, що передається і обробляється електронними засобами, яку вирішують за допомогою криптографічних методів.

# **РОЗДІЛ 1. ТЕОРЕТИЧНІ ПОЛОЖЕННЯ ЗГІДНО ІЗ ТЕМОЮ**

Тема: “Системи виявлення атак на комп'ютерну мережу”

Виявлення атак – це процес ідентифікації та регулювання на підозрілу діяльність, направлену на обчислювальні чи мережні ресурси.

Система виявлення атак (вторгнень) (англ. Intrusion Detection System, IDS) — програмний або апаратний засіб, призначений для виявлення фактів несанкціонованого доступу в комп'ютерну систему або мережу або несанкціонованого управління ними в основному через Інтернет. Про будь-яку активність шкідливого ПЗ або про порушення типової роботи централізовано збирається інформація SIEM-системою (англ. Security information and event management). SIEM-система обробляє дані отримані від багатьох джерел і використовує методи фільтрування тривог для розрізнення несанкціонованої активності від хибного спрацювання тривоги. Про що оповіщається або адміністратор або операційний центр безпеки.

Деякі системи виявлення вторгнень можуть виявити початок атаки на мережу, причому деякі з них здатні виявляти раніше не відомі атаки. Такі системи називають системами запобігання вторгненням (англ. Intrusion Prevention System, IPS). IPS не обмежуються лише оповіщенням, але й здійснюють різні заходи, спрямовані на блокування атаки (наприклад, розрив з'єднання або виконання скрипту, заданого адміністратором). На практиці досить часто програмно-апаратні рішення поєднують у собі функціональність двох типів систем. Їх об'єднання називають IDPS (IDS i IPS).

Перш ніж спрямувати погляд на перспективні технології виявлення та відображення мережевих атак, озирнемося назад і простежимо еволюцію систем захисту інформації.

70-ті роки. ПК в організаціях дуже рідкісні, а ті, що є, не об'єднані в єдину обчислювальну мережу. Питання забезпечення інформаційної безпеки вирішуються з допомогою вбудованих захисних механізмів операційних систем.

Початок 80-х. З'являються локальні обчислювальні мережі, які об'єднують кілька комп'ютерів. Загроза атак невелика, і адміністратори вручну перевіряють реєстраційні журнали системи захисту щодо виявлення “слідів”, залишених зловмисниками.

Середина 80-х – початок 90-х років. Масштаб мереж поступово зростає. Вже не рідкість, коли мережа підприємства об'єднує сотні та тисячі комп'ютерів, що знаходяться у різних, у тому числі й географічно віддалених точках. Адміністратор мережі вже не здатний вручну перевіряти всі журнали реєстрації - для цього створюються засоби автоматичної перевірки журналів (automated log checkers). Прикладом такого засобу можна назвати систему LogEvent Manager, розроблену фірмою March Information Systems, яку згодом поглинула компанія Internet Security Systems. Подібні програми завантажують журнали реєстрації віддалених комп'ютерів на робоче місце адміністратора безпеки, де здійснюється їх аналіз. Хоча адміністратору вже не треба вручну звертатися до кожного комп'ютера для аналізу його журналу реєстрації, залишається проблема, яка доводиться обробляти величезні обсяги даних.

Для вирішення цього завдання були розроблені засоби для автоматичного виявлення журналів реєстрації подій, що вказують на порушення політики безпеки. Ці засоби отримали назву систем виявлення атак на рівні хоста (Host-based Intrusion Detection Systems, IDS) або моніторів реєстраційних файлів (Log-File Monitors).

За останні 30 років способи виявити дії зловмисника суттєво вдосконалилися. Якщо раніше хакерів ловили через завербованих інформаторів, то зараз пошук слідів компрометації здійснюється засобами інтелектуального виявлення та автоматизованого аналізу. За часів перших шкідливих кампаній виявити шкідливе програмне забезпечення допомагали антивірусні рішення, які використовували сигнатури (зразки вірусного коду). Потім у засобах антивірусного захисту з'являються інтелектуальніші технології – евристичний спосіб виявлення та поведінковий аналіз. На прикладі розвитку антивірусних засобів, те саме можна сказати і про інші способи виявлення комп'ютерних атак, заснованих на аномаліях в мережевому трафіку і в цифрових подіях різних ІТ-систем. З появою нових технологій обробки та використання даних, а також створення засобів автоматизації бізнес-процесів з'являються і нові загрози. Розвиток технік та тактик атакуючих, поява нових загроз та вразливостей є драйвером розвитку технологій виявлення комп'ютерних атак, що застосовуються на різних рівнях інформаційних інфраструктур: від фізичного до прикладного. Але водночас свідчить, що технології з'являються постфактум комп'ютерних атак.

Для того, щоб надати огляд ефективного виявлення атак на основі методів глибокого навчання, важливо представити базові знання.

Розробка процесу виявлення атак

Атаки можна розпізнати як спроби обійти політику безпеки системи, що дає зловмисникам легший доступ для отримання або зміни інформації, навіть руйнуючи систему. З розвитком технологій у системах бездротового зв’язку з’явилися серйозні загрози безпеці мережі, особливо безпеці систем бездротового зв’язку, через більш часті дії мережевих атак через характеристики відкритості бездротових каналів. Оскільки ми живемо в епоху машинного навчання та великих даних, кібербезпека в системах бездротового зв’язку важлива для користувачів, щоб захистити мережу, комп’ютер і дані від атак. Існують різні типи атак для кіберсистем, такі як флуд, розподілена відмова в обслуговуванні, аномальна пакетна атака та спуфінг.

Для боротьби з такими потоками атак на кібербезпеку дослідники запропонували багато рішень. Серед рішень виявлення атак є одним із найефективніших способів, який пропонує повний і динамічний механізм безпеки для моніторингу, запобігання та протидії атакам. Зокрема, виявлення атак збиратиме інформацію шляхом моніторингу мережі, стану системи, поведінки та використання системи, що може автоматично виявляти неавторизоване використання системними користувачами та атаки зовнішніх зловмисників на систему.

В останні роки машинне навчання розвивається з неймовірною швидкістю. Серед різних методів машинного навчання структури глибокого навчання створюють штучні нейронні мережі для імітації взаємозв’язків нейронів людського мозку, що дає особливу силу для вирішення складних проблем. Таким чином, дослідники використовують різні методи глибокого навчання для виявлення атак, що призводить до значних досягнень. Однак через обмеженість методів глибокого навчання залишається багато невирішених проблем. Важливо зробити короткий опис того, як попередні методи використовують методи глибокого навчання для виявлення атак, що може принести нові ідеї для майбутніх розробок.

Усі існуючі технології виявлення мережевих атак можна розділити на два типи:

* методи на основі сигнатур (зразків і правил);
* методи на основі аномалій.

Класифiкацiї систем виявлення атак

За способом виявлення атаки:

* Виявлення аномального поводження (anomaly-based)
* Виявлення зловживань (misuse detection або signature-based)

За способом збору iнформацiї про атаку

За способом реагування:

* Пасивнi просто фіксують факт атаки, записують дані у файл журналу й видають попередження
* Активнi намагаються протистояти атаці

Аналіз активності атак

* Статичні - роблять «знімки» (snapshot) середовища та здійснюють їх аналіз, розшукуючи вразливе програмне забезпечення, помилки в конфігураціях і т. д. Виявляють сліди вторгнення.
* Динамічні - здійснюють моніторинг у реальному часі всіх дій, що відбуваються в системі, переглядаючи файли аудиту або мережні пакети, що передаються за певний проміжок часу. Постійно стежать за безпекою системи
* Мережеві - здійснюють контроль усього трафіку даних всієї підмережі та порівнюють трафік, який передається у підмережі з бібліотекою відомих атак. Як тільки розпізнана атака або визначено відхилення у поведінці, відразу відсилається попередження адміністратору
* Хостові - встановлюються на хості і виявляють зловмисні дії на ньому

Безпека мережі — заходи, які захищають інформаційну мережу від несанкціонованого доступу, випадкового або навмисного втручання в роботу мережі або спроб руйнування її компонентів.

Безпека інформаційної мережі включає захист:

* обладнання
* програмного забезпечення
* даних
* персоналу

Мережева безпека охоплює різні комп'ютерні мережі :

* приватні
* державні

Найбільш поширений і простий спосіб захисту мережевих ресурсів є присвоєння їм унікального імені та відповідного паролю.

Система безпеки мережі:

* Захищає від внутрішніх та зовнішніх мережних атак.
* Забезпечує конфіденційність обміну інформацією з будь-якого місця та в будь-який час.
* Контролює доступ до інформації, ідентифікуючи користувачів та їхні системи.
* Забезпечує надійність системи.

Архітектури систем захисту

Існують дві типові архітектури таких систем: автономна система та агент – менеджер. Перша надто проста і нині у комерційних продуктах практично не зустрічається. Програми, створені з її основі, потрібно встановлювати кожен контрольований комп'ютер. Зв'язок між автономними системами, встановленими на різних комп'ютерах, відсутній, і журнали реєстрації адміністратору доводиться завантажувати на свій комп'ютер за допомогою інших засобів або аналізувати ці журнали, здійснюючи віддалений вхід на кожен комп'ютер. Вочевидь, що у великих, територіально розподілених мережах це неможливо.

Для сучасних мереж ефективніша архітектура агент - менеджер, оскільки інформація про події, реєстрованих агентами кожному контрольованому комп'ютері, передається на центральну консоль (менеджер). З цієї консолі проводиться управління віддаленими агентами та їх налаштування.

До переваг систем виявлення атак, що аналізують журнали реєстрації, можна віднести:

* Можливість виявлення атаки на конкретний вузол;
* Можливість оцінити підсумки атаки. Оскільки IDS працює з подіями, які вже зафіксовані в журналі реєстрації, можна з упевненістю стверджувати, була відображена атака чи ні;
* Ефективність застосування в комутованих мережах та мережах з канальним шифруванням.

Однак такі системи мають свої недоліки. Вони не можуть виявити атаки, спрямовані на вузли, де не ведуться журнали реєстрації, або на платформи, для яких не існує відповідного програмного забезпечення. Крім того, їх не можна використовувати для раннього виявлення атак.

Для вирішення цих проблем були розроблені системи виявлення атак на рівні мережі (Network-based IDS): вони аналізують мережевий трафік у реальному масштабі часу та порівнюють його із заздалегідь визначеними сигнатурами атак. У мережевих IDS є дві основні переваги. По-перше, один агент такої системи може переглядати цілий сегмент мережі з великою кількістю вузлів. Можуть бути виявлені атаки проти будь-яких елементів мережі підприємства, починаючи від маршрутизаторів та міжмережевих екранів та закінчуючи Web-серверами та серверами баз даних. По-друге, мережеві IDS здатні ідентифікувати атаки у реальному часі, завдяки чому можуть бути оперативно вжиті заходи протидії.

Система виявлення вторгнень (IDS) — це програма або пристрій, який відстежує вхідний і вихідний мережевий трафік, постійно аналізуючи дії на наявність змін у шаблонах і сповіщаючи адміністратора, коли виявляє незвичну поведінку. Потім адміністратор переглядає тривоги та вживає заходів для усунення загрози.

Наприклад, IDS може перевіряти дані, що передаються мережевим трафіком, щоб перевірити, чи містять вони відоме шкідливе програмне забезпечення або інший шкідливий вміст. Якщо він виявляє цей тип загрози, він надсилає сповіщення вашій команді безпеки, щоб вони могли дослідити та усунути її. Після того, як ваша команда отримає сповіщення, вона повинна діяти швидко, щоб запобігти нападу на систему.

Щоб переконатися, що IDS не сповільнює продуктивність мережі, ці рішення часто використовують аналізатор комутованого порту (SPAN) або порт тестового доступу (TAP) для аналізу копії внутрішнього трафіку даних. Однак вони не блокують загрози, щойно вони потрапляють у мережу, як це роблять системи запобігання вторгненням.

Незалежно від того, налаштовано ви фізичний пристрій чи програму IDS, система може:

* Розпізнавати шаблони атак у мережевих пакетах.
* Відстежувати поведінку користувачів.
* Визначати аномальну активність трафіку.
* Переконайтися, що діяльність користувача та системи не суперечить політикам безпеки.

Інформація від системи виявлення вторгнень також може допомогти групі безпеки:

* Аудит мережі на наявність вразливостей і поганих конфігурацій.
* Оцінка цілісності критичних систем і файлів.
* Створити ефективні засоби контролю та реагування на інциденти.
* Проаналізувати кількість і типи кіберзагроз, що атакують мережу.

Крім переваг кібербезпеки, IDS також допомагає досягти відповідності вимогам регулярності. Краща видимість мережі та краще ведення журналів гарантують, що мережеві операції відповідають усім відповідним нормам.

Назвемо інші переваги таких систем.

Зловмисник не може приховати слідів своєї діяльності. При виявленні атак у реальному часі IDS мережного рівня використовують поточний мережевий трафік, тому хакер не може видалити сліди своєї присутності. Аналізовані дані містять не лише інформацію про метод атаки, але й відомості, які можуть допомогти при ідентифікації зловмисника та фігурувати як доказ у суді. Однак, оскільки багато хакерів добре знайомі з журналами реєстрації, вони знають, як маніпулювати цими файлами для приховування слідів своєї діяльності, щоб знизити ефективність систем рівня ОС, які користуються цією інформацією виявлення атаки.

Системи можуть виявляти невдалі атаки чи підозрілу діяльність. IDS мережного рівня, встановлена ​​із зовнішнього боку міжмережевого екрана (МСЕ), здатна фіксувати атаки, націлені на ресурси, що захищаються, навіть якщо екран відобразить ці спроби. Системи захисту, встановлені на вузлах усередині мережевого периметра, не бачать відбитих атак, які не досягають вузлів, захищених МСЕ. Тим часом, ця втрачена інформація може мати дуже важливе значення при оцінці та вдосконаленні політики безпеки.

Цілі систем виявлення вторгнень

Брандмауер сам по собі не забезпечує належного захисту від сучасних кіберзагроз. Зловмисне програмне забезпечення та інший зловмисний вміст часто доставляється за допомогою законних типів трафіку, наприклад електронної пошти чи веб-трафіку. IDS надає можливість перевіряти вміст цих повідомлень і ідентифікувати будь-які шкідливі програми, які вони можуть містити.

Основною метою IDS є виявлення аномалій до того, як хакери досягнуть своєї мети. Коли система виявляє загрозу, IDS інформує ІТ-персонал і надає таку інформацію про небезпеку:

* Адреса джерела вторгнення.
* Адреси цілей і жертв.
* Тип загрози.

Другою метою системи виявлення вторгнень є спостереження за зловмисниками та ідентифікація:

* До яких ресурсів намагаються отримати доступ зловмисники.
* Як хакери намагаються обійти контроль безпеки.
* Які типи кібератак ініціюють зловмисники.

Операційний центр безпеки компанії (SOC) і аналітики можуть використовувати цю інформацію для покращення стратегії безпеки мережі.

Виявлення аномалій і звітування є двома основними функціями системи виявлення вторгнень. Однак деякі системи виявлення можуть реагувати на зловмисну ​​активність, наприклад автоматично блокувати IP-адресу або закривати доступ до конфіденційних файлів. Системи з такими можливостями реагування є системами запобігання вторгненням (IPS).

Як працюють системи виявлення вторгнень?

IDS контролює трафік до та з усіх пристроїв у мережі. Система працює за брандмауером як вторинний фільтр для шкідливих пакетів і в основному шукає дві підозрілі ознаки:

* Сигнати відомих атак.
* Відхилення від регулярної діяльності.

Система виявлення вторгнень зазвичай покладається на кореляцію шаблонів для виявлення загроз. Цей метод дозволяє IDS порівнювати мережеві пакети з базою даних із сигнатурами відомих кібератак. Найпоширенішими атаками, які IDS може позначити за допомогою кореляції шаблонів, є:

* Шкідливе програмне забезпечення (хробаки, програми-вимагачі, трояни, віруси, боти тощо).
* Атаки сканування, які надсилають пакети в мережу для збору інформації про відкриті чи закриті порти, типи дозволеного трафіку, активні хости та версії програмного забезпечення.
* Асиметрична маршрутизація, яка надсилає зловмисний пакет і обходить контроль безпеки за допомогою різних маршрутів входу та виходу.
* Атаки переповнення буфера, які замінюють вміст бази даних шкідливими виконуваними файлами.
* Атаки на певний протокол (ICMP, TCP, ARP тощо).
* Порушення трафіку, що перевантажує мережу, наприклад DDoS-атака.

Як тільки IDS виявляє аномалію, система позначає проблему та подає сигнал тривоги. Сповіщення може варіюватися від простого запису в журналі аудиту до термінового повідомлення ІТ-адміністратору. Потім команда усуває проблему та визначає першопричину проблеми.

Які є типи систем виявлення вторгнень?

Існує два основних типи IDS залежно від того, де їх налаштовує команда безпеки:

* Система виявлення вторгнень у мережу (NIDS).
* Система виявлення вторгнень на хост (HIDS).

Те, як система виявлення вторгнень виявляє підозрілу активність, також дозволяє нам визначити дві категорії:

* Система виявлення вторгнень на основі сигнатур (SIDS).
* Система виявлення вторгнень на основі аномалій (AIDS).

Залежно від варіанту використання та бюджету ви можете розгорнути NIDS або HIDS або покладатися на обидва основні типи IDS. Те саме стосується моделей виявлення, оскільки багато команд створили гібридну систему з можливостями SIDS та СНІДу.

Проте системи виявлення мережевих атак лише на рівні мережі мають свої недоліки.

По-перше, їх застосування у високошвидкісних мережах пов'язані з певними проблемами. Сучасні комерційні системи захисту, незважаючи на декларовану можливість роботи з мережами Fast Ethernet (100 Мбіт/с), не можуть ефективно працювати в мережах із пропускною здатністю вище 80 Мбіт/с, отже, при швидкості передачі інформації понад 100 Мбіт/с (наприклад, для каналу ATM OC-3) вони не застосовуються.

По-друге, мережеві IDS неефективно працюють у комутованих мережах та мережах з канальним шифруванням.

Зазначимо, що IDS мережного та системного рівнів зі своїми перевагами та перевагами ефективно доповнюють один одного та взаємоусувають недоліки.

Завершуючи короткий опис наявних рішень у сфері виявлення атак, спробуємо відповісти питанням: чи підійдуть ці рішення у тому сучасному вигляді захисту мереж у недалекому майбутньому?

Щодо цього є серйозні сумніви. Мережі стали настільки складними, що важко контролювати існуючими методами. Число вузлів зростає з небувалою швидкістю. З появою гігабітних швидкостей і комутованих віртуальних приватних мереж обсяг трафіку, що передається по мережах, зріс на кілька порядків. Виникає необхідність у нових підходах до виявлення атак, що дозволяють впоратися з цими проблемами.

# **РОЗДІЛ 2. ПОРІВНЯЛЬНИЙ АНАЛІЗ ТА ФОРМУВАННЯ ВИМОГ**

Тема: “Розробка системи шифрування/розшифрування в режимі AES”

Advanced Encryption Standard (AES), також відомий під назвою Rijndael — симетричний алгоритм блочного шифрування (розмір блока 128 біт, ключ 128/192/256 біт), фіналіст конкурсу AES і прийнятий як американський стандарт шифрування урядом США. Вибір припав на AES з розрахуванням на широке використання та активний аналіз алгоритму, як це було із його попередником, DES. Державний інститут стандартів і технологій (англ. National Institute of Standards and Technology, NIST) США опублікував попередню специфікацію AES 26 жовтня 2001 року, після п'ятилітньої підготовки. 26 травня 2002 року AES оголошено стандартом шифрування. Станом на 2009 рік AES є одним із найпоширеніших алгоритмів симетричного шифрування.

Потреба у новому стандарті шифрування постала у середині 1990-х років. Наявний тоді стандарт DES, довжиною ключа 56 біт, давав змогу застосувати метод грубої сили для дешифрування даних. Успішні злами даних відбулись уже наприкінці 1990-х. Крім того, архітектура DES орієнтувалась на апаратну реалізацію, а програмна реалізація на платформах з обмеженими ресурсами не давала необхідної швидкості застосування. Модифікація DES 3-DES мала достатню довжину ключа, але при цьому була ще повільнішою.

12 жовтня 1997 р. NIST оголосив конкурс на обрання спадкоємця для DES, що був американським стандартом ще з 1977 року. Перед претендентами поставили такі основні вимоги

* блочний шифр
* довжина блоку 128 біт
* ключі довжиною 128, 192 і 256 біт.

Вибір алгоритму проходив у три етапи. 20 серпня 1998 року на 1-й конференції AES було оголошено список з 15 кандидатів. У серпні 1999 року на 2-й конференції AES список скоротився до п'яти фіналістів: MARS, RC6, Rijndael, Serpent і Twofish. За результатами доповідей 3-ї конференції, що проходила у Нью-Йорку 13—14 квітня 2000 року, 2 жовтня 2000 алгоритм, запропонований бельгійськими криптографами Д. Деймоном та В. Ріджменом, був оголошений переможцем конкурсу і почалась процедура стандартизації. 26 травня 2002 року AES був прийнятий як стандарт.

Для того, щоб алгоритм AES став гідною заміною DES, його архітектура повинна задовольняти декільком критеріям: високий ступінь захисту, проста структура і висока продуктивність. Очевидно, що найвищим пріоритетом для алгоритму AES є захист. Уже на рівні внутрішньої архітектури він повинен мати надійність, достатню для того, щоб протистояти майбутнім спробам його злому. Разом з тим структура алгоритму, на відміну від традиційних поглядів, повинна бути настільки простою, щоб гарантувати ефективну процедуру шифрування. Наступною вимогою, запропонованою до AES, є висока продуктивність. Широке поширення алгоритму на ринку потребує прийнятної продуктивності при роботі на самих різних платформах від смарт-карт до великих серверів. Відмінна продуктивність алгоритму вимагає високу швидкість роботи при шифруванні і дешифруванні, а також при реалізації графіка ключа.

Існує кілька моделей атак на цей алгоритм.

Атака за часом.

Ця атака ґрунтується на припущенні, що операції у пристроях виконуються за різний час залежно від вхідних даних. Таким чином, маючи набір точних вимірювань часу для різних операцій, криптоаналітик може відновити вихідні дані.

Розглянута атака неможлива на алгоритми, операції яких виконуються за однакове число тактів на всіх платформах (бітові операції над фіксованим числом біт), але так як алгоритм AES використовують операції додавання та множення, що не задовольняють цю вимогу, він піддається атаці за часом.

Можливі методи протидії такому виду атаки:

Зробити час виконання всіх операцій однаковим, після чого атака за часом стає неможливою. Слід зазначити, що цей метод реалізувати досить складно, оскільки оптимізації компілятора, звернення до кешу та інші фактори можуть вносити різні відхилення.

Додавати до часу виконання операції затримки випадкової тривалості, це значно ускладнить проведення атаки в такий спосіб.

Атака з енергоспоживання.

Ця атака ґрунтується на вимірі енергоспоживання пристрою з високою точністю в процесі роботи шифратора. Таким чином, криптоаналітик може отримати інформацію про операції, що виконуються у пристрої і у нього з'являється можливість витягти криптографічні ключі або іншу секретну інформацію з пристрою, не безпосередньо впливаючи на нього.

Можливими методами протидії такому виду атаки є:

Балансування енергоспоживання — під час операції задіяти всі апаратні частини пристрою.

Забезпечення незалежності флуктуацій енергоспоживання та процесів, що відбуваються в системі.

AES відноситься до SP-мереж і ґрунтується на новій архітектурі квадрата (square), для якої характерно:

* уявлення блоку, що шифрується, у вигляді двовимірного байтового масиву;
* шифрування за один раунд всього блоку даних;
* виконання криптографічних перетворень, як над окремими байтами масиву, так і над його рядками і стовпцями, що забезпечує дифузію даних одночасно в двох напрямках - по рядках і по стовпцях.

Відомі напади

Для криптографів криптографічний «злом» — це щось швидше, ніж атака грубою силою, тобто виконання одного пробного дешифрування для кожного можливого ключа в послідовності. Таким чином, перерва може включати результати, які неможливо досягти за допомогою сучасних технологій. Незважаючи на те, що це непрактично, теоретичні розриви іноді можуть дати розуміння моделей вразливості. Найбільша успішна публічно відома атака грубою силою проти широко застосованого алгоритму шифрування блокового шифру була проти 64-розрядного ключа RC5 distributed.net у 2006 році.

Ключовий простір збільшується в 2 рази для кожного додаткового біта довжини ключа, і якщо кожне можливе значення ключа є рівноімовірним, це означає подвоєння середнього часу пошуку ключа грубою силою. Це означає, що зусилля під час пошуку грубою силою експоненціально зростають із довжиною ключа. Довжина ключа сама по собі не означає захисту від атак, оскільки існують шифри з дуже довгими ключами, які виявилися вразливими.

AES має досить просту алгебраїчну структуру. У 2002 році Ніколя Куртуа та Йозеф Пепржик оголосили про теоретичну атаку під назвою «XSL-атака», яка нібито показала слабкість алгоритму AES, частково через низьку складність його нелінійних компонентів. З тих пір інші документи показали, що атака, як це було спочатку представлено, непрацездатна; див. атаку XSL на блокові шифри.

Під час процесу вибору AES розробники конкуруючих алгоритмів написали про алгоритм Rijndael: «Ми стурбовані [його] використанням... у критично важливих для безпеки програмах». Однак у жовтні 2000 року, наприкінці процесу відбору AES, Брюс Шнайер, розробник конкуруючого алгоритму Twofish, написав, що хоча він думав, що колись будуть розроблені успішні академічні атаки на Rijndael, він «не вірив, що хтось коли-небудь це зробить». виявити атаку, яка дозволить комусь читати трафік Rijndael».

До травня 2009 року єдиними успішними опублікованими атаками на повний AES були атаки по бічних каналах на деякі конкретні реалізації. У 2009 році було виявлено нову атаку зі зв’язаним ключем, яка використовує простоту розкладу ключів AES і має складність 2119. У грудні 2009 року її було покращено до 299,5. Це продовження атаки, виявленої раніше в 2009 році Алексом Бірюковим, Дмитром Ховратовичем та Івіцею Ніколічем, зі складністю 296 для одного з кожних 235 ключів. Проте атаки з пов’язаними ключами не викликають занепокоєння в будь-якому правильно розробленому криптографічному протоколі, оскільки належним чином розроблений протокол (тобто програмне забезпечення для реалізації) подбає про те, щоб не дозволити пов’язані ключі, по суті, обмежуючи засоби зловмисника щодо вибору ключів для пов’язаності.

Ще одна атака була опублікована Брюсом Шнайєром 30 липня 2009 року та опублікована як препринт 3 серпня 2009 року. Ця нова атака, авторами Алекса Бірюкова, Орра Данкельмана, Натана Келлера, Дмитра Ховратовича та Аді Шаміра, спрямована проти AES-256. який використовує лише два пов’язані ключі та 239 разів для відновлення повного 256-бітного ключа 9-раундової версії, або 245 разів для 10-раундової версії з сильнішим типом атаки пов’язаних підключів, або 270 разів для 11-раундової версії версія. 256-бітний AES використовує 14 раундів, тому ці атаки неефективні проти повного AES.

Практичність цих атак із сильнішими пов’язаними ключами була піддана критиці, наприклад, у статті Вінсента Реймена у 2010 році про атаки на AES-128 із зв’язками обраних ключів у середині.

У листопаді 2009 року перша відома атака з розпізнаванням ключа проти скороченої 8-раундової версії AES-128 була випущена як препринт. Ця атака з розпізнаванням відомого ключа є вдосконаленням відскоку або атаки «початок із середини» проти перестановок, подібних до AES, які розглядають два послідовні раунди перестановки як застосування так званого Super-S-box. . Він працює на 8-раундовій версії AES-128 із складністю часу 248 і пам’яті 232. 128-бітний AES використовує 10 раундів, тому ця атака не ефективна проти повного AES-128.

Перші атаки відновлення ключа на повний AES були розроблені Андрієм Богдановим, Дмитром Ховратовичем і Крістіаном Рехбергером, і вони були опубліковані в 2011 році. Атака є біклічною атакою та є швидшою за грубу силу приблизно в чотири рази. Для відновлення ключа AES-128 потрібно виконати 2126.2 операцій. Для AES-192 і AES-256 потрібно 2190,2 і 2254,6 операцій відповідно. Цей результат було додатково покращено до 2126.0 для AES-128, 2189.9 для AES-192 і 2254.3 для AES-256, що є поточними найкращими результатами в атаці відновлення ключа проти AES.

Це дуже незначна перевага, оскільки 126-бітний ключ (замість 128-біт) все одно потребуватиме мільярдів років для перебору на поточному та передбачуваному обладнанні. Крім того, автори розраховують найкращу атаку за допомогою своєї методики на AES зі 128-бітним ключем, що вимагає зберігання 288 біт даних. Це становить приблизно 38 трильйонів терабайт даних, що більше, ніж усі дані, які зберігалися на всіх комп’ютерах планети у 2016 році.

Таким чином, немає жодних практичних наслідків для безпеки AES. Пізніше просторову складність було покращено до 256 біт, що становить 9007 терабайт (залишаючи часову складність 2126,2).

Згідно з документами Сноудена, АНБ проводить дослідження щодо того, чи може криптографічна атака, заснована на статистиці тау, допомогти зламати AES.

Наразі не існує жодної відомої практичної атаки, яка дозволила б комусь без знання ключа читати дані, зашифровані за допомогою AES, якщо їх правильно застосувати.

За архітектурою квадрата побудовані також такі щифри, як SQUARE, CRYPTON, SERPENT. Загальні характеристики AES:

* Для шифрування і розшифрування використовуються 128-бітові блоки даних.
* Дозволяється використовувати три різних довжини ключа - 128,192 або 256 біт.
* Від розміру ключа залежить число раундів шифрування: довжина 128 біт-10 раундів; довжина 192 біта - 12 раундів; довжина 256 біт -14 раундів.
* Всі раунди, крім останнього, ідентичні.

Переваги алгоритму AES:

1. Широке застосування. Після того, як уряд США затвердив його, як стандарт, він підтримується більшістю постачальників.

2. Вибір з трьох можливих довжин ключів, дозволяє користувачам знайти компроміс між швидкістю та безпекою. Відповідно, збільшення довжини ключа збільшує час виконання шифрування та розшифрування. В той же час, всі три розміри ключа вважаються безпечними і найбільш відомі атаки на AES зменшують ефективний розмір ключа в найкращому раз на 3 біти.

3. AES використовує лише один S-box для всіх байтів у всіх раундах. DES, в свою чергу, використовує вісім різних S-box, які збільшують вимоги до реалізації.

Недоліки AES:

1. Використання дуже простої алгебраїчної структури.

2. Кожен блок зашифрований однаковим чином.

3. Складна програмна реалізація.

Обмежимось далі розглядом AES-128. В цьому випадку вхідними даними для алгоритму є масив з 16 байт in0 , in1 , ..., in15 . Перед початком шифрування байти цього масиву розміщуються послідовно в стовпці в матриці InputBlock (зверху вниз). Усередині алгоритму операції виконуються над матрицею станів State. Кінцеве значення матриці стану OutputBlock є виходом алгоритму і перетворюється в послідовність байтів шифротекста оut0 , out1 , ..., out15 .

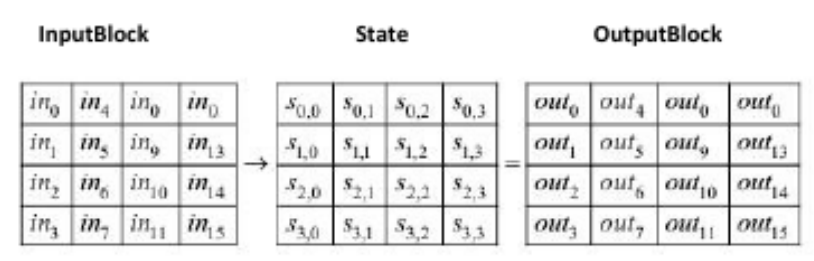


Рис.1. Основні матриці перетворень в AES-128

Аналогічно в стовпці матриці InputKey потрапляють і 16 байтів k0, k1, ..., k15 ключа шифру. (Рис.2.)

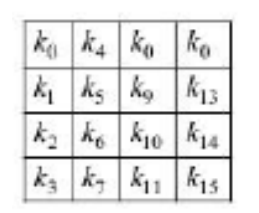


Рис.2. Матриця ключа в AES-128

Кожен байти ключа InputKey утворює слово , тобто фактично ключ шифру - це чотири слова w0, w1, w2, w3, де w0 =k0 k1 k2 k3, w1 =k4 k5 k6 k7 і т.д. З цих слів за допомогою спеціального алгоритму (розширення ключа) утворюється послідовність з 44 слів: w0, w1, w2,…,w43 (кожне слово по 32 біта).

На кожен раунд шифрування подаються по чотири слова цієї послідовності. Вони і будуть грати роль раундового ключа . Матриця, яка надходить на вхід кожного раунду, називається матрицею InputState , а на виході раунду утворюється матриця OutputState . Очевидно, на в ході першого раунду InputState = InputBlock, а на виході останнього раунду OutputState = OutputBlock. Схема перетворення даних показана на Рис.3.

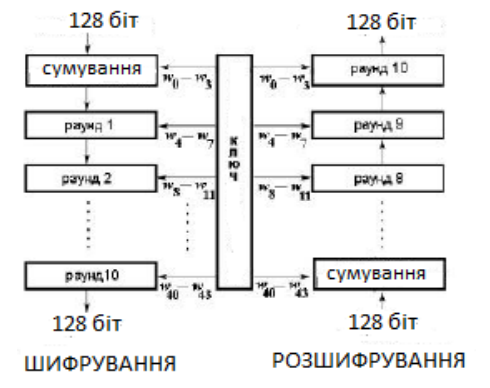


Рис.3. Схема перетворення даних в AES-128

Перед першим раундом виконується операція AddRoundKey (підсумовування по модулю 2 з початковим ключем шифру). Перетворення, виконані в одному раунді, позначають Round (State, RoundKey), де змінна State - матриця, що описуює дані на вході раунду і на його виході після шифрування ; змінна RoundKey - матриця, що містить раундовий ключ.

Раунд шифрування складається з 4 різних перетворень (Рис.4.):

1. SubBytes - побайтна підстановка а в S-боксі з фіксованою таблицею замін;

2. ShiftRows - побайтний зсув рядків матриці State на різну кількість байт;

3. MixColumns - перемішування байтів в стовпчиках;

4. AddRoundKey - складання з раундовим ключем (операція XOR ) .

Останній раунд дещо відрізняється від попередніх тим, що НЕ активізує функцію MixColumns . При розшифруванні в кожному раунді виконуються зворотні операції: InvShiftRows, InvSubBytes, AddRoundKey і InvMixColumns.

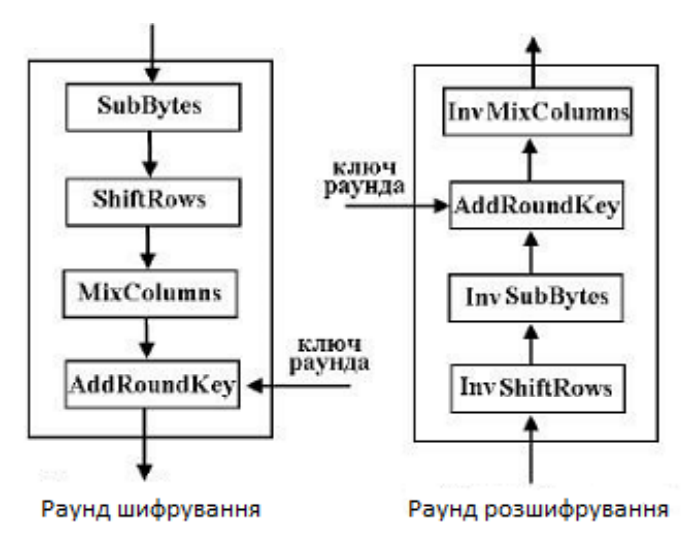


Рис.4. Операції в раундах шифрування/розшифрування AES-128

Розглянемо докладніше перетворення раунда шифрування.

1. SubBytes()

Процедура SubBytes() обробляє кожен байт стану незалежно (Рис.5.), проводячи нелінійну заміну байтів використовуючи таблицю замін (S-box).

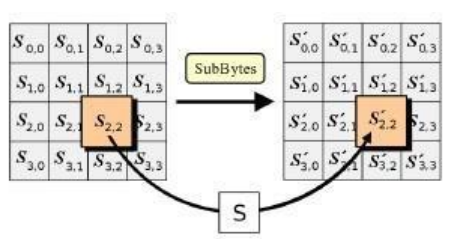


Рис.5. Операція SubBytes

Така операція забезпечує нелінійність алгоритму шифрування. S-box можна зобразити таблицею простої підстановки (Рис.6.).



Рис.6. S-box

2. ShiftRows()

Операція застосовується до рядків матриці State - її перший рядок нерухомий, а елементи нижніх трьох рядків циклічно зсуваються вправо на 1, 2 і 3 байти відповідно (Рис.7.).

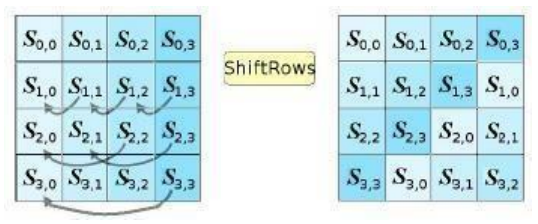


Рис.7. Операція ShiftRows

3. MixColumns()

За допомогою цієї операції виконується перемішування байтів в стовпчиках матриці State (Рис.8.). Кожен стовпець цієї матриці приймається за многочлен над полем GF(2^8 ) і множиться на фіксований многочлен c(x) по модулю многочлена x^4+1.

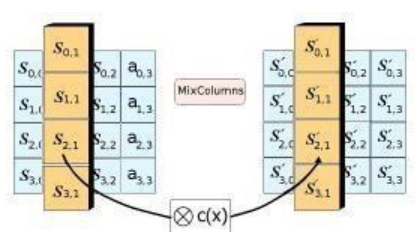


Рис.8. Операція MixColumns

4. AddRoundKey() Функція побітово складає елементи RoundKeyі елементи State (Рис.9.).

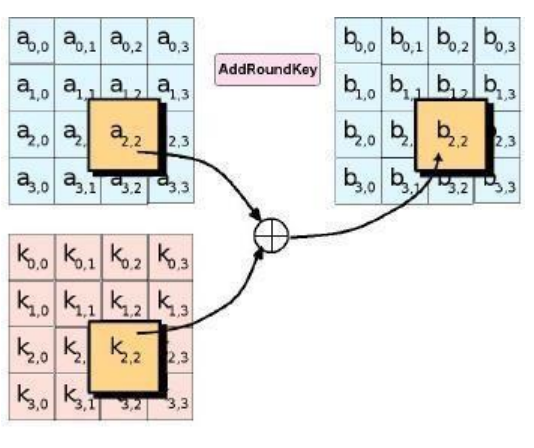


Рис.9. Операція AddRoundKey

# 

# **РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЇ**

Тема: “Розробка системи шифрування/розшифрування в режимі AES”

Що таке AES?

Розширений стандарт шифрування (AES) — це симетричний блоковий шифр, обраний урядом США для захисту секретної інформації.

AES реалізовано в програмному та апаратному забезпеченні по всьому світу для шифрування конфіденційних даних. Це має важливе значення для комп’ютерної безпеки уряду, кібербезпеки та захисту електронних даних.

Які особливості AES?

NIST зазначив, що новий алгоритм AES має бути блоковим шифром, здатним обробляти 128-бітні блоки, використовуючи ключі розміром 128, 192 і 256 бітів.

Інші критерії для обрання в якості наступного алгоритму AES включали наступне:

Безпека. Конкуруючі алгоритми повинні були оцінюватися за їхньою здатністю протистояти атакам порівняно з іншими представленими шифрами. Міцність безпеки мала вважатися найважливішим фактором змагань.

Вартість. Алгоритми-кандидати, які планувалося випустити на глобальній, неексклюзивній та безоплатній основі, мали оцінювати на ефективність обчислення та пам’яті.

Реалізація. Фактори, які слід враховувати, включали гнучкість алгоритму, придатність для реалізації апаратного чи програмного забезпечення та загальну простоту.

Яка різниця між AES і RSA?

AES широко використовується для захисту даних у стані спокою. Програми для AES включають:

диски з самошифруванням

шифрування бази даних

шифрування зберігання

Алгоритм RSA (Рівест-Шаміра-Адлемана) часто використовується у веб-браузерах для підключення до веб-сайтів, у віртуальних приватних мережах (VPN) і в багатьох інших програмах.

На відміну від AES, який використовує симетричне шифрування, RSA є основою асиметричної криптографії. Симетричне шифрування передбачає перетворення відкритого тексту на зашифрований за допомогою того самого ключа або секретного ключа для його шифрування та дешифрування. Термін «асиметричний» походить від того, що для шифрування використовуються два пов’язані ключі: відкритий і закритий. Якщо шифрування виконується за допомогою відкритого ключа, дешифрування може відбуватися лише за допомогою відповідного закритого ключа, і навпаки. Як правило, ключі RSA використовуються, коли є дві окремі кінцеві точки.

Хоча шифрування RSA добре працює для захисту передачі даних через географічні кордони, його продуктивність низька. Рішення полягає в тому, щоб поєднати шифрування RSA з шифруванням AES, щоб поєднати переваги безпеки RSA з продуктивністю AES. Це можна зробити, згенерувавши тимчасовий ключ AES і захистивши його за допомогою шифрування RSA.

Яка різниця між AES і DES?

Уряд США розробив алгоритми DES більше 40 років тому, щоб гарантувати, що всі державні системи використовують однаковий безпечний стандарт для полегшення взаємозв’язку.

DES служив стрижнею державної криптографії протягом багатьох років до 1999 року, коли дослідники зламали 56-бітний ключ алгоритму за допомогою розподіленої комп’ютерної системи. У 2000 році уряд США вирішив використовувати AES для захисту секретної інформації. DES все ще використовується в деяких випадках для зворотної сумісності.

Раніше DES був поширеним алгоритмом шифрування. У 2001 році його замінила AES.

Обидва стандарти є симетричними блоковими шифрами, але AES більш ефективний з математичної точки зору. Основна перевага AES полягає в параметрах довжини ключа. Час, необхідний для злому алгоритму шифрування, безпосередньо залежить від довжини ключа, який використовується для захисту зв’язку – 128-біт, 192-біт або 256-біт ключі. Тому AES є експоненціально сильнішим, ніж 56-бітний ключ DES. Шифрування AES також є значно швидшим, тому воно ідеально підходить для додатків, мікропрограм та апаратного забезпечення, які потребують низької затримки або високої пропускної здатності.

Написала програму на мові програмування java для реалізації системи шифрування/розшифрування в режимі AES

package test;

import javax.crypto.\*;

import javax.crypto.spec.IvParameterSpec;

import java.security.InvalidAlgorithmParameterException;

import java.security.InvalidKeyException;

import java.security.NoSuchAlgorithmException;

import java.util.Base64;

public class AESEncryption {

public static void main(String[] args) throws

NoSuchAlgorithmException, NoSuchPaddingException,

InvalidKeyException, IllegalBlockSizeException,

BadPaddingException, InvalidAlgorithmParameterException {

String message = "Розробка системи шифрування/розшифрування в режимі AES";

System.out.println("Original message : "+message);

KeyGenerator keyGenerator = KeyGenerator.getInstance("AES");

SecretKey secretKey = keyGenerator.generateKey();

String secretKeyString = Base64.getEncoder().encodeToString(secretKey.getEncoded());

System.out.println("Generated key : "+secretKeyString);

//Encrypt our message

Cipher encryptionCipher = Cipher.getInstance("AES/CBC/PKCS5Padding");

byte[] InitVectorBytes = keyGenerator.generateKey().getEncoded();

IvParameterSpec parameterSpec = new IvParameterSpec(InitVectorBytes);

encryptionCipher.init(Cipher.ENCRYPT\_MODE,secretKey,parameterSpec);

byte[] encryptedMessageBytes = encryptionCipher.doFinal(message.getBytes());

String encryptedMessage = Base64.getEncoder().encodeToString(encryptedMessageBytes);

System.out.println("Encrypted message : "+encryptedMessage);

//Decrypt the encrypted message

Cipher decryptionCipher = Cipher.getInstance("AES/CBC/PKCS5Padding");

decryptionCipher.init(Cipher.DECRYPT\_MODE,secretKey,parameterSpec);

byte[] decryptedMessageBytes = decryptionCipher.doFinal(encryptedMessageBytes);

String decryptedMessage = new String(decryptedMessageBytes);

System.out.println("Decrypted message : "+decryptedMessage);

}

}

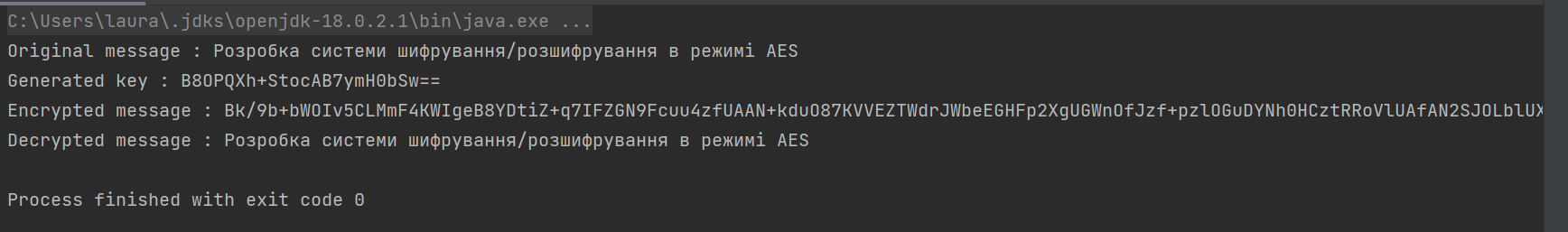


Рис.1. Результат роботи програми

Original message : Розробка системи шифрування/розшифрування в режимі AES

Generated key : B8OPQXh+StocAB7ymH0bSw==

Encrypted message : Bk/9b+bWOIv5CLMmF4KWIgeB8YDtiZ+q7IFZGN9Fcuu4zfUAAN+kduO87KVVEZTWdrJWbeEGHFp2XgUGWnOfJzf+pzlOGuDYNh0HCztRRoVlUAfAN2SJOLblUXm+nlC7sMM7PY72IksSZzyuEUnYAQ==

Decrypted message : Розробка системи шифрування/розшифрування в режимі AES

# **ВИСНОВКИ**

Проблема захисту комп'ютерних систем і мереж стає гострішою з розвитком і поширенням інформаційних технологій в усі сфери діяльності людини, суспільства і держави. Незважаючи на існуюче різноманітність сучасних засобів мережевої безпеки, інформаційні системи все більше піддаються впливам загроз, нові види яких породжують нові кібератаки на їх ресурси. В даній роботі я дослідила системи виявлення атак на комп'ютерну мережу та шифрування/розшифрування в режимі AES. Набір інструкцій AES тепер інтегрований у ЦП (пропонує пропускну здатність у кілька ГБ/с), щоб покращити швидкість і безпеку програм, які використовують AES для шифрування та дешифрування. Незважаючи на те, що минуло 20 років з моменту його впровадження, нам не вдалося зламати алгоритм AES, оскільки він неможливий навіть із сучасною технологією. На даний момент єдиною вразливістю залишається реалізація алгоритму. Завершуючи короткий опис наявних рішень у сфері виявлення атак, спробуємо відповісти питанням: чи підійдуть ці рішення у тому сучасному вигляді захисту мереж у недалекому майбутньому? Щодо цього є серйозні сумніви. Мережі стали настільки складними, що важко контролювати існуючими методами. Число вузлів зростає з небувалою швидкістю. З появою гігабітних швидкостей і комутованих віртуальних приватних мереж обсяг трафіку, що передається по мережах, зріс на кілька порядків. Виникає необхідність у нових підходах до виявлення атак, що дозволяють впоратися з цими проблемами.

# 

# **СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ**

1. <https://informationsecuritypalamarchuk.blogspot.com/2021/01/blog-post.html>
2. <https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%B2%D0%B8%D1%8F%D0%B2%D0%BB%D0%B5%D0%BD%D0%BD%D1%8F_%D0%B2%D1%82%D0%BE%D1%80%D0%B3%D0%BD%D0%B5%D0%BD%D1%8C#:~:text=%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0%20%D0%B2%D0%B8%D1%8F%D0%B2%D0%BB%D0%B5%D0%BD%D0%BD%D1%8F%20%D0%B0%D1%82%D0%B0%D0%BA%20(%D0%B2%D1%82%D0%BE%D1%80%D0%B3%D0%BD%D0%B5%D0%BD%D1%8C)%20>
3. <https://webcache.googleusercontent.com/search?q=cache:ofgDNG9najcJ:https://safe-surf.ru/specialists/article/5274/656701/&cd=7&hl=ru&ct=clnk&gl=ua>
4. <https://webcache.googleusercontent.com/search?q=cache:ofgDNG9najcJ:https://safe-surf.ru/specialists/article/5274/656701/&cd=7&hl=ru&ct=clnk&gl=ua>
5. <https://www.hindawi.com/journals/scn/2020/8872923/>
6. <https://phoenixnap.com/blog/intrusion-detection-system>
7. <https://uk.wikipedia.org/wiki/Advanced_Encryption_Standard>
8. <https://ela.kpi.ua/bitstream/123456789/23896/1/TZI_book.pdf>
9. <https://ela.kpi.ua/bitstream/123456789/26681/1/Kisilchuk_magistr.pdf>
10. <https://www.techtarget.com/searchsecurity/definition/Advanced-Encryption-Standard>