ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ТЕЛЕКОМУНІКАЦІЙ
Кафедра Мобільних та відеоінформаційних технологій

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Методи запобігання кіберзагрозам та супутниковим конфліктам у космічному просторі»

на здобуття освітнього ступеня магістра
зі спеціальності 172 Телекомунікації та радіотехніка
освітньо-професійної програми Апаратура радіозв'язку, радіомовлення і телебачення

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Микита КЛИМЕНКО
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Виконав: здобувач вищої освіти
групи АРДМ-62
Микита КЛИМЕНКО

Керівник: Анатолій МАКАРЕНКО

Рецензент: _____

Київ – 2023

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
Навчально-науковий інститут телекомунікацій

Кафедра Мобільних та відеоінформаційних технологій

Ступінь вищої освіти Магістр

Спеціальність 172 Телекомунікації та радіотехніка

Освітньо-професійна програма Апаратура радіозв'язку, радіомовлення і телебачення

ЗАТВЕРДЖУЮ
Завідувачка кафедрою МВТ
_____ Наталія РУДЕНКО
« » _____ 2023 року

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Клименко Микиті Володимировичу

1. Тема кваліфікаційної роботи: «Методи запобігання кіберзагрозам та супутниковим конфліктам у космічному просторі»,
керівник кваліфікаційної роботи Анатолій МАКАРЕНКО, д.т.н., професор,
затверджені наказом Державного університету інформаційно-комунікаційних технологій від 19.10.2023 р. №145.

2. Строк подання кваліфікаційної роботи 25.12.2023 р.

3. Вихідні дані до кваліфікаційної роботи:
   1. Структура телекомунікаційних супутникових систем .
   2. Технології шифрування супутникового звязку
   3. Оптимізація алгоритму шифрування.
   4. Науково-технічна література.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):
   1. Архітектура супутникових систем
   2. Архітектурна оптимізація паралельного алгоритму аутентифікованого шифрування для супутникових застосувань
   3. Імплементація методології і архітектурна оптимізація

5. Перелік ілюстративного матеріалу: презентація.

6. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1.	Підбір науково-технічної літератури	19.10.23	Викон.
2.	Архітектура супутникових систем	31.10.23	Викон.
3.	Архітектурна оптимізація паралельного алгоритму аутентифікованого шифрування для супутникових застосувань	15.11.23	Викон.
4.	Імплементація методології і архітектурна оптимізація	29.11.23	Викон.
5.	Висновки, вступ, реферат	11.12.23	Викон.
6.	Розробка презентації	18.12.23	Викон.

Здобувач вищої освіти Микита КЛИМЕНКО

(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Керівник кваліфікаційної роботи Анатолій МАКАРЕНКО

(підпис) *(Ім'я, ПРІЗВИЩЕ)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 81 стор., 26 рис., 23 дж.

Мета роботи – огляд кіберзагроз супутникових систем та розробка архітектурної оптимізації паралельного алгоритму аутентифікованого шифрування для супутникових застосувань.

Об'єкт дослідження – алгоритм шифрування супутникової системи комунікацій

Предмет дослідження – методи визначення ефективності алгоритму аутентифікованого шифрування для супутникових систем

Короткий зміст роботи: У роботі досліджено супутники та супутникові системи комунікацій. Оглянуто види кіберзагроз для космічних комунікацій і розглянуто реальні приклади атак. Досліджено вразливі частини супутникових систем та внутрішні системи штучних космічних комунікаційних об'єктів. Детально розглянуто AES, алгоритм симетричного шифрування та сам процес шифрування поетапно.

Запропоновано алгоритм що включає: генерацію IV, блочний шифр та алгоритм переключення ключів. Представлено методи реалізації, доступні для реалізації алгоритмів AE на FPGA. Також представлені різні архітектурні оптимізації для реалізації запропонованого алгоритму AE для супутникових додатків. Представлені результати реалізації, отримані в результаті реалізації запропонованого алгоритму AE на FPGA. Результати, отримані після застосування різних запропонованих оптимізацій, також наводяться разом з порівнянням з відомими алгоритмами AE, такими як алгоритм AES-GCM.

КЛЮЧОВІ СЛОВА: СУПУТНИК, СУПУТНИКОВА СИСТЕМА, КІБЕРЗАГРОЗИ, СУПУТНИКОВІ АТАКИ, АЛГОРИТМИ ШИФРУВАННЯ, АУТЕНТИФІКОВАНЕ ШИФРУВАННЯ, ТЕСТУВАННЯ, АРХІТЕКТУРНА ОПТИМІЗАЦІЯ, ПРОПУСКНА ЗДАТНІСТЬ,

ABSTRACT

Text part of the master's qualification work: 81 pages, 26 pictures, 23 sources.

The purpose of the work – review of cyber threats of satellite systems and development of architectural optimization of parallel authenticated encryption algorithm for satellite applications.

Object of research – satellite communication system encryption algorithm

Subject of research – methods of determining the effectiveness of the authenticated encryption algorithm for satellite systems

Summary of the work: The work examines satellites and satellite communication systems. The types of cyber threats to space communications are reviewed and real examples of attacks are considered. Vulnerable parts of satellite systems and internal systems of artificial space communication objects were studied. AES, the symmetric encryption algorithm and the encryption process itself are considered in detail step by step.

The proposed algorithm includes: IV generation, block cipher and key switching algorithm. Implementation methods available for implementing AE algorithms on FPGAs are presented. Various architectural optimizations for implementing the proposed AE algorithm for satellite applications are also presented. The implementation results obtained as a result of the implementation of the proposed AE algorithm on FPGA are presented. The results obtained after applying the various proposed optimizations are also presented along with a comparison with known AE algorithms such as the AES-GCM algorithm.

KEYWORDS: SATELLITE, SATELLITE SYSTEM, CYBER THREATS, SATELLITE ATTACKS, ENCRYPTION ALGORITHMS, AUTHENTICATED ENCRYPTION, TESTING, ARCHITECTURAL OPTIMIZATION, THROUGHTPUT CAPACITY,

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
Навчально-науковий інститут телекомунікацій

ПОДАННЯ
ГОЛОВІ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ
ЩОДО ЗАХИСТУ КВАЛІФІКАЦІЙНОЇ РОБОТИ
на здобуття освітнього ступеня магістра

Направляється здобувач Клименко М.В. до захисту кваліфікаційної роботи
(прізвище та ініціали)

за спеціальністю 172 Телекомунікації та радіотехніка
освітньо-професійної програми Апаратура радіозв'язку, радіомовлення і телебачення

на тему: Методи запобігання кіберзагрозам та супутниковим конфліктам у космічному просторі

Кваліфікаційна робота і рецензія додаються.

Директор ННІТ _____ Владислав КРАВЧЕНКО
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Висновок керівника кваліфікаційної роботи

Здобувач Клименко Микита Володимирович показав гарну теоретичну та інженерну підготовку, уміння володіти технологіями нового покоління, вміло працювати з програмно-апаратними засобами мобільного зв'язку, довідковою і науково-технічною літературою. Працюючи над завданнями, які доручались керівником, проявив ініціативність, сумлінність та хист до інженерної роботи.

Все це дозволяє оцінити виконану кваліфікаційну роботу Клименко Микити Володимировича на оцінку «відмінно» та присвоїти йому кваліфікацію «Магістр з телекомунікацій та радіотехніки».

Керівник кваліфікаційної роботи _____ Анатолій МАКАРЕНКО
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

“ ” _____ 2023 року

Висновок кафедри про кваліфікаційну роботу

Кваліфікаційна робота розглянута. Здобувач Клименко М.В.
(прізвище та ініціали)

допускається до захисту даної роботи в Екзаменаційній комісії.

Завідувачка кафедри
Мобільних та відеоінформаційних технологій _____ Наталія РУДЕНКО
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

“ ” _____ 2023 року

ВІДГУК РЕЦЕНЗЕНТА
на кваліфікаційну магістерську роботу

здобувача вищої освіти Клименко Микити Володимировича

на тему: «Методи запобігання кіберзагрозам та супутниковим конфліктам у космічному просторі»

Актуальність:

Однією з ключових причин, що підкреслює актуальність теми, є зростаюча залежність нашого сучасного суспільства від космічних технологій. Така значуща роль космічних технологій створює важливий контекст для розуміння, як кіберзагрози можуть впливати на ці системи. Інфраструктура космічних апаратів та супутників стає об'єктом зацікавлення для атак, оскільки їхній незахищений функціонал може призвести до серйозних наслідків. Постає питання у захищеному алгоритмі шифрування що буде спроможний обробити великі об'єми інформації достатньо забезпечивши захист інформації.

Тому, теоретичні та практичні дослідження направлені на вирішення даних завдань є актуальними і своєчасними.

Позитивні сторони:

1. Зміст магістерської роботи відповідає завданню. Її автор, показав високий рівень знань і ступінь підготовленості до майбутньої роботи за фахом.
2. Текст викладений якісно, ясно, послідовно. Детально розглянуто супутникові системи та загрози для них

Недоліки:

1. В роботі занадто багато уваги приділено теоретичним властивостям супутників та кіберзагрозам.
2. В роботі не в повному обсязі протестовано реалізації запропонованої архітектурної оптимізації алгоритму через відсутність необхідної технічної бази

Відзначені зауваження не впливають на загальну позитивну оцінку кваліфікаційної роботи.

Висновок: кваліфікаційна робота заслуговує оцінку «відмінно», а здобувач КЛИМЕНКО Микита Володимирович заслуговує присвоєння кваліфікації: «Магістр з телекомунікацій та радіотехніки».

Рецензент: д.т.н., проф. Єжов С.А.

(науковий ступінь, вчене звання) *(підпис)* *(Ім'я, ПРІЗВИЩЕ)*

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ 9
ВСТУП 11
1 АРХІТЕКТУРА СУПУТНИКОВИХ СИСТЕМ 14
1.1 Огляд супутників та супутникових систем 14
1.2 Архітектура супутникової системи 19
1.3 Поняття кібератак та супутникових конфліктів 20
1.4 Стандарт вдосконаленого шифрування 28
1.5 Корені довіри та конфіденційні обчислення 42
2 АРХІТЕКТУРНА ОПТИМІЗАЦІЯ ПАРАЛЕЛЬНОГО АЛГОРИТМУ АУТЕНТИФІКОВАНОГО ШИФРУВАННЯ ДЛЯ СУПУТНИКОВИХ ЗАСТОСУВАНЬ 46
2.1 Алгоритми шифрування 46
2.2 Алгоритми аутентифікованого шифрування 51
2.3 Запропонований алгоритм 54
3 ІМПЛЕМЕНТАЦІЯ МЕТОДОЛОГІЇ І АРХІТЕКТУРНА ОПТИМІЗАЦІЯ 58
3.1 Імплементація методології 58
3.2 Архітектурна оптимізація 59
3.3 Практичні результати 61
3.4 Результати Архітектурної оптимізації 64
3.5 Порівняння запропонованого алгоритму 70
3.6 Аналіз безпеки 71
ВИСНОВКИ 76
ПЕРЕЛІК ПОСИЛАННЬ 78

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

TT&C	Відстеження телеметрія та керування
HEO	Висока навколоземна орбіта
MEO	Середня навколоземна орбіта
GEO	Геостаціонарна орбіта
LEO	Низька навколоземна орбіта
BS	Базова станція
ES	Земна станція
MS	Мобільна станція
TDMA	Множинний доступ із поділом за часом
CDMA	Множинний доступ із поділом за кодом
MSC	Центр мобільної комутації
IER	Реєстр обладнання
VLR	Гостьовий реєстр
HLR	Домашній реєстр
AUC	Центр аутентифікації
ATM	Асинхронний режим передавання
PSTN	Телефонні мережі загального користування
SUMR	Реєстр картографування користувачів супутників
VSAT	Мала супутникова наземна станція
GPS/GNSS	Глобальна система супутникової навігації
IoT	Інтернет речей
AES	Удосконалений стандарт шифрування
RoT	Корінь довіри
TEE	Середовище, яке забезпечує рівень гарантії цілісності даних,
GCM	Лічильник з автентифікацією Галуа
FGPA	Програмній логічній матриці полей

HLS	Високорівневого синтезу
HDL	Мова текстового опису обладнання
LUT	Таблиці перетворення
SRAM	Статична оперативна пам'ять
AES-CTR	Розширений стандарт шифрування в режимі лічильника
AES-GCM	Розширений стандарт шифрування в режимі лічильника з автентифікацією Галуа
POET	Конвеєрне онлайн-шифрування з тегом автентифікації
NIST	Національний інститут стандартів і технологій США
DPA	Диференціальний аналіз потужності

ВСТУП

Супутниковий зв'язок є важливою складовою сучасної телекомунікаційної інфраструктури і відіграє вирішальну роль у забезпеченні широкосмугового зв'язку та передачі даних на великі відстані. За останні десятиліття супутниковий зв'язок зазнав значних технологічних проривів і став основою для безлічі сучасних застосувань, включаючи телекомунікаційні послуги, телевізійне мовлення, навігацію, наукові дослідження та зв'язок у важкодоступних регіонах.

В сучасному світі, орієнтованому на використання космічних технологій, проблема кібербезпеки та конфліктів у космічному просторі стає актуальною та надзвичайно важливою. Зростаючий обсяг космічної діяльності, включаючи запуск супутників, вивчення космосу та використання космічних ресурсів, призводить до збільшення ризику зазначення кіберзагроз та потенційних конфліктів у цій сфері. Важливі сектори, такі як телекомунікації, наукові дослідження, оборона та інші, все більше стають залежними від безперервного функціонування космічних технологій.

Однією з ключових причин, що підкреслює актуальність теми, є зростаюча залежність нашого сучасного суспільства від космічних технологій. Така значуща роль космічних технологій створює важливий контекст для розуміння, як кіберзагрози можуть впливати на ці системи. Інфраструктура космічних апаратів та супутників стає об'єктом зацікавлення для атак, оскільки їхній незахищений функціонал може призвести до серйозних наслідків. Атаки можуть спрямовуватися на збурення комунікацій, перешкоджання навігаційним системам, втручання у сучасні засоби оборони та, в загальному, порушення мирного використання космічного простору.

Крім того, оскільки космічні апарати та супутники використовуються для збирання та передачі важливої інформації, кіберзагрози можуть мати серйозні наслідки для збереження конфіденційності та цілісності даних. Шпигунство, крадіжка конфіденційної інформації чи зміна даних можуть стати серйозними загрозами для національної та глобальної безпеки. Особливу увагу слід приділяти методам захисту від проникнення зловмисників в обладнання через програмне

забезпечення або через погано захищені канали зв'язку. Адже доступ до даних супутника відіграє ключову роль в його фізичному існуванні, та забезпеченні правильного рівня доступу до користувачів

З урахуванням цих обставин, розробка та впровадження ефективних методів запобігання кіберзагрозам стає пріоритетним завданням, яке має на меті забезпечити стійкість та безпеку в космічному просторі, де сучасні технології та залежність від них визначають велику частину нашого повсякденного життя.

Сучасний стан кібербезпеки в космічному просторі породжує ряд викликів, серед яких особливе значення має питання алгоритмів шифрування. Однією з найбільш актуальних проблем є недостатня стійкість алгоритмів шифрування, що застосовуються в космічних системах та супутникових з'єднаннях.

Багато космічних систем були розроблені та запущені в епоху, коли кіберзагрози не ставали настільки складними та вдосконаленими, як сьогодні. Інфраструктура, яка була безпечною під час свого створення, може бути зараз вразливою до сучасних кіберзагроз. Традиційні алгоритми шифрування, які вважались стійкими, можуть бути піддані ризику через потужність квантових обчислень.

Супутники та космічні апарати стають привабливими об'єктами для кібератак через їхню значущість у сучасному світі. Несправний функціонал супутників, викликаний атакою на їхні алгоритми шифрування, може призвести до серйозних порушень безпеки та порушень в роботі супутникових систем.

Збільшення кількості даних, що передаються через супутники, робить актуальним питання конфіденційності цих даних. Атаки на алгоритми шифрування можуть покласти під загрозу конфіденційні дані, що пересилаються та зберігаються на космічних апаратах.

З урахуванням цих проблем, важливим стає розробка та впровадження нових, стійких до квантового криптоаналізу алгоритмів шифрування, а також вдосконалення та адаптація існуючих з метою забезпечення найвищого рівня кібербезпеки в космічному просторі.

Метою роботи слугує огляд кіберзагроз супутникових систем та розробка архітектурної оптимізації паралельного алгоритму аутентифікованого шифрування для супутникових застосувань.

Об'єктом дослідження являється алгоритм шифрування супутникової системи комунікацій

Предмет дослідження – це методи визначення ефективності алгоритму аутентифікованого шифрування для супутникових систем

1 АРХІТЕКТУРА СУПУТНИКОВИХ СИСТЕМ

1.1 Огляд супутників та супутникових систем

Супутник — в галузі телекомунікацій, розглядається як об'єкт створений людиною для використання в цілях передачі інформації між наземними станціями та іншими супутниками аби задовольнити доволі різноманітний спектр людських потреб. Що включає в себе супутники: для спостереження за природою, та прогнозування клімату; комунікації між різними точками планети, особливо що стосується важкодоступних місць для використання наземних мереж та магістралей зв'язку; для використання у військових цілях, спостереженні та виявлення різноманітних цілей та об'єктів з використанням технологій, що дають можливість пошуку в будь-який час доби та будь-якої погодної обстановки.

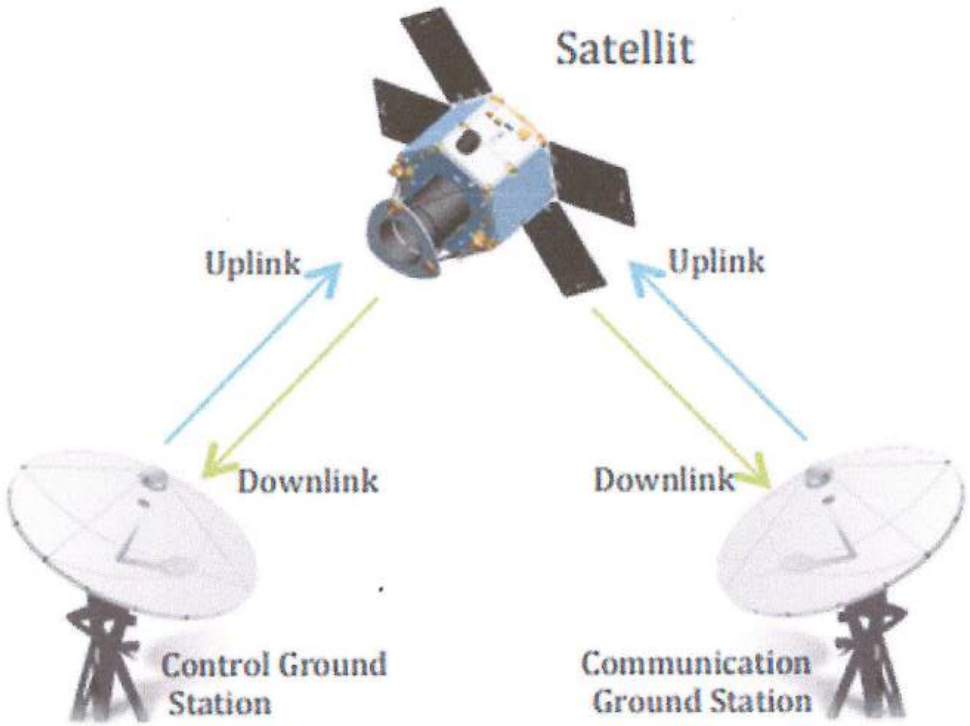


Рис.1.1 Супутникова система

Супутникова система — це сукупність пристроїв та космічних апаратів що взаємодіють із супутниками для обміну інформацією.(рис.1.1). Також такі системи використовують для автоматизації керування та обслуговування супутникового зв'язку. Загалом супутникова система складається з самого супутника, наземних

станцій зв'язку, на яких обробляються дані, такі як голос та зображення, наземної станції TT&C (відстеження, телеметрія та керування), яка гарантує залишення супутника на орбіті шляхом його відстеження, а також ліній зв'язку вниз і вгору, що з'єднують супутник з цими станціями. [1]

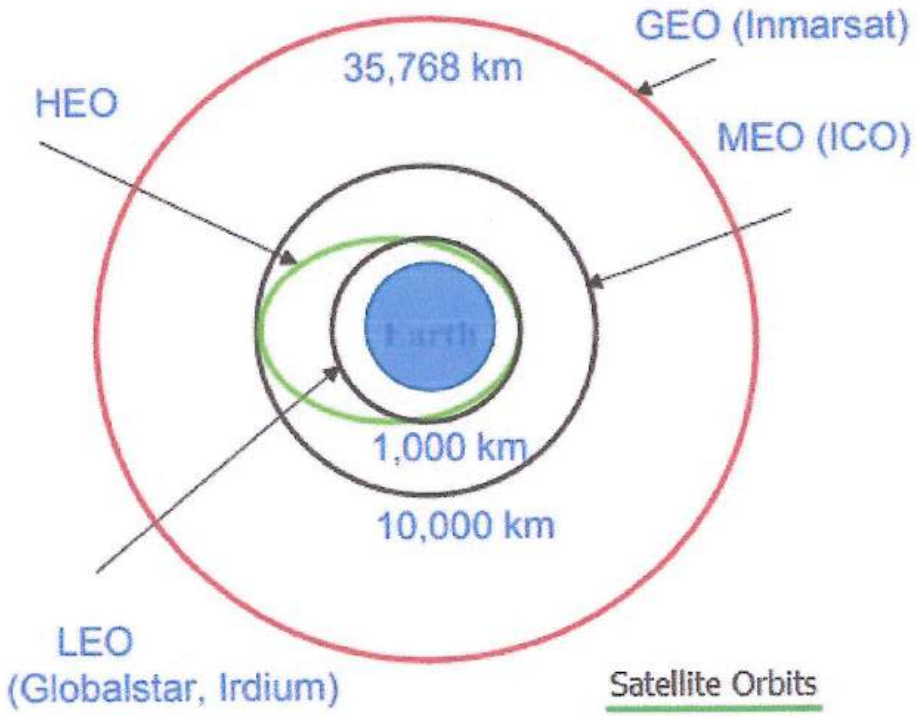


Рис1.2 Орбіти супутників

Загалом супутники це дуже різноманітні об'єкти запущені людиною для низки цілей, основна характеристика що визначає більшість його можливостей та призначень це тип навколоземної орбіти. Космічні апарати що обертаються довколо землі можуть відрізнятися за своєю орбітою (рис1.2.), котра в залежності від відстані до земної поверхні впливає на ключові характеристики та напрямки застосування. Взагальному класифікуютть такі види орбіт:

1. Висока навколоземна орбіта (HEO)
2. Середня навколоземна орбіта (MEO)
3. Низька навколоземна орбіта (LEO)
4. Геостаціонарна орбіта (GEO)

Супутники GEO, які знаходяться на висоті 35 768 км, обертаються в екваторіальній площині з нахилом нульового градуса і здійснюють рівно один оберт за день. В таких пристроях використовуються антени з фіксованим положенням, а

також група висхідних каналів (зворотний діапазон) 1634,5–1660,5 МГц і група низхідних каналів (прямий діапазон) у діапазоні 1530–1559 МГц. Для з'єднання між базовою станцією (земною станцією) та супутниками використовуються частоти діапазону Ku (11 і 13 ГГц). Супутник зазвичай має великий слід, який може покривати до 34% поверхні Землі, тому важко повторно використовувати частоти. Глобальне покриття невеликих мобільних телефонів і передачі даних зазвичай викликає високу затримку в діапазоні близько 275 мс.

Супутники LEO поділяються на малі та великі. Малі LEO мають менший розмір і знаходяться в діапазоні частот від 148 до 150. Вони викликають буквено-цифрові дисплеї зі низькою швидкістю передачі бітів (порядку 1 кбіт/с) для двосторонньої інформації та позиціонування. Великі супутники LEO мають достатню потужність і пропускну здатність для надання різних глобальних мобільних послуг (тобто передачі даних, пейджингу, факсимільного зв'язку та визначення місцезнаходження), а також високоякісні голосові послуги для мобільних систем, таких як керовані пристрої та автомобільні трансивери. Великі LEO передають у діапазоні частот від 1610 до 1626,5 МГц (висхідний канал) і від 2483,5 до 2500 МГц (низхідний канал) і обертаються на висоті близько 500–1500 км над поверхнею Землі. Затримка становить близько 5–10 мс, а супутник видимий близько 10–40 мс.

Для забезпечення глобального покриття потрібні кілька супутників. Той самий спектр частот використовується також MEO та GEO.

У системах MEO повільно рухомі супутники обертаються на висоті близько 5000–12000 км над Землею і мають затримку близько 70–80 мс.

Для забезпечення менших слідів і вищої потужності передачі використовуються спеціальні антени.

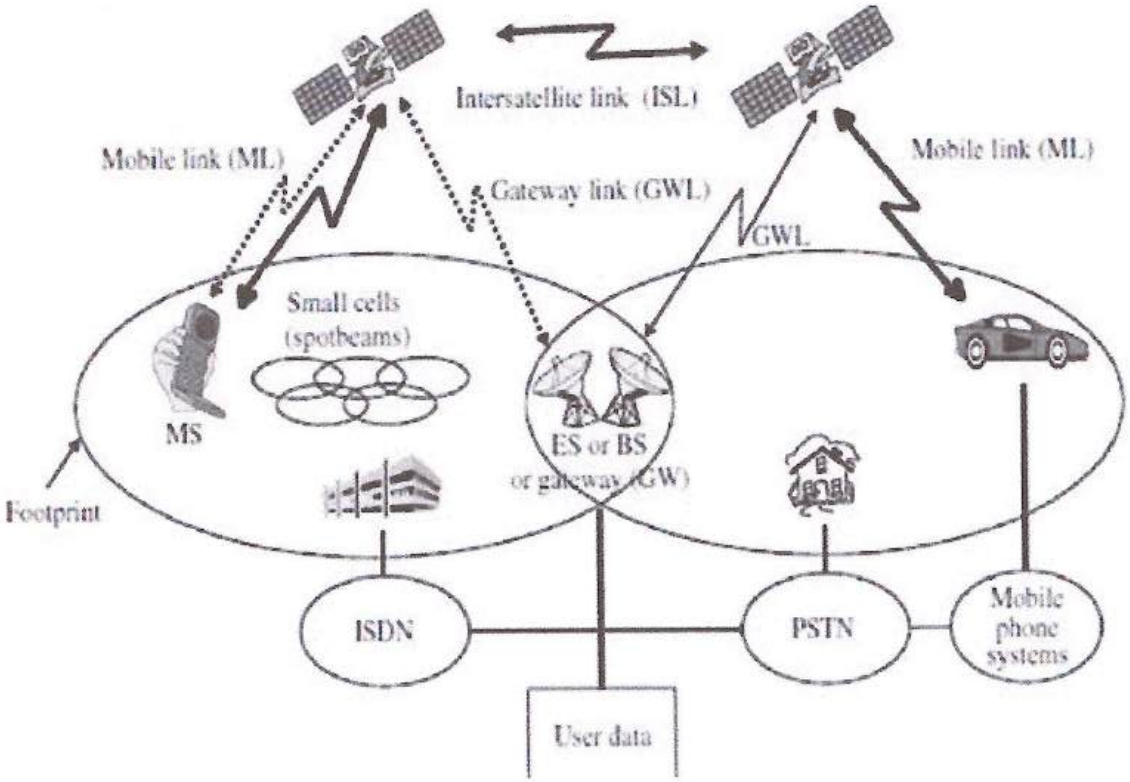


Рис.1.3. Загальна схема супутникової системи

Є багато груп елементів, які забезпечують роботу супутникової інфраструктури. Для розуміння роботи всієї системи необхідний детальний огляд. Приклад схемного представлення супутникової системи показано на рисунку 1.3 з численними компонентами, показаними явно: Інфраструктура супутникової системи. Після встановлення контакту між мобільною системою та супутником за допомогою променя LOS можна отримати доступ майже до будь-якої людини у світі за допомогою базової апаратної магістральної мережі на поверхні землі. Управління супутниками здійснюється базовими станціями (BS), розташованими на поверхні землі, які служать шлюзом. Міжсупутникові лінії зв'язку можна використовувати для передачі інформації від одного супутника до іншого, але вони все ще контролюються наземною BS (також відомою як земна станція або ES).Освітлена зона супутникового променя, яка називається слідом, — це область, у межах якої мобільний користувач може спілкуватися із супутником; багато променів використовуються для покриття великої площі. Крім того, супутники постійно обертаються навколо землі, і промінь може бути тимчасово заблокований або іншими літаючими об'єктами, або рельєфом земної поверхні. Таким чином,

концепція надлишковості, відома як рознесення, використовується для передачі одного повідомлення через кілька супутників, як показано на рисунку 1.4. Основна ідея рознесення шляхів полягає в тому, щоб забезпечити механізм, який може поєднувати два або більше корельованих інформаційних сигналів (переважно одну і ту саму копію), що рухаються різними шляхами і, отже, мають невиправлені характеристики шуму та/або завмирання. Така комбінація двох сигналів покращує якість сигналу, що дозволяє приймачу мати гнучкість у виборі сигналу кращої якості.

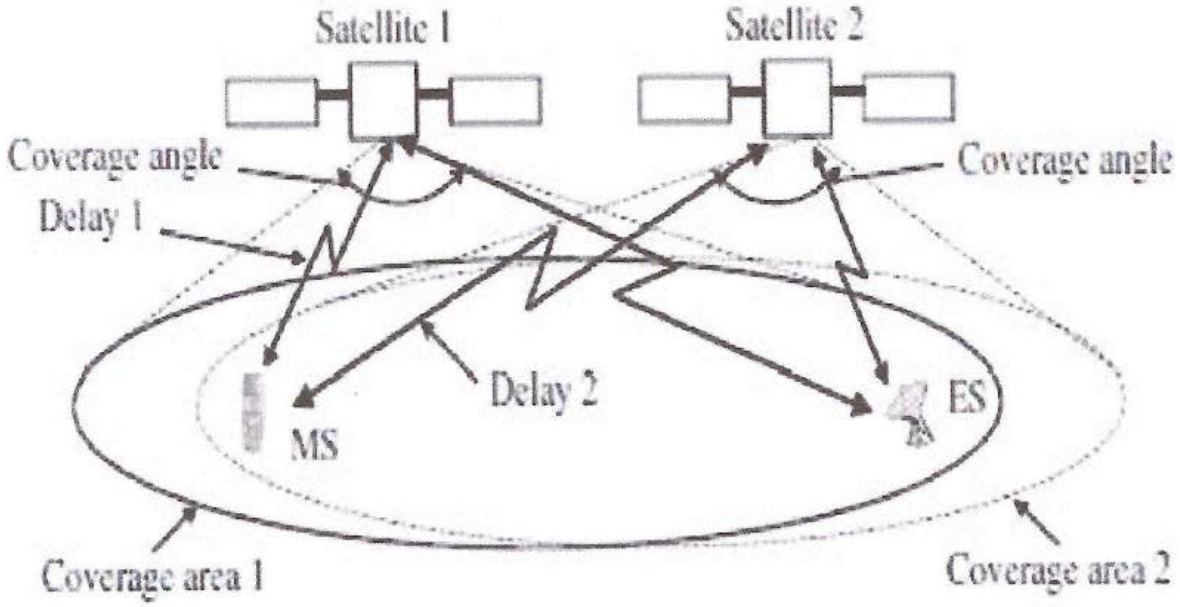


Рис.1.4 Рознесення супутникових з'єднань

В інфраструктурі супутникової системи основний інтерес викликає рознесення шляхів, хоча можливі й інші форми рознесення, такі як антена, час, частота, поле або код. Рознесення шляхів залежатиме від технології, яка використовується для передачі та отримання повідомлень. Використання рознесення може бути ініційоване MS або BS, розташованими на землі. Запит на рознесення від BS (ES) дозволяє MS знаходити та сканувати незатінені супутникові пейджингові канали для безперешкодного зв'язку. Така ситуація не може бути виявлена або визначена BS, навіть якщо місцезнаходження MS відоме BS. Використання рознесеного супутникового тракту може бути насамперед зумовлено

наступними умовами: Кут підйому: вищий кут підйому зменшує проблеми із затіненням. Одним із підходів є ініціювання рознесення траєкторії, коли кут підйому стає меншим за попередньо визначене порогове значення. Якість сигналу: якщо середній рівень сигналу (у ДБ), якість (у BER) або тривалість завмирання виходять за межі деякого порогу, тоді можна використовувати рознесення тракту. Якість сигналу є функцією таких параметрів, як кут підйому, доступна ємність, поточна модель мобільності MS або очікуваний майбутній попит. Опція очікування: канал може бути обраний і зарезервований як резервний для рознесення щоразу, коли MS виявляє перетин порогу. Такий резервний канал використовується лише тоді, коли основний канал заблокований. Оскільки використання рознесення вважається рідкісним явищем, кілька MS можуть спільно використовувати один і той же резервний канал. Екстрена передача обслуговування: щоразу, коли втрачається з'єднання MS із супутником, MS із супутником втрачається; MS намагається здійснити екстрену передачу обслуговування.

1.2 Архітектура супутникової системи

Загальна архітектура супутникової системи зображена на рисунку 1.5. Інфраструктура супутникової системи ES (BS), що становить серце загального управління системою. ES виконує функції, подібні до BSS стільникової бездротової системи. ES відстежує всі MS, розташовані в зоні, і контролює розподіл і дерозподіл радіоресурсів. Це включає використання діапазону частот або каналу в FDMA, часовий інтервал для TDMA, призначення коду для CDMA. І ATM, і VLR є важливими частинами BS і забезпечують функції, подібні до функцій стільникової мережі. База даних EIR (Equipment Identity Register), AUC (Authentication Center) і HLR також виконують ті самі операції, що й у звичайних бездротових системах, і є невід'ємною частиною загальної супутникової системи. Пара HLR-VLR (Home Location Register - Visitor Location Register) підтримує базовий процес керування мобільністю.

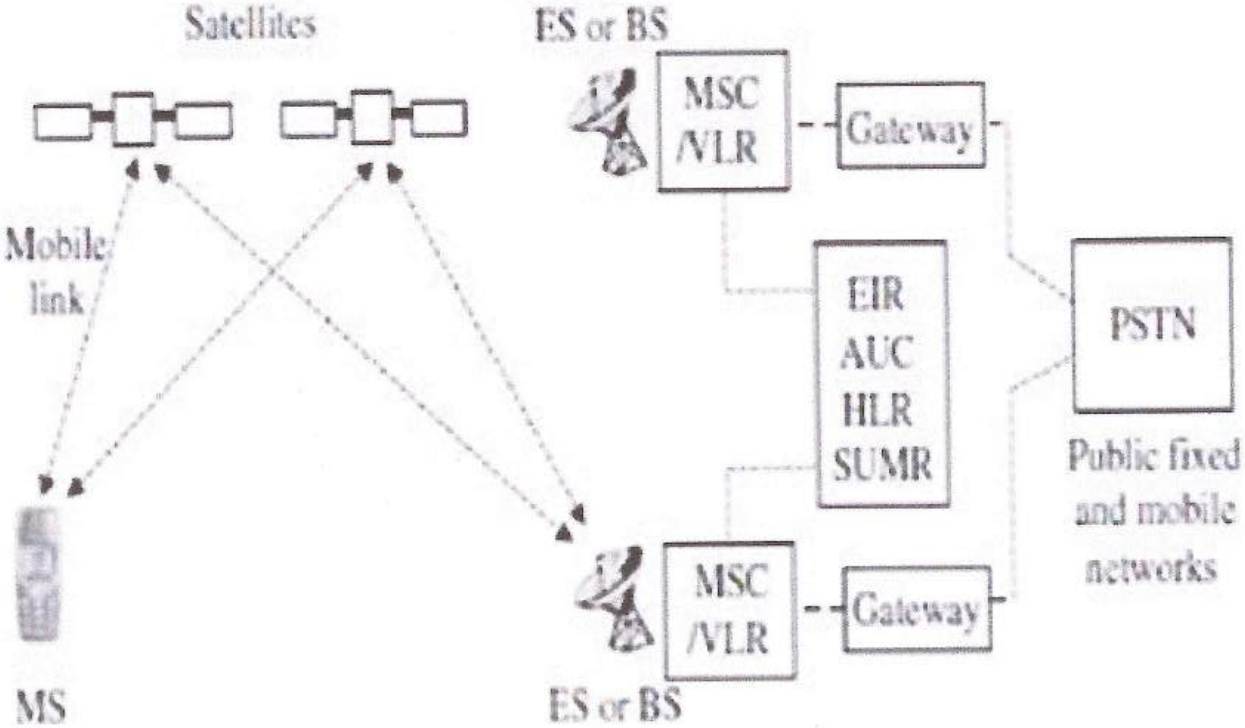


Рис.1.5 Загальна архітектура супутникової системи

Реєстр картографування користувачів супутників (SUMR) також підтримується на BS, щоб відзначати розташування всіх супутників і вказувати супутник, призначений кожній MS. Всі ці системи пов'язані з БС для мінімізації ваги супутників. Насправді супутники можна розглядати як ретрансляційні станції з охопленням у всьому світі, враховуючи, що більша частина процесу розвідки та прийняття рішень виконується BS. Ці базові станції також підключені до PSTN і магістралі ATM через відповідний шлюз, щоб можна було маршрутизувати та встановлювати виклики на звичайні домашні телефони, а також на стільникові пристрої.

1.3 Поняття кібератак та супутникових конфліктів

Автономна робота в досить суворому космічному просторі та відсутність можливості вільно та суттєво керувати супутниками призводить до великої кількості проблем. Насьогодні супутникові системи є досить вразливими до зовнішніх факторів, а саме космічного забруднення, що спричинене досить

маштабним використанням космічних апаратів що знаходяться на різних орбітах довкола землі, та діяльність людей що керують іншими космічними апаратами що можуть зашкодити або навіть знищити штучний супутник. Супутникові конфлікти у космічному просторі з'явилися через те що космічний простір став предметом ростучого інтересу та активної діяльності багатьох країн. З цим зростаючим інтересом і виникають потенційні конфлікти в космосі. Причинами таких конфліктів можуть бути:

Заборона на випробування та розгортання зброї. Хоч багато країн визнають потребу в міжнародних домовленостях та договорах, що забороняють військове використання космосу. Проте відсутність чітких правил і норм може створити потенційний простір для конфліктів.

Розвиток антисупутникової зброї, такої як ракети, які здатні збити чи зруйнувати супутник, створює загрозу для супутників і космічних систем зв'язку.

Кібератаки на космічні системи, включаючи управління супутниками, можуть вимкнути або ушкодити космічні об'єкти та викликати конфлікти.

Заборона на випробування та розгортання супутників. Конфлікти можуть виникнути внаслідок спроб розгортання супутників, які одна країна вважає загрозою для своєї національної безпеки.

Запити на обрання однією супутниковою орбітою однакової орбітальної області можуть викликати супутникові конфлікти.

Використання космосу для військового розвідування. Військові супутники використовуються для розвідки і збору інформації. Це може викликати напруженість та конфлікти між країнами.

Кіберзагрози та супутникові конфлікти у космічному просторі - це два взаємопов'язані поняття, які становлять серйозну загрозу для національної безпеки та стабільності комунікаційних систем. Кіберзагрози - це будь-які дії, спрямовані на порушення функціонування космічних систем за допомогою комп'ютерних атак.

Кіберзагрози в космічному просторі стали актуальними через зростання супутникового зв'язку, важливості космічних систем для нашого сучасного способу життя та економіки. Основні причини та загрози, пов'язані з кіберзагрозами у

космічному просторі. Сучасна суспільність дуже сильно залежить від супутникового зв'язку для зв'язку, навігації, метеорологічних даних і багатьох інших сфер. Атаки на супутникові системи можуть призвести до великих руйнувань та перебоїв у комунікаціях, що вплине на безпеку і ефективність. Супутники мають обмежені ресурси, такі як обчислювальна потужність і споживана енергія. Кібератаки можуть використовувати ці ресурси, що може призвести до зниження функціональності супутників. Кібератаки можуть переповнити супутникові канали зайвими даними, що може призвести до зниження якості обслуговування та перешкод в зв'язку. Кібератаки можуть впливати на системи керування супутниками і втрату зв'язку з ними, що може вести до тимчасового або навіть постійного втрати контролю над супутниками. Кібератаки можуть націлюватися на космічні апарати та зонди, втручаючи в їх функцію або призводячи до їх збоїв. Космічний простір також важливий для військових дій, і кібератаки можуть використовуватися для завдання шкоди ворожим космічним системам. Кібератаки можуть призвести до порушення роботи космічних місій, що може коштувати життя і значні фінансові ресурси.

Кібератаки можуть бути використані для розвідки і збирання цінної інформації про космічні системи та діяльність космічних агентств.

Для захисту від цих загроз важливо розвивати кіберзахист космічних систем, моніторинг космічного простору на наявність загроз та співпрацювати на міжнародному рівні для забезпечення безпеки в космосі. Для запобігання супутниковим конфліктам і вирішення суперечок у космічному просторі існують різні міжнародні угоди і договори, такі як Договір про космічну діяльність та Договір про заборону атомних вибухів у космосі. Однак дотримання таких договорів і міжнародних норм може бути предметом суперечок та незгод між країнами.

Можливі напрямки кібератак

Кібератаки, вже далеко не новинка для сучасних наземних мереж зв'язку, але при цьому також слід враховувати паралельно з атаками на наземний сегмент

змінюються та вдосконалюються методи атак на супутникові системи, що в останні десятиліття є надважливою ланкою комунікацій світу.

Кібератака не є монолітною загрозою — вона може приймати різні форми, мати різноманітні вектори проникнення та використання, а також може спричинити безліч паралізуючих ефектів, коли її спрацьовує.

- Взлом - це практика зміни та налаштування комп'ютерного обладнання або програмного забезпечення з метою досягнення мети, яка не відповідає цілям творця комп'ютерного програмного забезпечення/обладнання.[2] Подібно до комп'ютерів, системи супутникового зв'язку вразливі до злому. Якщо супутникову систему зламано, зловмисник може контролювати, порушувати та пошкоджувати супутник. Більшість поширених способів злому - це злом пристрою зв'язку VSAT, який, якщо його зламати, дані та пов'язані з ним пристрої будуть під загрозою.
- Підробка означає провокування системи-одержувача на прийняття підтвердження особи чи чогось шляхом надсилання помилкових сигналів на приймач системи. Згідно зі звітом Управління з питань загального обліку, деякі супутникові організації не шифрують і не шифрують відстеження та керування висхідного каналу, що робить супутник надзвичайно вразливим до атак підробки[3]
- Заглушення - це порушення сигналів радіочастоти шляхом передачі інших сигналів радіочастоти, щоб надіслана частота не досягла пункту призначення. Крім того, заглушення - це надсилання сигналів шуму передачі через один GPS/GNSS або більше, що викликає перевантаження схеми приймача. Тому заглушення вважається найпростішим способом злому.[4]

Заглушення може відбуватися на висхідному або низхідному каналі, але вимоги до заглушення кожного з них відрізняються. Оскільки висхідний інтерфейс із супутником на великій площі, заглушення висхідного каналу має менший вплив, але на короткий проміжок часу. З іншого боку, заглушення каналу управління

висхідного каналу може перешкодити супутнику отримувати команди від наземних станцій.

Щоб заглушити, глушник повинен мати необхідну потужність, щоб він перевершив джерело частоти, також потрібно, щоб глушник знав про ефективну частоту. Крім того, проста зловмисна кібератака на комп'ютери наземної станції управління може призвести до заглушення супутника

У 2013 році в Лондоні стався випадок заглушення, коли сигнали GPS були заблоковані. Внаслідок цього навігаційні системи в автомобілях перестали працювати, що призвело до негативних наслідків для фінансових торгів.

- Викрадення - це незаконна передача за допомогою супутника або захоплення контролю над сигналом, таким як трансляція, та перемикання його на інший.[5] Тоді як управління є найскладнішою формою злому, при якій хакери контролюють супутник на орбіті, порушуючи та беручи під контроль наземну станцію TT&C, корисне навантаження або/та шину, що. Порушуючи зв'язки TT&C, хакери можуть віддавати команди супутнику, щоб дати йому неправильні напрямки, спалити його, повторно ввівши в атмосферу Землі, та інші. Через найвищий рівень безпеки на цьому рівні супутникових частин, управління супутником є значно складним. Однією з умов, за якої відбувається контроль, є виявлення слабкого місця в управлінні та контролі комерційних супутників, таких як хаби VSAT.

Хоча система супутника знаходиться в стадії розробки на землі, вона піддається впливу багатьох тих самих загроз, з якими стикаються традиційні вбудовані системи. Сам супутник можна порівняти з пристроєм Інтернету речей (IoT) у космосі. Зловмисники, безсумнівно, шукають способи використати наявні вразливості для отримання доступу до можливостей космічних апаратів та впливу на них.

До підсистем космічних апаратів, які вразливі до кібератак, належать(рис1.6.):

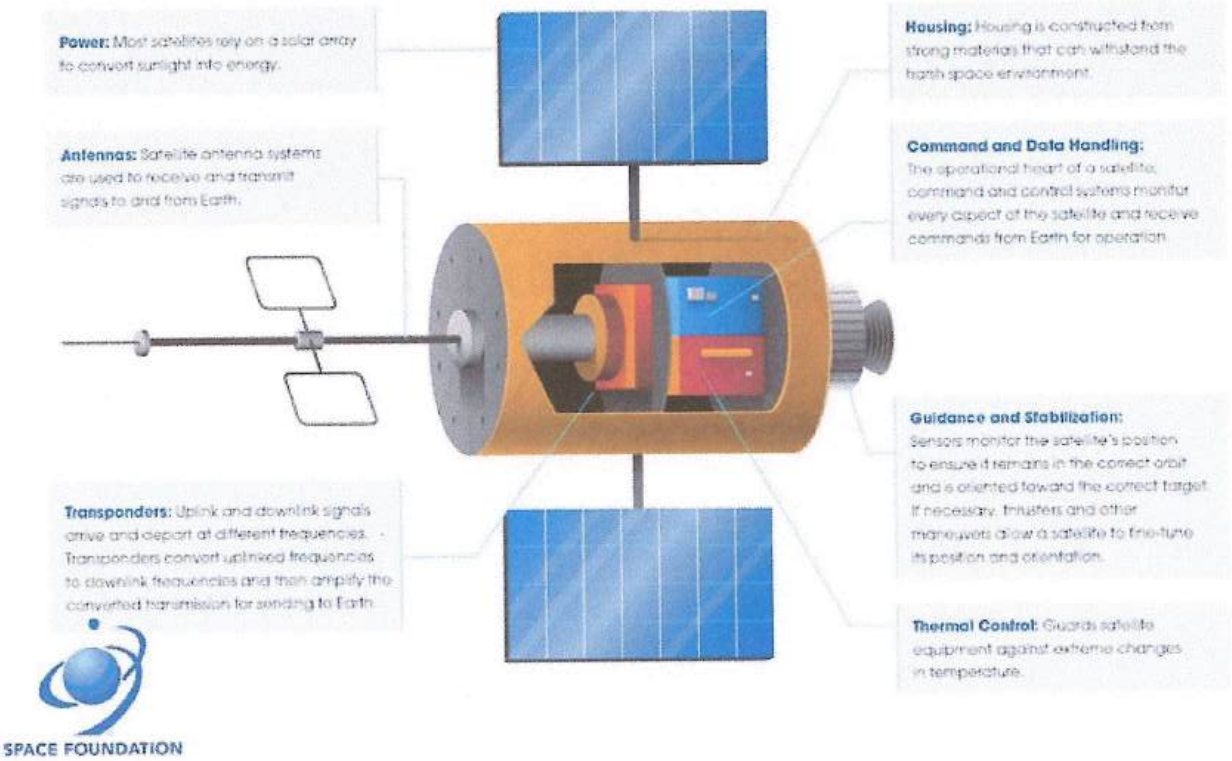


Рис.1.6 Підсистеми космічного апарату

- Підсистема визначення та управління орієнтацією (AD&C): AD&C відповідає за визначення та підтримку правильного положення та орієнтації супутника. Якщо зловмисник отримає доступ до AD&C, він може вивести супутник з ладу або направити його на неправильну ціль.
- Підсистема обробки команд і даних (C&DH): C&DH відповідає за обробку команд, що надходять із Землі, та телеметрії, що надходить із супутника. Якщо зловмисник отримає доступ до C&DH, він може віддати супутнику шкідливі команди або підробити телеметрію.
- Підсистема електропостачання та розподілу (EPDS): EPDS відповідає за генерування, зберігання та розподілу електроенергії на супутнику. Якщо зловмисник отримає доступ до EPDS, він може вивести супутник з ладу, вимкнувши живлення критичних систем.
- Рухова підсистема (PS): PS відповідає за переміщення супутника на орбіті. Якщо зловмисник отримає доступ до PS, він може змінити орбіту супутника або вивести його з ладу.

- Підсистема конструкцій та механізмів (SMS): SMS відповідає за фізичну структуру супутника та його механічні системи. Якщо зловмисник отримає доступ до SMS, він може пошкодити супутник або вивести його з ладу.
- Телеметрія, відстеження та управління (TT&C): TT&C відповідає за зв'язок між супутником та Землею. Якщо зловмисник отримає доступ до TT&C, він може перехопити дані або порушити зв'язок.
- Підсистема терморегулювання (TCS): TCS відповідає за підтримкою належної температури на супутнику. Якщо зловмисник отримає доступ до TCS, він може пошкодити супутник, перегрівши або переохолодивши його компоненти.

Інциденти кібератак на супутники

Першим широко відомим інцидентом успішної кібер атаки на супутник спостереження був інцидент 15 серпня 2008 року. Група хакерів, які, ймовірно, були пов'язані з Іраном, захопила управління супутником NASA Terra EOS. Супутник Terra EOS є супутником спостереження за Землею, який використовується для збору даних про клімат, навколишнє середовище та інші аспекти Землі. Хакери змогли отримати доступ до системи управління та контролю супутника, скориставшись уразливістю в програмному забезпеченні системи.

Уразливість, яку використовували хакери, була пов'язана з тим, що супутник Terra EOS використовував ненадійний алгоритм шифрування для захисту своїх команд. Алгоритм був відомий як RC4 і був розроблений в 1987 році. RC4 є слабким алгоритмом шифрування, і він був зламаний у 1994 році.

Хакери змогли зламати алгоритм RC4, використовуючи інструмент, який називається "алгоритм атакування ключів". Алгоритм атакування ключів - це метод, який використовується зловмисниками для зламу алгоритму шифрування. Алгоритм атакування ключів дозволяє хакерам спробувати різні ключі, поки вони не знайдуть правильний ключ. Цей інструмент дозволяє хакерам спробувати різні ключі, поки вони не знайдуть правильний ключ.

Після того, як хакери отримали контроль над супутником, вони змінили його напрямок, направивши його на території Ірану. Супутник зробив кілька знімків цих територій, які, ймовірно, були використані Іраном для військових цілей. NASA було повідомлено про атаку через кілька годин після того, як вона відбулася. Агентство змогло відновити контроль над супутником і запобігти подальшому поширенню шкоди. NASA впровадило новий алгоритм шифрування для захисту своїх команд. Цей алгоритм є більш надійним, ніж RC4. Атака на супутник Terra EOS була важливим попередженням про потенційну загрозу кібератак для супутників. Вона показала, що супутники є вразливими до кібератак і що необхідно вжити заходів для захисту цих систем.

Остання з відомих нині атак на супутникову мережу, була доповненням до військового наступу що почали росіяни на Україну. Атака відбулася 23 лютого 2023 року, за день до вторгнення Росії в Україну. Хакери використовували шкідливе програмне забезпечення, яке називається Acid Rain, щоб стерти вміст тисяч модемів, які використовувалися для підключення до супутника Viasat KA-SAT. Ці модеми використовувалися для надання широкосмугового доступу до Інтернету, супутникового телебачення та інших послуг у Європі та частині Близького Сходу. Атака призвела до того, що від 40 000 до 45 000 модемів стали непридатними для використання. Це призвело до значних перешкод у зв'язку в Україні та інших країнах Європи.

Атака відбулася в два етапи. На першому етапі хакери використали шкідливе програмне забезпечення Acid Rain, щоб стерти вміст модемів. На другому етапі хакери обробили мережу Viasat KA-SAT величезною кількістю запитів, що призвело до її перевантаження. Це зробило неможливим для модемів підключитися до мережі. Атака на Viasat мала значний вплив на зв'язок в Україні та інших країнах Європи. В Україні атака призвела до того, що багато людей втратили доступ до Інтернету, що ускладнило їм отримання інформації про вторгнення росії, також українське військове командування покладалося на використання саме Viasat для керування військами. Атака також призвела до перешкод у зв'язку в інших країнах Європи. У деяких випадках атака призвела до того, що люди втратили доступ до

супутникового телебачення та інших послуг. Після атаки Viasat впровадила ряд заходів для посилення безпеки своєї мережі. Viasat удосконалило свої процедури безпеки для захисту своїх супутників від кібератак. Ці процедури включають в себе більш ретельну перевірку облікових даних і більш активний моніторинг мережі.

Протидія кіберзагрозам супутників

Однією з основних стратегій запобігання супутниковим кібератакам є шифрування. Шифрування – це процес, який перетворює дані на код для запобігання несанкціонованому доступу. Супутникові оператори використовують передові методи шифрування для захисту зв'язку між наземною станцією та супутником. Це гарантує, що навіть якщо хакеру вдасться перехопити сигнали, він не зможе розшифрувати інформацію без унікального ключа дешифрування.

Наземна атака може відбутися на етапі розробки супутника, наприклад, шляхом розміщення невидимого бекдору або шкідливого програмного забезпечення на борту перед запуском, або через наземну інфраструктуру під час роботи, наприклад, шляхом доступу до хмарних сервісів, наземних станцій або керування місією.

1.4 Стандарт вдосконаленого шифрування

Крім того, сучасні тенденції, такі як розширення супутникового зв'язку, дедалі складніше та потужніше бортове програмне забезпечення, використання сторонніх служб, збільшують площу атаки, відкриваючи більше вразливостей.

Щоб захистити супутниковий зв'язок та дані, більшість супутникових операторів використовують шифрування, зокрема стандарт AES (Advanced Encryption Standard), для шифрування висхідного та нисхідного каналу зв'язку. AES є стандартом безпеки з 2002 року і є надзвичайно стійким до атак методом грубої сили.

AES – це алгоритм симетричного шифрування, який використовує ключ для шифрування та дешифрування даних. Ключ розділяється між відправником і отримувачем, і його використовують для перемішування даних перед їхнім

надсиланням. Коли дані досягають пункту призначення, отримувач використовує той самий ключ, щоб розшифрувати дані та зробити їх читальними.

AES застосовується в супутниковому зв'язку, щоб гарантувати, що лише призначений одержувач може отримати доступ до даних. Також його використовують для захисту даних від перехоплення та прочитання сторонніми особами. Це особливо важливо для конфіденційних даних, таких як фінансова інформація або державні таємниці.

AES також використовується для захисту даних від зміни або спотворення під час передачі. Це особливо важливо для супутникового зв'язку, оскільки дані повинні переміщуватися через космос і можуть зазнавати впливу від інших сигналів. Шифруючи дані, значно важче будь-якому втручанню змінити дані під час передачі.

Використання шифрування AES у супутниковому зв'язку є критично важливим елементом забезпечення безпеки та надійності цих послуг. Завдяки AES постачальники супутникового зв'язку можуть гарантувати, що передані дані є захищеними та що лише призначений одержувач може отримати до них доступ. Це допомагає захистити конфіденційність користувачів і зберігати безпеку важливих даних.

Безпека даних супутників є надзвичайно важливою у сьогоднішньому цифровому світі. Оскільки все більше організацій використовують дані супутників для різних цілей, потреба в безпечному передаванні даних стає дедалі більш важливою. Одним із найефективніших способів забезпечення безпеки даних супутників є використання шифрування за стандартом Advanced Encryption Standard (AES).

Шифрування AES працює, перетворюючи звичайний текст на зашифрований текст за допомогою ключа. Потім зашифрований текст передається через безпечний канал, а одержувач використовує той самий ключ для розшифровки повідомлення.

Шифрування AES забезпечує низку переваг для безпеки даних супутників. По-перше, воно гарантує, що дані залишаються конфіденційними та захищеними від несанкційованого доступу. Шифрування AES також гарантує, що дані не будуть

пошкоджені або підроблені під час передачі. Крім того, шифрування AES є дуже ефективним і може використовуватися для швидкого та безпечного шифрування великих обсягів даних.

Окрім забезпечення безпечного способу передачі даних, шифрування AES також пропонує низку інших переваг. Наприклад, шифрування AES стійке до атак методом грубої сили, тобто хакерам важко зламати шифрування. Крім того, шифрування AES є дуже масштабованим, тобто його можна використовувати для шифрування великих обсягів даних без будь-якого зниження продуктивності.

Шифрування AES є важливим інструментом для забезпечення безпеки даних супутників. Воно забезпечує безпечний спосіб передачі даних і є дуже стійким до атак методом грубої сили. Крім того, шифрування AES є дуже ефективним і може використовуватися для швидкого та безпечного шифрування великих обсягів даних. З цих причин шифрування AES є безцінним інструментом для організацій, які покладаються на дані супутників для своєї роботи.

Стандарт AES - широко використовуваний алгоритм симетричного шифрування, який став де-факто стандартом для безпечного шифрування даних. Розроблений Національним інститутом стандартів і технологій (NIST) у 2001 році, AES використовується для захисту конфіденційної інформації, як-от номери кредитних карт, паролі та військові повідомлення.

AES - це блочний шифр, тобто він бере блок певного розміру відкритого тексту і перетворює його на блок шифрованого тексту такої ж довжини. Алгоритм базується на мережі підстановки-перестановки, яка використовує серію математичних операцій для перемішування даних. AES вважається одним із найбезпечніших алгоритмів шифрування, який використовують багато урядових та військових організацій для захисту своїх даних.

Вплив шифрування AES на супутникові технології був значним. Шифрування AES використовується для захисту даних, переданих через супутникові зв'язки, гарантуючи, що лише авторизовані користувачі можуть отримати доступ до даних.[6] Це особливо важливо для супутникового зв'язку, оскільки його часто використовують для передачі конфіденційної інформації, як-от фінансові дані або

військова розвідка. Шифрування AES також допомагає захиститися від підслуховування та інших форм перехоплення даних.

Крім того, шифрування AES використовується для захисту супутникових навігаційних систем, таких як GPS. Шифрування AES використовується для шифрування навігаційних сигналів, що надсилаються з супутників на GPS-приймачі, гарантуючи, що лише авторизовані користувачі можуть отримати доступ до навігаційних даних. Це особливо важливо для військових та урядових застосувань, оскільки допомагає запобігти несанкціонованому доступу до конфіденційних навігаційних даних.

Загалом, шифрування AES справило значний вплив на супутникові технології. Забезпечуючи безпечний спосіб захисту даних, переданих через супутникові зв'язки, AES допоміг гарантувати безпеку конфіденційної інформації та навігаційних даних.

AES вважається дуже безпечним алгоритмом шифрування, його використовує уряд США для захисту секретної інформації. Також його використовують військові, фінансові установи та інші організації для захисту конфіденційних даних.

AES базується на мережі підстановки-перестановки. Це тип криптосистеми, який використовує поєднання операцій підстановки та перестановки для перетворення відкритого тексту на зашифрований. Процес шифрування починається з розбиття відкритого тексту на блоки по 128 біт, а потім кожен блок піддається серії раундів. Кожен раунд складається з чотирьох етапів:

Етап підстановки: полягає в заміні кожного біта відкритого тексту іншим бітом на основі таблиці підстановки.

Етап перестановки: полягає в переупорядкуванні бітів відкритого тексту в певному порядку.

Етап змішування: полягає в змішуванні бітів відкритого тексту з бітами ключа.

Етап додавання ключа: полягає в додаванні бітів ключа до бітів відкритого тексту.

Процес дешифрування є оберненим процесом шифрування. Зашифрований текст розбивається на блоки по 128 біт, а потім кожен блок піддається серії раундів. Кожен раунд складається з чотирьох етапів:

- Етап зворотної підстановки.
- Етап зворотної перестановки.
- Етап зворотного змішування.
- Етап зворотного додавання ключа.

AES - потужний алгоритм шифрування, який використовується в супутниковому зв'язку для захисту даних від несанкціонованого доступу. Він базується на мережі підстановки-перестановки, що робить його дуже важким для зламу. Використовуючи AES, організації можуть гарантувати безпеку своїх даних і захищеність від зловмисників.

Дані супутників - цінний актив для бізнесу та організацій по всьому світу. Однак вони також вразливі до кібератак, які можуть призвести до витоку даних та інших проблем безпеки. Щоб захистити дані супутників від кіберзагроз, багато організацій звертаються до шифрування за стандартом AES (Advanced Encryption Standard).

AES - це симетричний алгоритм шифрування, який використовується для захисту конфіденційних даних. Він використовує ключ для шифрування та дешифрування даних, тобто лише ті, хто має ключ, можуть отримати доступ до даних. AES вважається одним із найбезпечніших алгоритмів шифрування, доступних на сьогодні, що робить його ідеальним вибором для захисту даних супутників.

Шифрування AES використовується для захисту даних супутників кількома способами. Наприклад, його можна використовувати для шифрування даних, перш ніж їх надсилають із супутника на наземну станцію. Це гарантує, що дані є безпечними під час передачі та доступ до них можуть отримати лише ті, хто має ключ.

Структура AES

Стандарт AES — це специфікація для шифрування електронних даних, встановлена Національним інститутом стандартів і технологій США (NIST) у 2001 році. AES широко використовується сьогодні, оскільки він значно потужніший за DES і потрійний DES, хоча його складніше реалізувати.

- AES — це блочний шифр.
- Розмір ключа може бути 128/192/256 біт.
- Шифрує дані блоками по 128 біт кожен.

Це означає, що він бере 128 біт як вхідні дані та виводить 128 біт зашифрованого тексту як вихідні дані.(рис. 1.7)

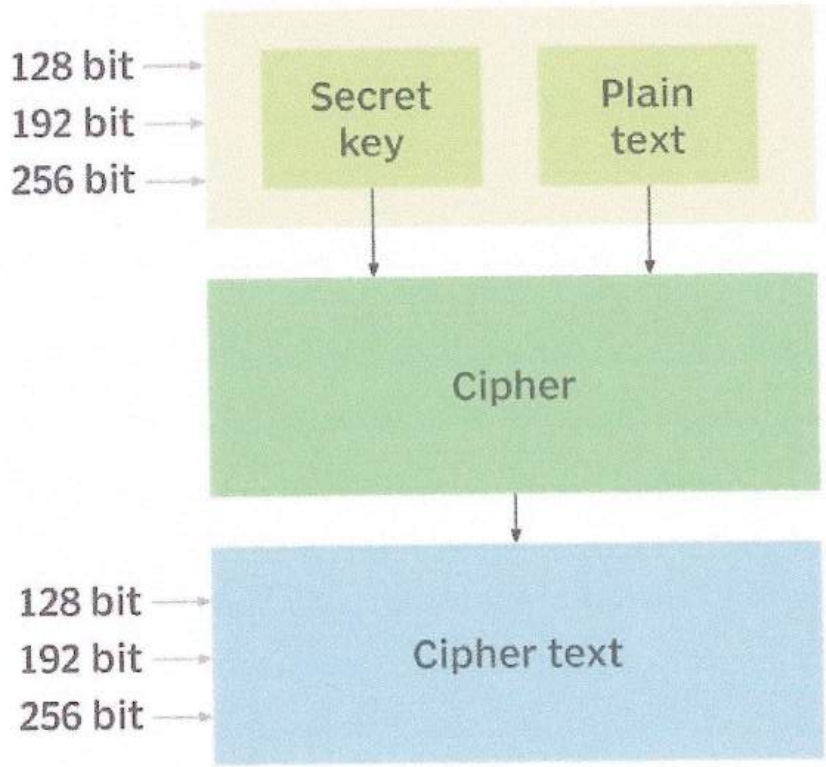


Рис.1.7 Структура AES

AES базується на принципі мережі підстановки-перестановки, тобто він виконується за допомогою серії пов'язаних операцій, які включають заміну та перемішування вхідних даних.

AES виконує операції над байтами даних, а не над бітами. Оскільки розмір блоку становить 128 біт, шифр обробляє 128 біт (або 16 байт) вхідних даних одночасно.

Кількість раундів залежить від довжини ключа:

- 128-бітний ключ — 10 раундів
- 192-бітний ключ — 12 раундів
- 256-бітний ключ — 14 раундів

Створення проміжних ключів

AES використовує спеціальний алгоритм підготовки ключів (Key Schedule algorithm), щоб обчислити всі проміжні ключі, які будуть використані в різних раундах шифрування. Іншими словами, вихідний ключ використовується для створення багатьох різних проміжних ключів, які будуть застосовуватися в відповідному раунді шифрування.

Цей алгоритм гарантує, що кожен раунд шифрування використовує унікальний ключ, що підвищує загальну безпеку алгоритму.(рис.1.8) Уявіть собі замок, де замість одного ключа використовується ціла гірлянда різних ключів, кожен з яких відкриває тільки одну з багатьох дверей. Навіть якщо хтось отримає один ключ, він не зможе відкрити всі двері та отримати доступ до всіх даних.

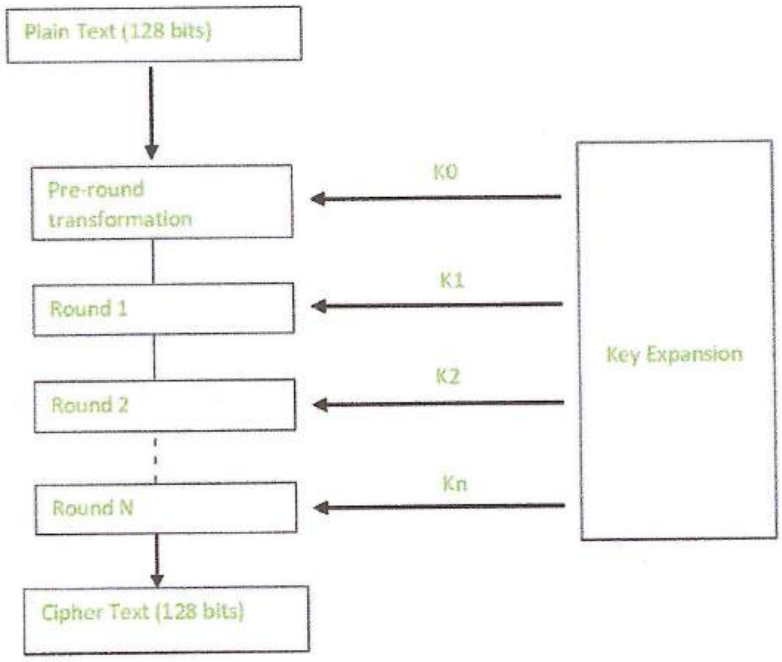


Рис.1.8 Створення проміжних раундів ключів

Алгоритм підготовки ключів виконує серію операцій над початковим ключем, щоб створити проміжні ключі.

Ці операції включають:

- Обертання бітів: Циклічне переміщення бітів ключа.
- Підстановки: Заміна окремих бітів ключа за допомогою спеціальної таблиці підстановки (S-box).
- Лінійні операції: Використання математичних функцій для змішування бітів ключа.

Кількість проміжних ключів, що створюються, залежить від довжини початкового ключа (128, 192, або 256 біт) і кількості раундів шифрування (10, 12, або 14 відповідно).

Завдяки цьому алгоритму підготовки ключів AES вважається дуже стійким до атак методом грубої сили, оскільки зловмиснику потрібно зламати не один, а цілу низку різних ключів.

Шифрування в AES

AES розглядає кожен блок як сітку з 16 байт (4 байти x 4 байти = 128) у вигляді розміщення за стовпцями. Це означає, що кожен байт блоку шифрування має свій власний рядок і стовпець.(рис.1.9)

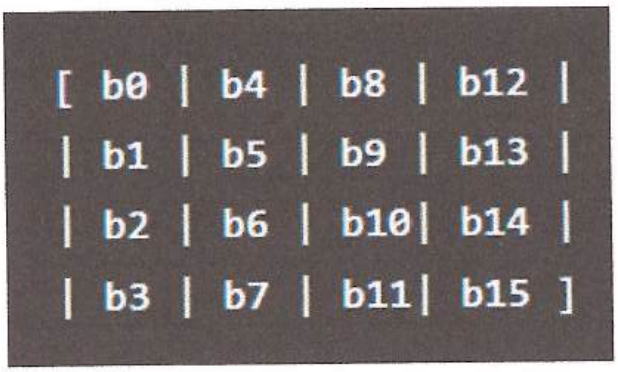


Рис.1.9 Блок даних у вигляді сітки

Процес шифрування AES складається з 10, 12 або 14 раундів, залежно від довжини ключа. У кожному раунді вхідні дані блоку шифрування піддаються серії математичних операцій, які включають:

- Заміну: Кожен байт вхідного блоку замінюється іншим байтом на основі таблиці підстановки.
- Перестановку: Порядок байт вхідного блоку перемішується за допомогою таблиці перестановки.

- Мішання: Вхідні дані блоку обробляються за допомогою операції xor з раундовим ключем.

Ось як виглядає загальний процес шифрування AES:

1. Вхідні дані: Алгоритм отримує блок даних для шифрування.
2. Розширений ключ: Алгоритм отримує раундовий ключ для поточного раунду.
3. Заміна: Кожен байт вхідного блоку замінюється іншим байтом на основі таблиці підстановки.
4. Перестановка: Порядок байт вхідного блоку перемішується за допомогою таблиці перестановки.
5. Мішання: Вхідні дані блоку обробляються за допомогою операції xor з раундовим ключем.
6. Повторити кроки 3-5 для наступних раундів.
7. Вихідні дані: Наприкінці процесу алгоритм виводить зашифрований блок даних.

Цей процес шифрування є надзвичайно складним і стійким до атак. Він робить AES одним із найбезпечніших алгоритмів шифрування, доступних сьогодні.

Остаточній раунд AES не містить етапу Мішання. Це цікава особливість алгоритму, яка забезпечує його ефективність і безпеку.

Симетричність: AES є симетричним алгоритмом, тобто використовує той самий ключ для шифрування та дешифрування. Відсутність MixColumns в остаточньому раунді спрощує процес дешифрування, оскільки зворотну операцію можна виконати без додаткового кроку.

Хоча MixColumns підвищує безпеку алгоритму, його відсутність в остаточньому раунді не робить AES вразливим. Інші етапи, такі як SubBytes і ShiftRows, забезпечують достатню перестановку та змішування даних для захисту від атак.

Ефективність: Виняток MixColumns з остаточного раунду покращує загальну ефективність алгоритму. Це зменшує кількість обчислень, необхідних для

шифрування та дешифрування, що може бути важливим для пристроїв з обмеженими ресурсами.

Підсумовуючи, відсутність MixColumns в остаточньому раунді AES є свідомим рішенням, яке балансує між безпекою, ефективністю та симетричністю алгоритму. Це робить AES потужним і універсальним інструментом для захисту даних.

Етап заміни в AES: перемішування даних за допомогою S-box

Цей етап реалізує заміну байтів. Кожен байт замінюється іншим байтом за допомогою таблиці підстановки, також відомої як S-box. Ця заміна виконується таким чином, що байт ніколи не замінюється самим собою і також не замінюється іншим байтом, який є доповненням поточного байта. Результатом цього етапу є матриця 16 байтів (4 x 4), як і раніше.

S-box: Це ядро етапу заміни. S-box є таблицею розміром 256 x 8, яка містить значення для кожного можливого байта вхідних даних. Значення S-box визначають, який байт потрібно використовувати для заміни поточного байта.

Немає самозаміни: Байти ніколи не замінюються самими собою. Це гарантує, що вхідний і вихідний блоки не можуть бути однаковими, що підвищує безпеку.

Без доповнення: Байти не замінюються своїми доповненнями. Це запобігає простим атакам, які намагаються зламати шифр, використовуючи властивості бітових операцій.

Завдяки S-box і обмеженням на самозаміну та доповнення, цей етап ефективно перемішує дані блоку, роблячи їх більш складними для дешифрування без правильного ключа.

Зсув рядків: перемішування даних в AES

Так, етап ShiftRows дійсно працює саме так, як і його назва. Кожен рядок матриці даних зсувається на певну кількість позицій. Це один з етапів перемішування даних в AES, який додає ще один рівень складності до алгоритму шифрування.

Ось як відбувається зсув рядків(рис.1.11)

```
[ b0  | b1  | b2  | b3  ]          [ b0  | b1  | b2  | b3  ]
| b4  | b5  | b6  | b7  |    ->    | b5  | b6  | b7  | b4  |
| b8  | b9  | b10 | b11 |          | b10 | b11 | b8  | b9  |
[ b12 | b13 | b14 | b15 ]          [ b15 | b12 | b13 | b14 ]
```

Рис.1.10 Зсув рядків AES

Перший рядок – Залишається незмінним. Другий рядок – Зсувається на один байт вліво. Це означає, що перший байт стає другим, другий стає третім, і так далі, останній байт стає першим. Третій рядок – Зсувається на два байти вліво. Аналогічно до другого рядка, кожен байт рухається на дві позиції вперед. Четвертий рядок – сується на три байти вліво. Кожен байт рухається на три позиції вперед.

Цей зсув створює ефект коли байти з різних стовпців змішуються між собою, що ускладнює визначення їхнього початкового розташування без правильного ключа.

MixColumns (Змішування стовпців)

Цей крок фактично є множенням матриці. Кожен стовпець множиться на конкретну матрицю, і, таким чином, позиція кожного байта у стовпці змінюється як результат.(рис 1.11)

Цей крок пропускається в останньому раунді.

```
[ c0 ]       [ 2  3  1  1 ] [ b0 ]
| c1 |  =     | 1  2  3  1 |    | b1 |
| c2 |       | 1  1  2  3 |    | b2 |
[ c3 ]       [ 3  1  1  2 ]    [ b3 ]
```

Рис. 1.11 Змішування стовпців

Додавання Ключів Раунду

Тепер отриманий вихід попереднього етапу використовується для операції XOR з відповідним ключем раунду. Тут 16 байтів розглядаються не як сітка, а просто як 128 біт даних.(рис 1.12)

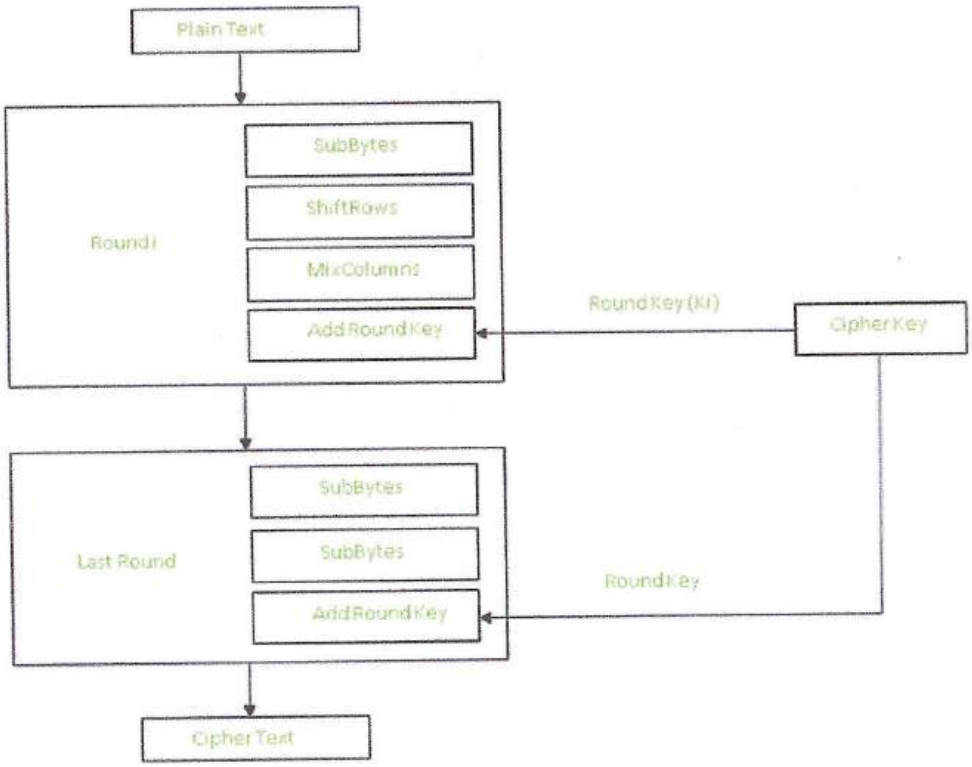


Рис.1.12 Додавання ключів раунду

Після усіх цих раундів 128 біт зашифрованих даних повертаються як вихід. Цей процес повторюється до тих пір, поки всі дані, які мають бути зашифровані, проходять цей процес.

Розшифрування

Етапи раундів можна легко скасувати, оскільки у них є протилежні операції, які, виконані, скасовують зміни. Кожні 128 блоків проходять 10, 12 або 14 раундів в залежності від розміру ключа.

Етапи кожного раунду при розшифруванні наступні:

- Додати ключ раунду (Add round key)
- Обернена операція MixColumns (Inverse MixColumns)
- Зсув рядків (ShiftRows)
- Обернена операція SubByte (Inverse SubByte)

Процес розшифрування - це процес шифрування, виконаний у зворотному порядку, тому я пояснюватиму кроки з відзначеними відмінностями.

Обернене змішування стовпців

Цей крок схожий на крок змішування стовпців у шифруванні, але відрізняється матрицею, яка використовується для виконання операції.(рис.1.13)



Рис.1.13 Обернене змішування стовпців

Обернена SubBytes

Використовується обернена S-блок як таблиця пошуку, за допомогою якої відбувається заміна байтів під час розшифрування.

Застосування

AES широко використовується в багатьох областях, які вимагають безпечного зберігання та передачі даних. Деякі загальні випадки використання включають:

Бездротова безпека: AES використовується для захисту бездротових мереж, таких як Wi-Fi, для забезпечення конфіденційності даних і запобігання несанкціонованому доступу.

Шифрування баз даних: AES можна застосовувати для шифрування чутливої інформації, збереженої в базах даних. Це допомагає захищати особисту інформацію, фінансові записи та інші конфіденційні дані в разі витоку даних.

Безпечні комунікації: AES широко використовується в протоколах для Інтернет-комунікацій, електронної пошти, миттєвих повідомлень та голосового/відео виклику. Він забезпечує конфіденційність даних.

Зберігання даних: AES використовується для шифрування чутливих даних, збережених на жорстких дисках, USB-накопичувачах та інших засобах зберігання, що захищає їх від несанкціонованого доступу в разі втрати чи крадіжки.

Віртуальні приватні мережі (VPN): AES часто використовується в протоколах VPN для захисту комунікації між пристроєм користувача та віддаленим сервером.

Він гарантує, що дані, відправлені та отримані через VPN, залишаються приватними і не можуть бути розшифровані слідопитами.

Безпечне зберігання паролів: Шифрування AES часто використовується для безпечного зберігання паролів. Замість зберігання паролів у відкритому вигляді, зберігається зашифрована версія, що додає додатковий рівень безпеки та захищає дані користувача в разі несанкціонованого доступу до зберігання.

Шифрування файлів та дисків: AES використовується для шифрування файлів та тек на комп'ютерах, зовнішніх пристроях зберігання та хмаровому сховищі. Це захищає чутливі дані, збережені на пристроях чи під час передачі даних, щоб запобігти несанкціонованому доступу.

Крім того, AES можна використовувати для шифрування даних, що зберігаються на самому супутнику, що запобігає несанкційованому доступу.

Крім захисту даних супутників від кіберзагроз, шифрування AES також можна використовувати для захисту від фізичних загроз. Наприклад, AES можна використовувати для шифрування даних, що зберігаються на жорсткому диску супутника, що запобігає фізичному доступу до даних. Це особливо важливо для супутників, які розташовані в віддалених місцях, оскільки вони більш вразливі до фізичних атак.

Загалом, шифрування AES - це ефективний інструмент для захисту даних супутників від кіберзагроз. Він є безпечним, надійним і його можна використовувати для захисту даних різними способами. Використовуючи AES-шифрування, організації можуть гарантувати, що їхні дані супутників є безпечними та захищеними від кібератак.

Інструкційний набір AES тепер інтегрований у процесор (забезпечує пропускну здатність кілька гігабайтів в секунду), щоб поліпшити швидкість та безпеку застосунків, які використовують AES для шифрування та розшифрування. Навіть після 20 років з моменту його введення нам не вдалося взламати алгоритм AES, оскільки це недосяжно навіть за сучасних технологій. До цього часу єдиним вразливим місцем залишається реалізація алгоритму.

Однак самого AES недостатньо для забезпечення безпечного зв'язку для малих супутників. Перетворення звичайного тексту на шифр під час шифрування потребує алгоритму та криптографічного ключа (також відомого як секрет).

Оскільки AES є симетричним алгоритмом, ключ, який використовується для шифрування телекоманди на Землі, буде тим самим ключем, який використовується для його дешифрування на борту. Якщо ключ будь-яким чином скомпрометований, тоді скомпрометована також інформація, що передається.

Симетричне шифрування демонструє ще більші обмеження, коли багато супутників є частиною єдиного сузір'я. Неможливо уявити використання одного секрету для всіх них. Компрометація одного призведе до компрометації всього флоту.

В результаті рівень захисту, що забезпечується алгоритмом шифрування, безпосередньо пов'язаний з рівнем довіри до секретів, що використовуються цим алгоритмом, незалежно від його складності.

Секрети можуть бути скомпрометовані, наприклад, шляхом використання слабких методів їх генерації, поганих практик управління на Землі, таких як обмін секретами через незахищені служби електронної пошти, або недооцінки ризику атаки після їх введення на борт, але до того, як супутник запущений (наприклад, під час транспортування або на майданчику запуску).

1.5 Корені довіри та конфіденційні обчислення

Як зазначено вище, важливо мати можливість довіряти криптографічним операціям, ключам і, більш загально, будь-яким програмам, які вважаються критично важливими для місії, які використовуються на Землі та на борту. Основа цієї довіри полягає в концепції кореня довіри (RoT – root of trust).

RoT відноситься до середовища, де генеруються та зберігаються секрети, як правило, на наземних серверах або в хмарній інфраструктурі Центру управління польотами (MCC) або в бортовому комп'ютері (OBC). RoT складається з високо

надійних компонентів апаратного, мікропрограмного та програмного забезпечення, які призначені для виконання цих дуже специфічних критичних функцій безпеки.

RoT забезпечують міцну основу для побудови безпеки та довіри, але простого існування RoT недостатньо. Зловмисники також можуть використовувати логіку, пов'язану з апаратним і програмним забезпеченням, які використовують секрети, щоб отримати доступ до конфіденційних даних, що зберігаються в пам'яті. Захист цієї логіки вимагає додаткового рівня безпеки, відомого як конфіденційні обчислення.

Конфіденційні обчислення представляють собою наступний кордон кібербезпеки для малих супутників, дозволяючи операторам обробляти зашифровані дані в пам'яті для захисту даних під час їх використання.

Прикладом конфіденційних обчислень у дії є довірене середовище виконання (TEE). TEE – це середовище, яке забезпечує рівень гарантії цілісності даних, конфіденційності даних та цілісності коду, а також запобігає несанкціонованому доступу до даних та логіки.

До цих суб'єктів належать інші програми на хості, операційна система хоста та гіпервізор, системні адміністратори, постачальники послуг, власник інфраструктури або будь-хто інший, хто має фізичний доступ до апаратного або програмного забезпечення.

TEE набувають поширення на наземних ринках і є способом захисту космічної інфраструктури – як на Землі для захисту програмного забезпечення управління місією (MCS) та секретів, що використовуються для зв'язку з супутником, так і на борту для забезпечення безпечного середовища для програмного забезпечення, що виконується в ОВС.

Архітектура безпеки від кінця до кінця

Хоча атаки під час польоту можуть бути лише наземними, тим не менш, надзвичайно важливо довіряти бортовим секретам, програмному та апаратному забезпеченню, які супутник використовуватиме для збору, обробки та передачі конфіденційної інформації на Землю, зберігаючи при цьому їх конфіденційність, цілісність та доступність.

Для цього потрібен комплексний підхід до безпечного зберігання криптографічних секретів, що дозволяють здійснювати космічні операції на рівнях платформи та корисного навантаження. Архітектура безпеки від кінця до кінця передбачає механізми безпеки як на Землі, так і на борту, які можуть виконувати такі операції(рис 1.14)

- шифрувати/дешифрувати висхідний канал зв'язку,
- шифрувати/дешифрувати спадний канал зв'язку,
- аутентифікувати супутники та наземні сегменти,
- створювати/перевіряти підпис оновлень програмного забезпечення для переналаштування та
- захищати життєвий цикл управління ключами.

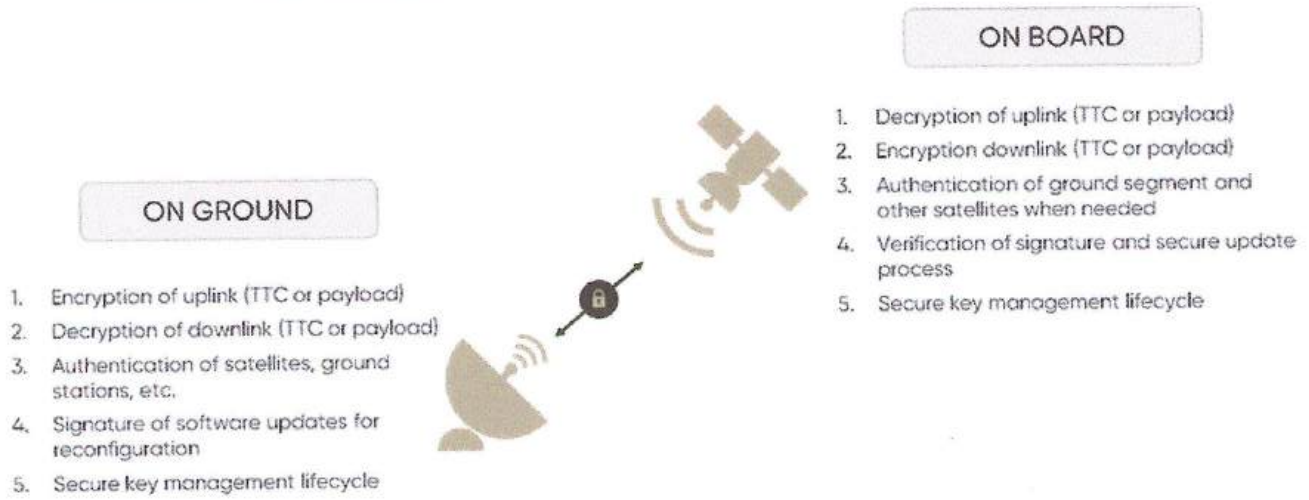


Рис.1.14 Архітектура безпеки від кінця до кінця

Хоча існує кілька відомих механізмів захисту наземного сегмента, розробка для космічного сегмента відстає.

CYSEC працювала над тим, щоб закрити цю прогалину за допомогою свого пакету конфіденційних обчислень, розробленого для комерційних космічних місій. ARCA на Землі, довірене середовище виконання з апаратним криптографічним бекендом та надійною операційною системою, захищає криптографічні секрети та програми, такі як MCS.

ARCASPACE надає бортовий RoT для зберігання криптографічних секретів та виконання криптографічних операцій та управління ключами. Результатом є

безпечний канал зв'язку від кінця до кінця між Землею та супутником, що забезпечує конфіденційність, цілісність та доступність переданих даних

2 АРХІТЕКТУРНА ОПТИМІЗАЦІЯ ПАРАЛЕЛЬНОГО АЛГОРИТМУ АУТЕНТИФІКОВАНОГО ШИФРУВАННЯ ДЛЯ СУПУТНИКОВИХ ЗАСТОСУВАНЬ

2.1 Алгоритми шифрування

Високошвидкісний обмін даними стає важливим для багатьох застосувань, включаючи супутниковий зв'язок. Алгоритми безпеки, що використовуються для передачі інформації, також повинні мати високу швидкість для того, щоб впоратися зі швидкістю передачі. Крім того, алгоритми аутентифікованого шифрування (AE) забезпечують високошвидкісний зв'язок та служби безпеки, включаючи шифрування даних, автентифікацію та цілісність. Алгоритми AE доступні з послідовною та паралельною архітектурою; серед них алгоритм Galois/Counter Mode (GCM) має паралельну архітектуру. Режим синтетичного початкового вектора ініціалізації (SIV) в алгоритмі AES-GCM-SIV забезпечує захист від неправильного використання нонсу за допомогою алгоритму GCM. Крім того, алгоритм AES-GCM-SIV забезпечує знижену пропускну здатність даних порівняно з алгоритмом AES-GCM. У цій роботі представлено паралельний алгоритм з переналаштуванням ключів та рандомізацією вектора ініціалізації для забезпечення високої пропускної здатності даних, захисту від неправильного використання нонсу та захисту від атак з побічними каналами. Реалізація запропонованого алгоритму виконується на FPGA і порівнюється з FPGA-реалізаціями AES-GCM, AES-GCM-SIV та нещодавно введених алгоритмів. Оптимізацію запропонованого алгоритму та аналіз безпеки для космічного застосування представлено за допомогою різних оптимізацій та комбінації оптимізацій.

Високошвидкісні комунікаційні системи необхідні через високі обчислювальні вимоги для надання розширених можливостей споживачам цифрових систем. Застосування високошвидкісного обладнання варіюється від системи стільникового зв'язку, включаючи використання технології зв'язку 5G, до супутникового зв'язку з розширеними послугами для багатьох застосувань,

включаючи послуги зображення, навіть з геостаціонарного супутника. Обсяг даних, необхідних для різних застосувань, що використовують супутник, збільшився, оскільки супутникові дані використовуються майже у всіх типах застосувань, включаючи управління надзвичайними ситуаціями, контроль повітряного руху [7] та багато інших застосувань. Основною причиною збільшення використання комунікаційних послуг, що пропонуються супутниками, є пропускна здатність даних, вбудоване зберігання даних та кількість каналів зв'язку.

Проблеми для комунікаційних систем однаковою мірою стосуються алгоритмів безпеки даних, пов'язаних з ними, для забезпечення безпечного зв'язку. Алгоритми безпеки даних надають три основні послуги, такі як конфіденційність, автентичність і цілісність даних. Використання цих послуг пов'язано з типом алгоритму, що використовується в додатку; наприклад, алгоритми шифрування даних забезпечують послугу конфіденційності. Алгоритми автентифікації даних надають послуги автентифікації та цілісності. Алгоритми аутентифікованого шифрування (AE) надають послуги конфіденційності, автентичність і цілісності. Консультативний комітет з питань космічних даних (CCSDS) та Європейське космічне агентство (ESA) рекомендують використовувати ці послуги відповідно до вимог місії. [8] Збільшення використання даних з супутників вимагає використання алгоритму безпеки даних, який може забезпечити високий рівень безпеки та надійності даних. Алгоритми AE надають усі три основні послуги, необхідні для гарантування високої безпеки та надійності даних. Крім того, залишається складною проблема забезпечення високої швидкості комунікаційних систем, що пов'язана з вибором високошвидкісних алгоритмів безпеки. Раніше для забезпечення безпеки даних багатьох застосувань використовувалися AE-алгоритми з послідовною архітектурою, такі як Advanced Encryption Standard in Counter mode (AES-CTR) з алгоритмом шифру блоку ланцюжка повідомлень (AES-CCM). Алгоритм AES-CCM використовує алгоритм AES-CTR з алгоритмом автентифікації на основі коду автентифікації повідомлень (MAC). Хоча алгоритм CCM має обмежену пропускну здатність даних і не є рекомендованим варіантом для застосувань з високою пропускною здатністю даних, таких як супутниковий

зв'язок. Також все частіше використовуються AE-алгоритми з паралельною архітектурою, такі як алгоритм Galois Counter Mode (GCM). CCSDS рекомендує алгоритм GCM для супутникових застосувань. Паралельна архітектура алгоритму GCM забезпечує перевагу високої пропускної здатності даних. Реалізація та розуміння дизайну стає простішим, оскільки він використовує сучасний алгоритм AES-CTR з автентифікацією на основі хешування. Відомі алгоритми AE, такі як AES-CCM та AES-GCM, використовують алгоритм AES-CTR.

В алгоритмі AES-CTR вектор ініціалізації (IV) складається з нонсу, вектора констант і вектора лічильника. Крім того, повторення IV може призвести до компрометації безпеки алгоритму; тому IV збільшується на одиницю для генерації кожного шифротексту. Хоча вектор нонсу залишається тим самим, а інкремент відбувається лише у векторі лічильника. Останнім часом безпека алгоритму GCM стала сумнівною через атаки, засновані на тому, що IV має постійний нонс.[9] Якщо постійний нонс використовується для двох повідомлень, може статися витік звичайного тексту. Використання виключної операції OR звичайного тексту з ключовим потоком для генерації шифротексту з постійним нонсом створить проблеми безпеки неправильного використання нонсу.

Раніше для генерації синтетичного вектора ініціалізації (SIV) в алгоритмі AE використовувався алгоритм аутентифікації повідомлень на основі шифру (CMAC). Де алгоритм CMAC використовувався для автентифікації додаткових даних автентифікації (AAD), а алгоритм AES-CTR використовувався для шифрування та генерації тегів. Крім того, в алгоритмі AES-GCM-SIV для реалізації стійкості до помилок використання нонсу пропонується оптимізація алгоритму AES-GCM. Реалізація забезпечує захист від помилок використання нонсу за рахунок зниження пропускної здатності даних. Нещодавно дослідник порівняв реалізації алгоритмів AES-GCM та AES-GCM-SIV на FPGA.

У 2012 році був проведений конкурс під назвою Competition for Authenticated Encryption: Security, Applicability, and Reliability (CAESAR) для вибору алгоритмів AE для протистояння майбутнім викликам. Конкурс CAESAR включає алгоритми для вирішення проблеми помилкового використання нонсу та вибору нових

алгоритмів для сучасного високошвидкісного зв'язку. Після оцінки алгоритмів, представлених на конкурсі, було рекомендовано загалом шість алгоритмів для AE даних для різних застосувань. Пропоновані алгоритми в конкурсах CAESAR використовуються для різних застосувань, а також проводилися порівняльні дослідження для різних застосувань, включаючи Інтернет речей (IoT).

Крім того, різні дослідники запропонували слабкі сторони алгоритмів, запропонованих для CAESAR. Деякі з запропонованих схем в CAESAR мають кращу обчислювальну швидкість порівняно з алгоритмом AES-GCM, але все ж таки деякі схеми відстають у різних аспектах, наприклад, вони не є стійкими до помилок використання нонсу. Деякі з фіналістів конкурсу CAESAR повністю, а деякі частково стійкі до помилок використання нонсу. Дослідження щодо реалізації різних запропонованих алгоритмів у конкурсі CAESAR показало, що існує компроміс між ефективністю та використанням ресурсів різних алгоритмів для досягнення захисту від помилок використання нонсу. Наприклад, алгоритм PRIMATE APE має меншу площу, але він має нижчу пропускну здатність даних порівняно з іншими алгоритмами, стійкими до помилок використання нонсу.

Крім атак помилкового використання нонсу, пасивні атаки, такі як атаки з побічним каналом, стають предметом занепокоєння дослідників для забезпечення бездоганної безпеки алгоритмів AE. Атаки з побічним каналом базуються на моніторингу фізичних параметрів (таких як вимірювання струму, електромагнітного випромінювання тощо). Атаки з побічним каналом більш помітні на алгоритмах AE через використання одного ключа в алгоритмі безпеки. Багато блочних шифрів, запропонованих у конкурсі CAESAR, використовують один ключ. Використання одного ключа багатьма запропонованими алгоритмами в CAESAR зробило їх вразливими до атак з побічним каналом[10] та атак диференціального аналізу потужності (DPA). Ці атаки пов'язані з фізичним контактом з обладнанням для аналізу випромінюваної потужності. Цей вид атаки відстежує розсіювання потужності або використання електромагнітного поля пристрою для відновлення секретного ключа.

Нещодавно дослідник показав, що багато алгоритмів, таких як AES-GCM, Ascon, ACRON, CLOC, JAMBU, SILC та Ketje Jr., зазнають впливу DPA під час реалізації цих алгоритмів на FPGA.[11] Тому необхідно прийняти контрзаходи для протидії впливу цих атак. Методи контрзаходів включають маскування алгоритму шифрування та алгоритм переключення ключів. Процес маскування полягає в приховуванні алгоритму, але він додає додаткове навантаження на алгоритм. Процес переключення ключів полягає в генеруванні нового ключа сеансу для досягнення внутрішнього захисту від цих атак. Perfect Forward Secrecy (PFS) використовує підхід переключення ключів, де для кожного повідомлення використовується новий ключ. Концепції переключення ключів і прямої секретності швидко застосовуються в різних областях застосування. У випадку витоку безпеки лише одне повідомлення буде скомпрометовано.

З іншого боку, космічне середовище має негативний вплив на електронні пристрої порівняно з земним середовищем. Космічне середовище містить високий тиск, вакуум та випромінювання, які створюють виняткові умови експлуатації для електронної апаратури. Тому необхідно приймати спеціальні заходи для захисту електронних пристроїв для протистояння космічному середовищу. Крім того, для безпомилкової роботи алгоритмів у космічному середовищі потрібна оптимізація алгоритмів, реалізованих на пристрої. Особливо алгоритми безпеки, такі як алгоритми AE, повинні бути оптимізовані для роботи в космосі для забезпечення надійного та безпечного зв'язку. Випромінювання в космосі впливає на пам'ять в електронній апаратурі; ці ефекти називаються Single Event Effects (SEE)[12]. Ці ефекти змушують алгоритм працювати неправильно. Крім того, існує багато методів пом'якшення помилок SEE, але модифікація архітектури для оптимізації її для алгоритмів AE допомагає зменшити SEE без створення додаткового навантаження. Дослідники запропонували, що проектування алгоритму AE шляхом уникнення використання пам'яті, такої як статична оперативна пам'ять (SRAM), може допомогти зменшити SEE. На щастя, дизайн алгоритмів безпеки (таких як алгоритм AES) використовує меншу кількість пам'яті. Особливо в алгоритмі AES пам'ять використовується для зберігання таблиці ящика підстановки (S-box).

Реалізація алгоритму, такої як S-box замість використання таблиці, може допомогти уникнути використання пам'яті. Тому вимоги до високошвидкісного та безпечного алгоритму AE мають такі конкретні проблеми.

Алгоритм повинен бути алгоритмом паралельної архітектури для забезпечення високої пропускної здатності даних.

Алгоритм повинен бути стійким до неправильного використання нонсу для забезпечення безпечного зв'язку.

Алгоритм повинен бути стійким до атак з побічними каналами, такими як DPA, для забезпечення безпечного зв'язку.

Для забезпечення роботи алгоритмів безпеки в космосі слід реалізувати архітектурну оптимізацію для космічного середовища.

Поточні алгоритми не відповідають викликам, згаданим вище. У випадку, якщо деякий алгоритм відповідає вимогам високої пропускної здатності (наприклад, AES-GCM); вони не забезпечують стійкості до зловживання випадковими числами. З іншого боку, якщо алгоритм забезпечує стійкість до зловживання випадковими числами (наприклад, AES-GCM-SIV); він не має високої пропускної здатності даних. Крім того, вимоги до повторного ключового слова або маскування для усунення атак побічного каналу не реалізовані в широко використовуваних алгоритмах, таких як алгоритм GCM. Аналогічно, оптимізації архітектури потрібні для оптимізації алгоритму для багатьох додатків, включаючи супутникове застосування. Тому в цій роботі запропоновано паралельну архітектуру AE-алгоритму для забезпечення високої пропускної здатності даних разом із захистом від зловживання випадковими числами, стійкістю до атак побічного каналу та оптимізацією для супутникового застосування за допомогою алгоритму AES-CTR.

2.2 Алгоритми аутентифікованого шифрування

Алгоритми AE досліджувались та розроблялися протягом певного часу. Зростаючі вимоги до високошвидкісного зв'язку та зростаюча кількість атак на безпеку викликають у дослідників занепокоєння щодо розробки нових алгоритмів з кращими характеристиками та охопленням безпеки. Алгоритм AES-GCM та його вдосконалення для захисту від невірного використання випадкових чисел в

алгоритмі AES-GCM-SIV демонструють еволюцію алгоритмів AE через ефективність атак на алгоритми безпеки. Крім того, багато дослідників запропонували оптимізацію алгоритму AES-GCM за рахунок ефективного використання площі та підвищення пропускної здатності.[13]

Тим часом, для виконання вимог до високої пропускної здатності сучасних обчислень був введений алгоритм автентифікаційного коду повідомлень на базі паралельного шифру (PCMAC). Алгоритми PCMAC забезпечують пропускну здатність, порівнянну з пропускною здатністю алгоритму AES-GCM, хоча він використовує автентифікацію на основі MAC замість автентифікації на основі хешу. Також запропоновано алгоритм PCMAC Synthetic Initialization Vector (PCMAC-SIV) для реалізації стійкості до зловживання випадковими числами. Алгоритм PCMAC-SIV забезпечує пропускну здатність, порівнянну з пропускною здатністю алгоритму AES-GCM-SIV.

Запропоновані алгоритми AE в конкурсі CEASAR також запропонували алгоритми, стійкі до зловживання випадковими числами, і навіть деякі алгоритми забезпечують кращу пропускну здатність, ніж алгоритм AES-GCM. Алгоритм POET (Pipeline able On-line Encryption with the authentication Tag) запропонував алгоритм AE, що використовує чотири раунди алгоритму AES. Алгоритм POET є легким і підходить для Інтернету речей (IoT) та багатьох інших додатків. Алгоритм POET реалізовано паралельно і має властивість стійкості до зловживання випадковими числами. Алгоритм POET має низькі вимоги до ресурсів та паралельну реалізацію з меншою кількістю тактових циклів, що дозволило йому забезпечувати високу пропускну здатність даних.

Останнім часом, зі зростаючим попитом на алгоритми AE для забезпечення високої пропускної здатності даних та захисту від зловживання випадковими числами, багато дослідників провели порівняння між кандидатами конкурсу CEASAR. Порівняльна реалізація в основному виконується за допомогою Польової програмованої вентильної матриці (FPGA) на апаратній платформі, використовуючи її можливості паралельної реалізації. Порівняння реалізації алгоритму Deoxys проводиться з алгоритмом AES-GCM за допомогою апаратної

платформи FPGA [14], і багато інших алгоритмів проводили подібні порівняння. Тому в цій роботі апаратна платформа FPGA вибрана для реалізації запропонованого алгоритму для забезпечення високої пропускної здатності даних, реалізації стійкості до зловживання випадковими числами та алгоритму, стійкого до атак побічного каналу.

Атака зловживання випадковими числами обумовлена використанням постійного випадкового числа в IV алгоритмів блокового шифру, таких як алгоритм AES-CTR. Деякі автори запропонували свої методики модифікації IV для генерації випадковості в IV. Недавно один із таких методів запропонував використання генератора GEFFE зі збільшенням IV на одиницю, що використовується для супутникових зображень для випадковості шифротексту. Аналогічним чином дослідники запропонували використовувати операцію зсуву вправо та збільшення для випадковості IV [15]. Згенерований шифротекст був протестований для тесту NIST (Національного інституту стандартів і технологій) та частотного тесту для перевірки випадковості згенерованого шифротексту.

Також атаки побічного каналу показали свою ефективність на різних алгоритмах, включаючи алгоритм AES. Також алгоритми AE, що базуються на блочному шифрі, такі як алгоритми AES-GCM, схильні до атак побічного каналу. Нещодавно атака DPA вплинула на багато алгоритмів AE, включаючи алгоритм AES-GCM. Алгоритм переключення ключів може захистити алгоритми AE від атак DPA. Алгоритм переключення ключів передбачає використання різного ключа для кожного сеансу для генерації шифротексту. У цій роботі запропонований алгоритм пропонує алгоритм випадковості IV для методу стійкості до зловживання випадковими числами та алгоритм переключення ключів для захисту даних від атак побічного каналу, таких як атака DPA.

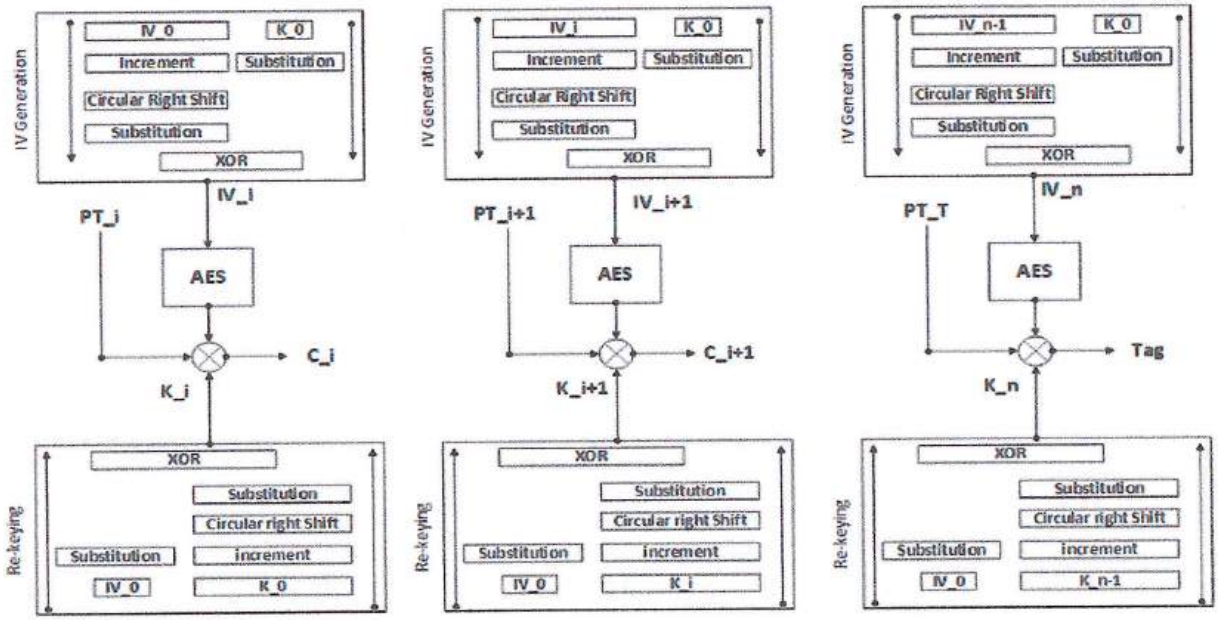


Рис. 2.1 Схема запропонованого паралельного, стійкого до неправильного використання алгоритму AE

Багато дослідників запропонували архітектурні оптимізації для реалізації алгоритму AES-GCM на апаратній платформі FPGA. Також нові алгоритми, запропоновані на конкурсі CEASAR, були реалізовані на FPGA з використанням деяких запропонованих архітектурних оптимізацій [16]. Архітектурні оптимізації для супутникових додатків є більш-менш схожими до оптимізацій, що виконуються на наземних системах. Крім того, є деякі екстремальні умови в космічному середовищі, які змусили провести додаткову архітектурну оптимізацію для супутників. Ці екстремальні умови – це випромінювання в космосі через пояс радіації Ван Аллена та сонячні спалахи. Багато дослідників запропонували різні оптимізації для реалізації, щоб зменшити вплив випромінювання на електронне обладнання [17]. Використання цього алгоритму для супутникових додатків змусило нас запропонувати додаткові архітектурні оптимізації для запропонованого алгоритму AE.

2.3 Запропонований алгоритм

Запропонований алгоритм складається з трьох частин: алгоритму генерації IV, блочного шифру та алгоритму переключення ключів. Алгоритм генерації IV рандомізує IV для забезпечення захисту від атаки зловживання випадковими числами. Друга частина використовує алгоритм блокового шифру, будь-який

алгоритм блокового шифру може бути використаний, але ми вибрали алгоритм AES-CTR. Третя частина - це алгоритм переключення ключів, який захищає від атак побічного каналу, таких як атака DPA. Алгоритм переключення ключів надає окремий ключ для кожного сеансу для генерації шифротексту, що робить атаки побічного каналу неефективними проти запропонованого алгоритму. Блок-схема запропонованого алгоритму наведена на рисунку 2.1. Запропонований алгоритм може використовувати ключ будь-якого розміру, наприклад 128-біт, 192-біт та 256-біт. Крім того, у цій роботі реалізовано розмір ключа 128 біт. PT_T позначає виключне-або всіх відкритих текстів, виключне-або-добуток відкритих текстів буде використано для генерації тегу автентифікації. IV та секретний ключ попередньо розподіляються між користувачами для дешифрування та автентифікації даних.

Алгоритм IV генерації

Захист від зловживання випадковими числами можна виконати шляхом модифікації випадкового числа для кожного відкритого тексту, який залишався постійним у багатьох відомих алгоритмах AE, таких як алгоритм GCM. Також багато дослідників запропонували модифікацію IV для її рандомізації. Ступінь цієї випадковості було перевірено за допомогою тестів запуску та частотності NIST. У запропонованому алгоритмі AE ми пропонуємо схожий підхід до попередньої обробки IV для рандомізації IV. IV генерується за допомогою випадково згенерованого секретного IV. Крім того, можна також використовувати загальний IV, що складається з постійного вектора, випадкового числа та векторного лічильника. До IV застосовуються три процеси: збільшення IV на одиницю, циклічний правий зсув IV і заміщення значення IV за допомогою блоку підстановки (S-Box). S-Box, який використовується для генерації IV, такий самий, як і той, що використовується в алгоритмі AES. IV спочатку збільшується на одиницю для кожного відкритого тексту. Потім збільшене IV циклічно праворуч зміщується для обертання IV. Зміщене IV потім замінюється за допомогою S-Box. Нарешті, ключ, згенерований з алгоритму переключення ключів, буде використано в алгоритмі генерації IV. Ключ замінюється за допомогою алгоритму S-Box, а потім він ексклюзивно-або з заміщеним IV для генерації нового IV. Нове згенероване IV

(тобто IV1) подається на вхід алгоритму генерації IV для генерації наступного шифротексту. На відміну від цього, початковий секретний ключ для переключення ключів буде вводитися для генерації IV для генерації всіх шифротекстів. Поток алгоритму генерації IV показано на рисунку 2.2.

Алгоритм блокового шифру

Шифрування відкритого тексту виконується за допомогою алгоритмів блокового шифру; у запропонованому алгоритмі для шифрування відкритого тексту використовується AES-CTR. Алгоритм AES-CTR має унікальну властивість використання архітектури паралельної реалізації.

Паралельна архітектура є причиною її використання в багатьох алгоритмах AE з високою пропускною здатністю даних. Тому ми також використовували алгоритм AES-CTR для шифрування відкритого тексту в запропонованому алгоритмі. Запропонований алгоритм використовує IV, секретний ключ і відкритий текст (PT) як вхід і генерує шифротекст і тег автентифікації (Tag) як вихід. Секретний ключ використовується як вхід для алгоритму AES-CTR, а також у алгоритмах генерації IV і переключення ключів.

Алгоритм перемикання ключів

Атаки побічного каналу є дуже помітними атаками, що впливають на алгоритми AE. Захист від атак побічного каналу став необхідним для забезпечення надійного та безпечного зв'язку. Запропоновано багато методів захисту даних від атак побічного каналу за допомогою маскування даних, переключення ключів тощо. Запропонований алгоритм використовує алгоритм переключення ключів для захисту алгоритму від атак побічного каналу. Запропонований алгоритм використовує початковий секретний ключ (K_0) як вхід і генерує інший ключ для кожного сеансу (K_i).

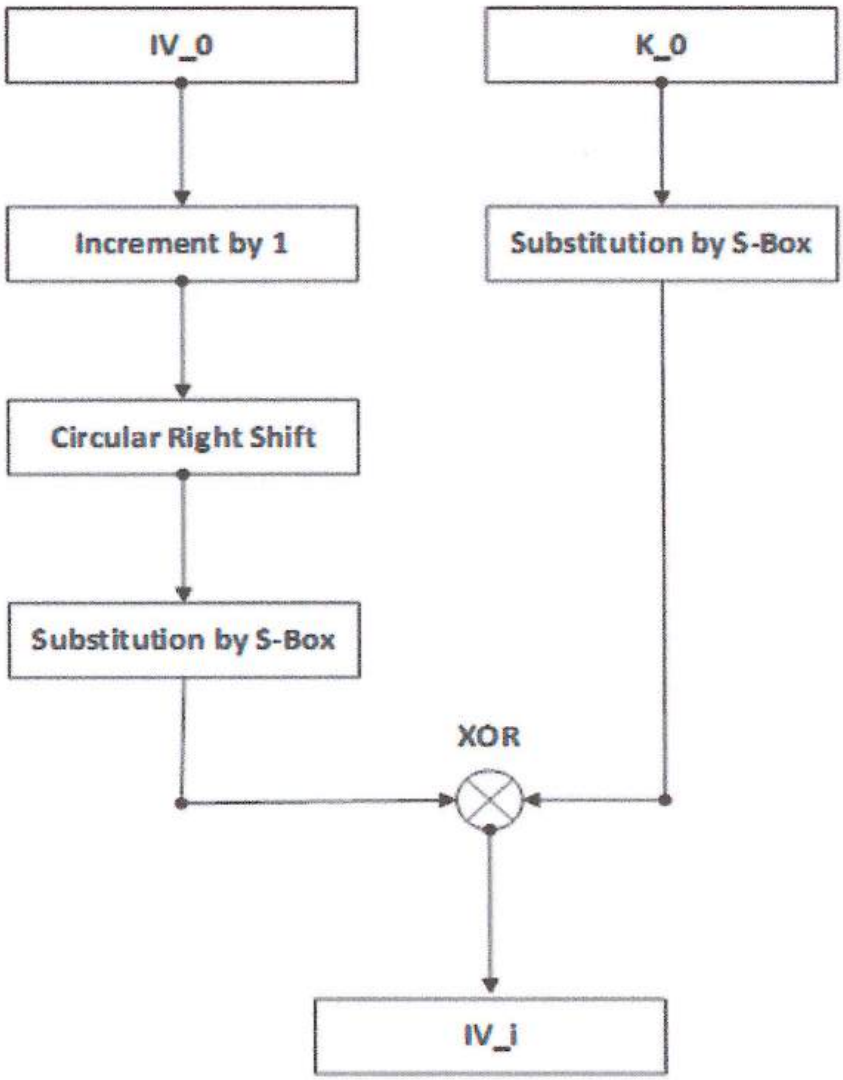


Рис.2.2 Потік графіка запропонованого алгоритму генерації IV

Секретний ключ збільшується на одиницю, циклічно праворуч зміщується, а потім замінюється за допомогою S-Box. Початкове IV (тобто IV_0) приймається як вхід в алгоритм переключення ключів. Початкове IV замінюється за допомогою S-Box, а потім ексклюзивно-або з заміщеним оригінальним ключем. Новий згенерований ключ буде ключем для наступного блоку. Алгоритм генерації IV та алгоритм переключення ключів є однаковими, за винятком того, що входи різні

3 ІМПЛЕМЕНТАЦІЯ МЕТОДОЛОГІЇ І АРХІТЕКТУРНА ОПТИМІЗАЦІЯ

3.1 Імплементація методології

У цьому розділі представлено методи реалізації, доступні для реалізації алгоритмів AE на FPGA. Обговорення методів реалізації має важливе значення для розуміння переваг і недоліків кожної техніки. Також представлені різні архітектурні оптимізації для реалізації запропонованого алгоритму AE для супутникових додатків.

Існує кілька методів реалізації, доступних для реалізації будь-якого алгоритму на FPGA, включаючи алгоритми AE. Методи включають проектування ядра Hardware Description Language (HDL), проектування ядра Intellectual Property (IP) та проектування ядра на основі високорівневого синтезу (HLS).

Проектування ядра HDL

Проектування ядра HDL передбачає проектування алгоритму за допомогою мов HDL Verilog і VHDL [18]. Проектування виконує окремий дизайнер або дослідницька група для їх використання або розповсюдження за відкритою ліцензійною угодою. Проектування має багато переваг, таких як контроль над модифікацією алгоритму, можливість реалізації на FPGA будь-якого виробника, модифікацію алгоритму для використання для дослідницьких цілей, реалізацію архітектурних оптимізацій, відсутність проблем з ліцензією та багато іншого.

Проектування IP ядра

Проектування ядра IP передбачає проектування алгоритму за допомогою попередньо визначених ядра IP, наданих виробником FPGA або сторонньою компанією. Ядра IP включають реалізацію відомих алгоритмів, а також їх оптимізовану реалізацію для конкретного сімейства FPGA певного виробника. Проектування на основі ядра IP допомагає в швидкому створенні прототипів і високо оптимізованій реалізації. Крім того, проектування ядра IP призначене для певного виробника і не може використовуватися з іншим виробником FPGA; ядро IP має обмежений контроль над модифікацією деяких параметрів. Ядро IP не може

використовуватися для дослідницьких цілей, архітектурні модифікації не можуть застосовуватися тощо. Недоліки проектування ядра IP вплинули на використання ядер IP для багатьох додатків, включаючи дослідження. Недавно було представлено порівняння між проектуванням ядра HDL та проектуванням ядра IP щодо їх корисності в різних додатках.

Дизайн на основі високорівневого синтезу

Проектування на основі HLS передбачає проектування алгоритмів за допомогою програмного забезпечення, такого як MATLAB. Проектування на основі HLS виконується шляхом програмування алгоритму за допомогою мови високого рівня, такої як мова C, а потім перетворення програми мови C в HDL. Багато виробників FPGA, включаючи Xilinx, Microsemi Actel та багато інших, надали свої інструменти для імпорту коду в MATLAB для реалізації на їх FPGA. Реалізація за допомогою такого методу може допомогти швидко реалізувати складний алгоритм за допомогою попередньо запрограмованого алгоритму з MATLAB та іншого програмного забезпечення на FPGA. Алгоритм забезпечує контроль над проектуванням алгоритму, архітектурні оптимізації можна застосовувати до алгоритму, і проект можна реалізувати на FPGA будь-якого виробника, інструмент якого інтегровано з MATLAB або надано виробником. Крім того, недоліком цієї методики є те, що реалізація не оптимізована щодо використання площі. Реалізацію різних алгоритмів за допомогою HLS представили різні дослідники [19].

3.2 Архітектурна оптимізація

Існує багато архітектурних оптимізацій, представлених для оптимізації використаної площі, підвищення пропускної здатності та багатьох інших цілей в залежності від області застосування. У цій роботі ми зосереджені на архітектурній оптимізації для супутникових додатків, які будуть використовуватися в космосі та на землі. Реалізація на землі досить схожа з загальними архітектурними оптимізаціями, хоча архітектурна оптимізація для супутників у космосі має різні цілі. Космічне середовище впливає вакуум, високий тиск і особливо

випромінювання. Тому оптимізації повинні бути реалізовані з урахуванням цих ефектів. Основними вимогами є висока пропускна здатність даних для швидкого зв'язку та реалізація для вирішення проблем у космічному середовищі.

Імплементація паралельної архітектури

Запропонований алгоритм AE має паралельну архітектуру і використовує алгоритм AES-CTR, який не має режиму ланцюжка або зворотного зв'язку для забезпечення високої пропускної здатності даних. Паралельна реалізація вимагає меншої кількості тактових циклів для проектування реалізації. Крім того, алгоритми з режимом ланцюжка або зворотного зв'язку зазвичай оптимізуються за допомогою методу конвеєризації. Запропонований алгоритм вже має паралельну реалізацію, хоча він споживатиме більше площі для реалізації порівняно з послідовною реалізацією. Запропонований алгоритм вимагає небагато тактових циклів для генерації IV та ключа, на додаток до тактових циклів, необхідних для блочного шифру. Паралельна оптимізація може попередньо обчислити ключі та IV за допомогою комбінаційної логіки для збереження цих кількох тактових циклів. В результаті зменшується кількість тактових циклів, необхідних для генерації ключа та IV для генерації кожного шифротексту.

Легка реалізація

Алгоритм AES складається з десяти раундів і споживає десять тактових циклів для реалізації. Проте, зменшене використання тактових циклів також є безпечним згідно з аналізом попередніх досліджень. Зменшені раунди, що пропонуються, також приймаються моїми нещодавно запропонованими алгоритмами AE для легких операцій. Аналогічно, пропускну здатність даних можна збільшити, використовуючи менш раундову реалізацію блокового шифру, використовуючи той самий запропонований алгоритм для легкої реалізації. Результати реалізації обговорюються в розділі експериментальних результатів для його застосовності.

Імплементація попередньо згенерованої таблиці

Запропонований алгоритм AE використовує S-Box для підстановки для алгоритму AES-CTR, а також у реалізації для алгоритму генерації IV і

переключення ключів. Звичайний метод реалізації S-Box — це реалізація за допомогою попередньо згенерованої таблиці, що зберігається в пам'яті. Крім того, пам'ять, конкретно SRAM, найбільше страждає від випромінювання в космосі. Тому використання пам'яті в космосі викличе збої в його роботі. Методи пом'якшення включають алгоритм більшості голосів, застосований до ресурсів пам'яті, що втричі збільшує використовувані ресурси пам'яті. Інший підхід полягає в тому, щоб уникати використання ресурсів пам'яті, можна використовувати реалізацію S-Box на полі вежі, щоб уникнути пам'яті. Ця оптимізація поля вежі для запропонованого алгоритму AE може захистити від впливу випромінювання для супутникових додатків.

3.3 Практичні результати

У цьому розділі представлені результати реалізації, отримані в результаті реалізації запропонованого алгоритму AE на FPGA. Результати, отримані після застосування різних запропонованих оптимізацій, також наводяться разом з порівнянням з відомими алгоритмами AE, такими як алгоритм AES-GCM. Експеримент проводиться на FPGA Xilinx Virtex-6, а симуляція проводиться за допомогою програмного забезпечення для симуляції Modelsim для перевірки результатів. Вектори тестів використовуються зі стандарту режиму роботи NIST [20]. Апаратний дизайн написано на Verilog HDL. Експеримент проводиться з використанням різної кількості блоків відкритого тексту; для експериментів використовується розмір ключа 128 біт. Використання ресурсів для реалізації для різної кількості блоків відкритого тексту показано на рисунку 3.1.

№	Імплементація	Ділянка	Блокова RAM	Пропускна здатність
1	Block Size-1	LUT: 1296 Reg.: 806	10	6.27
2	Block Size-2	LUT: 1950 Reg.: 1209	15	8.70
3	Block Size-3	LUT: 2588 Reg.: 1612	20	10.87
4	Block Size-4	LUT: 3235 Reg.: 2015	25	12.76
5	Block Size-5	LUT: 3877 Reg.: 2418	30	14.40
6	Block Size-6	LUT: 4525 Reg.: 2821	35	15.86
7	Block Size-7	LUT: 5178 Reg.: 3224	40	17.76
8	Block Size-8	LUT:5825 Reg.: 3627	45	18.35
9	Block Size-9	LUT:6474 Reg.: 4030	50	19.43
10	Block Size-10	LUT: 7136 Reg.: 4433	55	20.39

Рис. 3.1 Використання ресурсів для реалізації запропонованого алгоритму

Використання ресурсів обчислюється у формі таблиці пошуку (LUT), реєстрів, блокової оперативної пам'яті (BRAM), тактової частоти та пропускної здатності. Тактова частота реалізації запропонованої конструкції на FPGA Xilinx Virtex-6 становить 318,67 МГц. Пропускна здатність розраховується шляхом множення кількості бітів на тактову частоту реалізації та поділу отриманого результату на кількість тактових циклів, використаних для реалізації. Загальна кількість тактових циклів реалізації для одного блоку відкритого тексту довжиною 128 біт становить дванадцять тактових циклів. Де один тактовий цикл використовується для генерації IV та переключення ключів, десять тактових циклів використовуються для блокового шифру AES і один тактовий цикл для ексклюзивного-або ключа, відкритого тексту та потоку ключів. Було помічено, що зі збільшенням блоку відкритого тексту на 128 біт збільшується один тактовий цикл для реалізації. Оскільки алгоритм генерації IV та переключення ключів працює в режимі ланцюжка, а вхід наступного IV залежить від виходу попереднього згенерованого IV. Аналогічно працює алгоритм переключення ключів на тому ж

явищі. Отже, генерація одного шифротексту та тегу автентифікації вимагає тринадцяти тактових циклів.

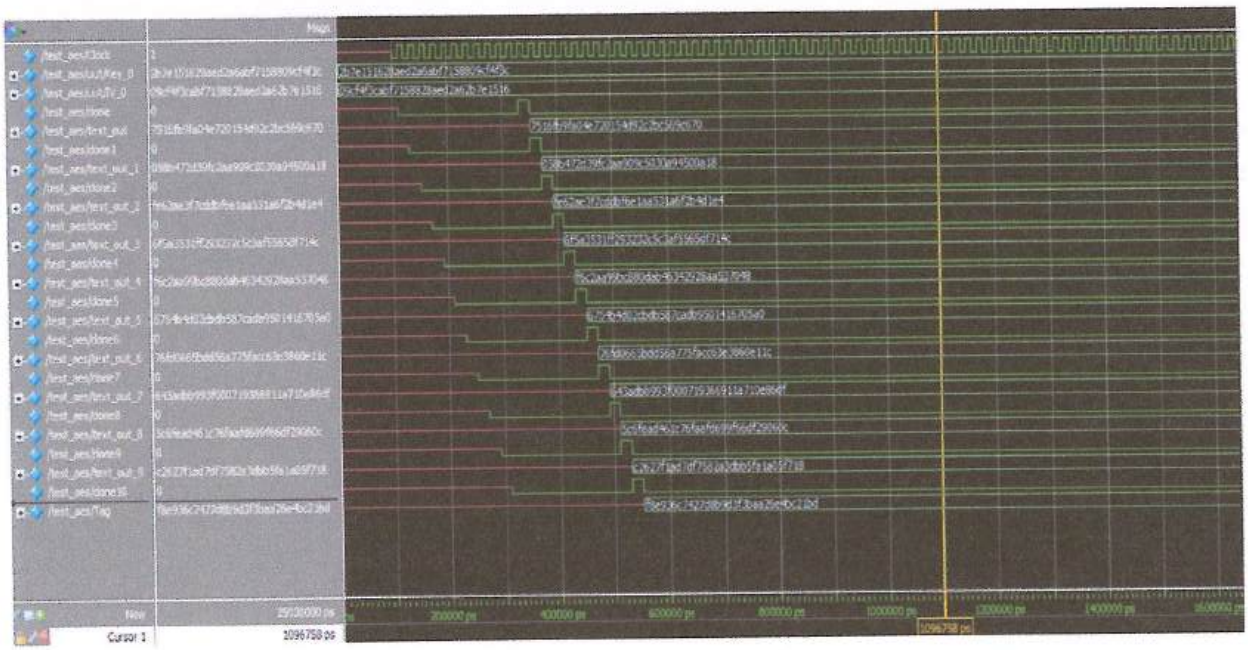


Рис.3.2. Результати симуляції запропонованого алгоритму

Одиночний шифротекст генерується за дванадцять тактових циклів, поєднуючи один тактовий цикл для генерації ключа та IV, десять тактових циклів для десяти раундів алгоритму AES та один тактовий цикл для ексклюзивного-або ключа, IV та потоку ключів. Крім того, наступний шифротекст потребує додавання ще одного тактового циклу для генерації шифротексту. Таким чином, генерація десяти шифротекстів включає двадцять два тактових цикли. Тег автентифікації вимагає ще одного тактового циклу і загалом двадцять три тактових цикли використовуються для генерації десяти блоків. Також алгоритм забезпечує блочний вихід 128 біт на кожен тактовий цикл після затримки в тринадцять тактових циклів. Ця властивість робить пропускну здатність алгоритму кумулятивно 40,78 Гбіт/с (128*318.67/1). Рисунок 3.2. показує результати симуляції реалізації запропонованого алгоритму. Реалізація запропонованого алгоритму AE виконується за допомогою того самого відкритого тексту (f0f1f2f3f4f5f6f7f8f9000000000000). Він також використовує один ключ для генерації більшої кількості ключів з алгоритму переключення ключів та для блокового шифру для генерації шифротексту. Також він використовує один IV для

генерації IV. На рисунку 3.3. показано значення згенерованих IV та ключів. Згенеровані IV та ключі були протестовані за допомогою тесту NIST на прогон та частоту, IV та ключі пройшли тест на прогон та частоту.

Значення	IV Генерування	Ключі
0	09CF4F3CABF7158828AED 2A62B7E1516	2B7E151628AED2A6ABF7 158809CF4F3C
1	AE67054CC8EBCB389833A 0295882E3D6	2B82E3D69833A029C8EB CB380367054C
2	FF304A6377796CFA4BBC2 93E90092702	2BF227024BBC293E5D796 CFA8D9E4A63
3	E7B56680DE81FBDB5D75 A31F537858E7	5813F9E75D75A31F0581F BDBAB796664
4	7EA4344E9CEDE171869C6 7B7D2EFF579	908B34E6869C67B7275EE 171F2963464
5	84F3FB8B1BDC39484C479 A7D1F7FA991	E9E43C6478479A7DE89D E4486840FB64
6	DD450DE169CC296D954EE 4767282CC36	BE03F6C8894EE4768BCB 3C12E944A670
7	D9C91DCBB96A4F6A16341 9261309B744	9FF68BA8793419265A3D BE2563C9B440
8	3F9A409EB23179F149CAA 71800D53DD1	8B85EAA389D0A718EC96 2BED399A0EF0
9	314EEEC3FF49D0652BB1B 4A062888FF5	A7AF623A7EF3B4A00C57 EC662F4E9CFB
10	5CAFACA822AD2E0748090 E97C69124E4	EC844C4F17DE0E975B15 F7E701AF76B4
11	15A8AF67B6553D5F2B9A4 E7710D8CBAB	43A673275FB79C77A19A BA293CFDB3F9

Рис.3.3 Значення згенерованих IV та ключів

3.4 Результати Архітектурної оптимізації

Різні архітектурні оптимізації, запропоновані для реалізації запропонованого алгоритму, реалізовані на FPGA Virtex-6 для реалізації переваг та недоліків запропонованої реалізації.

Результат імплементації паралельної архітектури

Хоча архітектура є паралельною, алгоритми генерації IV та переключення ключів реалізуються в режимі ланцюжка. Крім того, алгоритм можна використовувати повністю в паралельному режимі за допомогою попереднього обчислення IV та ключів. Попереднє обчислення виконується за допомогою комбінаційної логіки. Попереднє обчислення IV та ключів для переключення

ключів заощадить тактові цикли та забезпечить паралельну реалізацію без режиму ланцюжка алгоритмів генерації IV та переключення ключів. Ця оптимізація збільшить пропускну здатність десяти блоків відкритого тексту довжиною 128 біт з 21,28 Гбіт/с до 40,78 Гбіт/с. Це пов'язано з тим, що тактові цикли для реалізації будуть десять циклів для блокового циклу AES один для ексклюзивного-або ключів, відкритих текстів і потоків ключів (128*11*318.67 /11).

Результати використання ресурсів оптимізації паралельної архітектури 1 показано на рисунку 3.4.

№	Імплементація	Ділянка	Блокова RAM	Пропускна здатність
1	Block Size-1	LUT: 1296 Reg.: 806	10	7.41
2	Block Size-2	LUT: 1950 Reg.: 1209	15	11.12
3	Block Size-3	LUT: 2588 Reg.: 1612	20	14.83
4	Block Size-4	LUT: 3235 Reg.: 2015	25	18.54
5	Block Size-5	LUT: 3877 Reg.: 2418	30	22.25
6	Block Size-6	LUT: 4525 Reg.: 2821	35	25.96
7	Block Size-7	LUT: 5178 Reg.: 3224	40	29.67
8	Block Size-8	LUT:5825 Reg.: 3627	45	33.38
9	Block Size-9	LUT:6474 Reg.: 4030	50	37.09
10	Block Size-10	LUT: 7136 Reg.: 4433	55	40.78

Рис.3.4. Результати використання ресурсів оптимізації паралельної архітектури

Результат легкої реалізації

Легка функціональність бажана для багатьох додатків, включаючи супутникові програми. Зростання тенденції до розвитку мікро- та міні-супутників обмежило ресурси на борту. Замість того, щоб обмежувати функціональність, легкі алгоритми можуть забезпечувати широке застосування на менших бортових ресурсах. Алгоритм AES складається з десяти раундів, і, як зазначалося в попередньому аналізі [48], чотири раунди є значними для забезпечення безпеки даних. Реалізація чотирираундового алгоритму AES використовується в алгоритмі,

відомому як алгоритм POET, представленому на конкурсі CEASAR. У цій роботі для запропонованого алгоритму AE ми також пропонуємо легковажну реалізацію за допомогою чотирьох раундів алгоритму AES. Використання ресурсів для цієї оптимізації наведено на рисунку 3.5. Крім того, збільшується пропускна здатність даних. Легковажна версія вимагає шести тактових циклів для реалізації одного блоку розміром 128 біт.

№	Імплементація	Ділянка	Блокова RAM	Пропускна здатність
1	Block Size-1	LUT: 1296 Reg.: 806	10	11.65
2	Block Size-2	LUT: 1950 Reg.: 1209	15	15.29
3	Block Size-3	LUT: 2588 Reg.: 1612	20	18.12
4	Block Size-4	LUT: 3235 Reg.: 2015	25	20.39
5	Block Size-5	LUT: 3877 Reg.: 2418	30	22.24
6	Block Size-6	LUT: 4525 Reg.: 2821	35	23.79
7	Block Size-7	LUT: 5178 Reg.: 3224	40	25.10
8	Block Size-8	LUT:5825 Reg.: 3627	45	26.22
9	Block Size-9	LUT:6474 Reg.: 4030	50	27.19
10	Block Size-10	LUT: 7136 Reg.: 4433	55	28.04

Рис.3.5 Використання ресурсів запропонованої оптимізації

Крім того, тег автентифікації вимагає ще одного тактового циклу та робить загальну кількість тактових циклів сім для реалізації. Тактова частота залишається такою ж, як і для реалізації запропонованого алгоритму, але тактові цикли, необхідні для легковажної реалізації, зменшуються.

Результат імплементації попередньо згенерованої таблиці

Третя оптимізація для запропонованого алгоритму базується на використанні алгоритмів у космічному середовищі. Пам'ять, наприклад SRAM, впливає на випромінювання та змушує пов'язаний алгоритм працювати неправильно. Раніше було реалізовано імплементацію поля вежі [22]. Тому в запропонованому алгоритмі AE, щоб уникнути використання пам'яті, пропонується прийняти підхід поля вежі

для оптимізації продуктивності. Реалізація поля вежі реалізує таблицю S-Box на логічній області FPGA замість того, щоб зберігати таблицю S-Box у блоці пам'яті на FPGA. Імплементація вимагає більшого використання площі з точки зору таблиць пошуку (LUT) та реєстрів, але вона не використовуватиме пам'ять. Використання ресурсів алгоритму для запропонованого алгоритму показано в таблиці 5. Частота реалізації запропонованого алгоритму AE зменшується з оптимізацією -3 (тобто 240,39 МГц). Тому пропускна здатність даних також зменшилася порівняно з запропонованим алгоритмом.

№	Імплементація	Ділянка	Блокова RAM	Пропускна здатність
1	Block Size-1	LUT: 2829 Reg.: 1057	0	4.39
2	Block Size-2	LUT: 4219 Reg.: 1583	0	6.15
3	Block Size-3	LUT: 5621 Reg.: 2109	0	7.69
4	Block Size-4	LUT: 7023 Reg.: 2635	0	9.04
5	Block Size-5	LUT: 8425 Reg.: 3161	0	10.25
6	Block Size-6	LUT: 9827 Reg.: 3687	0	11.33
7	Block Size-7	LUT: 11229 Reg.: 4213	0	12.30
8	Block Size-8	LUT:12631 Reg.: 4739	0	13.18
9	Block Size-9	LUT:14033 Reg.: 5265	0	13.98
10	Block Size-10	LUT: 15458 Reg.: 5791	0	14.71

Рис.3.6 Використання ресурсів з S-BOX оптимізацією

Окрім індивідуальних оптимізацій, комбінація оптимізацій дуже корисна для оптимізації продуктивності запропонованих алгоритмів AE. Оптимізацію 1 можна застосовувати в поєднанні з оптимізацією 2 для легкої та паралельної реалізації запропонованого алгоритму. Результати застосування обох оптимізацій (1 і 2) споживають п'ять тактових циклів для будь-якої кількості блоків розміром 128 біт. Аналогічно, оптимізацію 2 можна використовувати разом з оптимізацією 3 для легкої та стійкої до випромінювання реалізації.

Окрім окремих оптимізацій, комбінація оптимізацій дуже корисна для оптимізації продуктивності запропонованих алгоритмів AE. Оптимізацію 1 можна застосовувати в поєднанні з оптимізацією 2 для легкої та паралельної реалізації запропонованого алгоритму. Результати застосування обох оптимізацій (1 і 2) споживають п'ять тактових циклів для будь-якої кількості блоків розміром 128 біт. Аналогічно, оптимізацію 2 можна використовувати разом з оптимізацією 3 для легкої та стійкої до випромінювання реалізації.

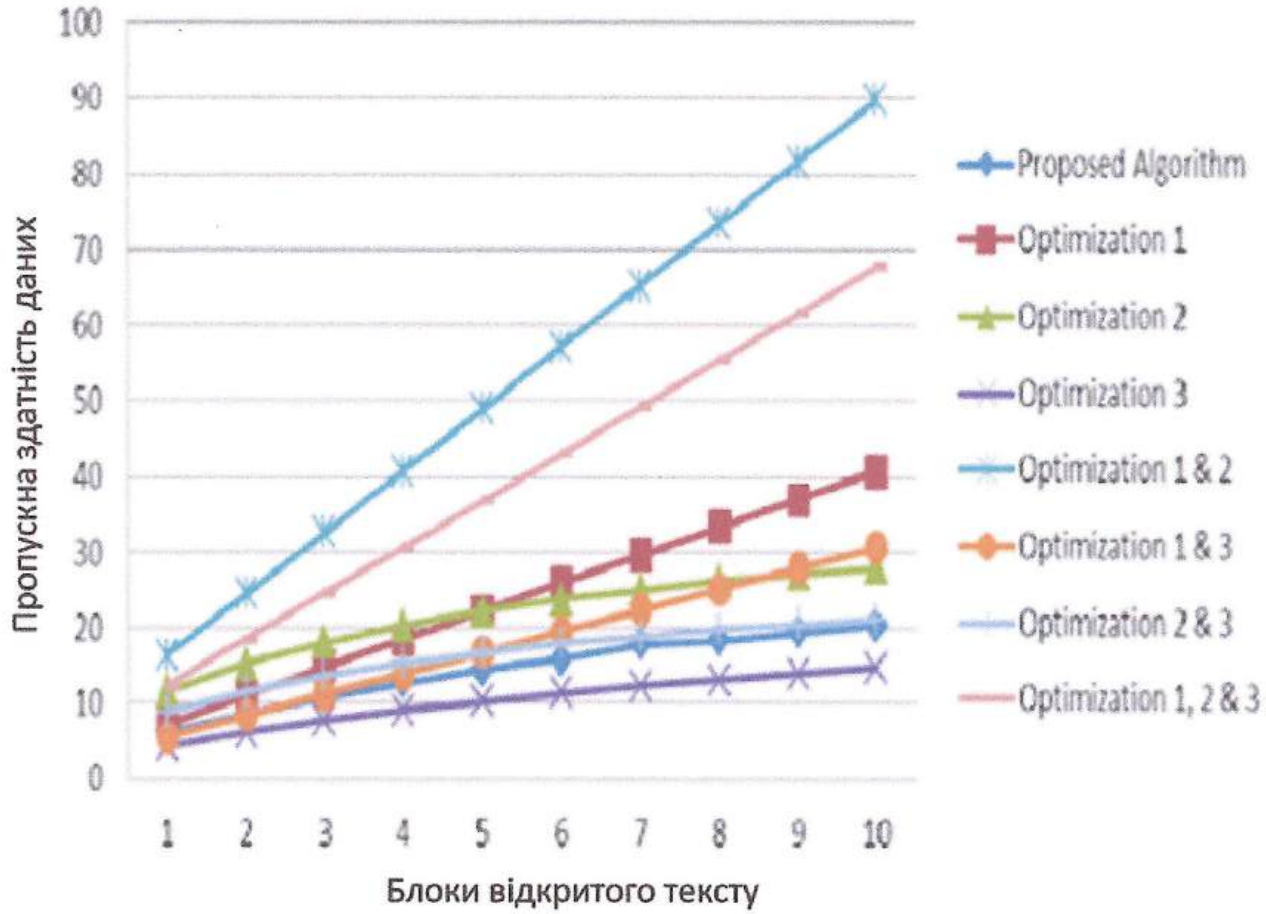


Рис.3.7 Графік порівняння пропускної здатності запропонованого алгоритму з різними комбінаціями оптимізацій

Результати використання обох оптимізацій (2 і 3) споживають сім тактових циклів для одного блоку розміром 128 біт; збільшення блоків відкритого тексту збільшує тактові цикли на 1. Частота реалізації також зменшилася до тієї ж частоти оптимізації S-Box.

Аналогічно, оптимізацію 1 можна застосовувати в поєднанні з оптимізацією 3 для паралельної та стійкої до випромінювання реалізації. Результати

використання обох оптимізацій (2 і 3) споживають 11 тактових циклів для будь-якої кількості блоків розміром 128 біт. Частота реалізації також зменшилася до тієї ж частоти оптимізації S-Box.

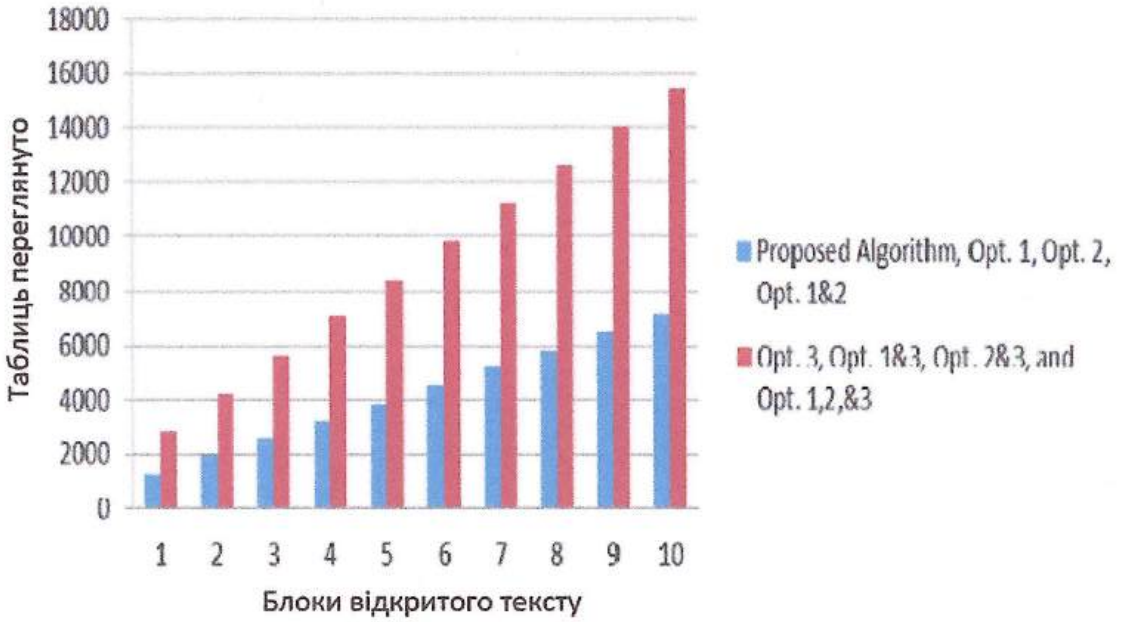


Рис.3.8 Графік порівняння використання ресурсів запропонованого алгоритму з різними комбінаціями оптимізацій

Крім того, оптимізацію 1 можна застосовувати в поєднанні з оптимізацією 2 та оптимізацією 3 для паралельної, легкої та стійкої до випромінювання реалізації. Результати використання всіх оптимізацій (1, 2 і 3) споживають п'ять тактових циклів для будь-якої кількості блоків розміром 128 біт. Частота реалізації також зменшилася до тієї ж частоти оптимізації S-Box.

Таким чином, оптимізації можна впроваджувати окремо, а також з комбінаціями оптимізації для подальшого вдосконалення. Вплив індивідуальної оптимізації та комбінації оптимізацій для пропускної здатності даних показано на рисунку 3.7, а для використання ресурсів - на рисунку 3.8.

Рисунок 3.7 показує результати різних оптимізацій та комбінації різних оптимізацій для пропускної здатності даних. Для реалізації десяти блоків відкритого тексту комбінація оптимізації 1 та 2 забезпечує найвищу пропускну здатність, потім йде комбінація оптимізації 1, 2 та 3. Крім того, пропускна здатність даних оптимізації 3 має найнижче значення.

Рисунок 3.8 показує використання ресурсів для різних оптимізацій та комбінації різних оптимізацій. Оптимізація 3, комбінація оптимізації 1 та 3, комбінація оптимізації 2 та 3 і комбінація оптимізації 1, 2 та 3 забезпечують високе використання ресурсів. Крім того, оптимізація 1, оптимізація 2 і комбінація оптимізації 1 та 2 забезпечують низьке використання ресурсів.

Проведений аналіз показує, що комбінація оптимізації 1 та 2 забезпечує високу пропускну здатність даних при низькому споживанні ресурсів, хоча вона не уникає використання SRAM для стійкості до випромінювання. Тому комбінація оптимізації 1, 2 та 3 забезпечує кращі характеристики, такі як висока пропускна здатність і стійкість до випромінювання, але споживає більше апаратних ресурсів. Космічне середовище в значній мірі впливає на випромінювання, тому комбінація оптимізації 1, 2 та 3 рекомендована для використання в супутникових додатках.

3.5 Порівняння запропонованого алгоритму

Існують різні алгоритми AE із захистом від зловживання нонсом, такі як алгоритм AES-GCM-SIV. Також алгоритм AES-GCM може забезпечувати високу пропускну здатність реалізації та використовуватися в широкому спектрі додатків. Стандарти CCSDS та ESA рекомендують алгоритм AES-GCM для супутникових додатків, тому в цій роботі проводиться порівняння з AES-GCM та AES-GCM-SIV.

У минулому реалізацію алгоритму AES-GCM на FPGA представили B. Yang et al.. Імплементація забезпечує оптимізацію конвеєра для збільшення пропускної здатності даних. S. Lemsitzer et al. представили свою реалізацію AES-GCM з акцентом на оптимізацію реалізації AES за допомогою FPGA Xilinx Virtex-4. Нещодавно J. Vliegen et al. показали реалізацію алгоритму AES-GCM на FPGA Xilinx Virtex 7 із захистом від атак DPA. G. Zhou et al. реалізували свою швидку реалізацію на FPGA Xilinx за допомогою множника Каратсуби. L. Henzen et al. представили свою 100 Гбіт/с реалізацію алгоритму AES-GCM для додатків Ethernet на FPGA Xilinx Virtex 5. K. M. Abdellatif et al. показав свою високошвидкісну реалізацію для програми зі зміною ключа та алгоритму AEGIS за допомогою FPGA Xilinx Virtex 5. Крім того, Y. Zhang et al. представили реалізацію алгоритму AES-GCM із високою пропускною здатністю за допомогою конвеєрної архітектури на

FPGA Xilinx Virtex-5. Koteshwara et al. реалізували AES-GCM та AES-GCM-SIV та порівняли їхні ресурси разом із оптимізаціями на FPGA Altera Cyclone V.

Використання ресурсів запропонованого алгоритму AE, разом з двома рекомендованими оптимізаціями для супутникових додатків, порівнюється з попередніми реалізаціями. Запропонований алгоритм AE з оптимізацією не тільки захищає від атак побічних каналів і зловживання нонсом, але й забезпечує високу пропускну здатність даних і кращу ефективність. Рисунок3.9. надає порівняння використання площі, а також порівняння пропускної здатності даних запропонованого алгоритму з різними реалізаціями.

Імплементація	Тактова частота (MHz)	Ділянок	Блокова RAM	Пропускна здатність (Gb/s)	Ефективність (Mbps/Ділянку)
B.Yang	271	463328	0	34.7	0.07
S.Lemsitzer	120	27800	0	15.3	0.55
J.Vliegen	119	38241	0	15.2	0.39
G.Zhou	305	8077	0	39.0	4.83
L.Hanzen	233	18505	0	48.8	2.63
K.M. Abdellatif	264	7475	0	31.4	4.19
K.M. Abdellatif 2	232	5512	0	29.7	5.38
Y. Zhang	381	6482	0	48.8	7.54
Koteshwara GCM	50	4087	0	4.1	1.02
Koteshwara GCM-SIV	50	4262	0	3.9	0.93
Запропонований метод	318	7136	55	20.3	2.85
Оптимізація 1 & 2	318	7136	55	89.7	12.57
Оптимізація 1, 2 & 3	240	15458	0	67.6	4.37

Рис.3.9 Порівняння AES-GCM і AES-SIV алгоритмів

3.6 Аналіз безпеки

Запропонований алгоритм AE складається з трьох частин: генерації IV, повторного ключового розподілу та алгоритмів AES-CTR. Алгоритм, відповідальний за конфіденційність запропонованого алгоритму AE, - це алгоритм AES-CTR; багато дослідників надали аналіз безпеки алгоритму AES-CTR для десяти раундів [23] та чотирьох раундів. Крім того, безпека AES-CTR залежить від IV, IV не можна повторювати, оскільки використання того ж ключа та пари IV може генерувати той самий шифротекст. Тому найважливішими вхідними даними для алгоритму AES-CTR є ключ та IV; тому в запропонованому алгоритмі для рандомізації обох векторів для підвищення безпеки проти лінійних та

диференціальних атак використовується алгоритм генерації IV разом з алгоритмом повторного ключового розподілу.

Атака методом грубої сили та частота букв

Атака методом грубої сили полягає в переборі всіх можливих комбінацій ключа та IV для порушення безпеки алгоритму AES-CTR. Універсальний алгоритм AES-CTR вимагає 2128 комбінацій IV разом з 2128 комбінаціями відкритого тексту, що становить 2256 комбінацій для атаки методом грубої сили зі сталим ключем.

У запропонованому алгоритмі AE IV рандомізується, а алгоритм повторного ключового розподілу генерує рандомізований ключ для кожного шифротексту. Тому для атаки методом грубої сили 2128 комбінацій IV, 2128 комбінацій ключа та 2128 комбінацій відкритого тексту в сукупності становитимуть 2384 комбінації зі сталим ключем для блочного шифру. Атака частотності букв передбачає спробу зв'язку між частотою букв шифротексту для припущення відкритого тексту для порушення безпеки алгоритму AES-CTR. У запропонованому алгоритмі AE IV більш рандомізовано порівняно з загальним алгоритмом, а алгоритм переключення ключів також генерує більш рандомізований ключ для кожного шифротексту. Тест частотності букв виконується за допомогою статистичних тестів NIST на випадковість, таких як тести прогону та частотності. Результати тестів прогону та частотності NIST для згенерованих ключів алгоритмом переключення ключів та IV за допомогою алгоритму генерації IV наведено в рисунку 3.10 IV та ключ, згенеровані за допомогою запропонованого алгоритму, пройшли тест прогону та частотності NIST.

Алгоритм переключення ключів був розроблений для генерації нового ключа сеансу для кожного шифротексту, згенерованого за допомогою комбінації ключа та IV. Імовірність використання одного й того ж ключа з тим самим IV та відкритим текстом дуже низька, тому ключ є безпечним та як OTP. Отже, ми можемо бачити, що навіть якщо ключ повторюється, шанси на той самий відповідний IV незначні.

Блоки	Частота IV	Частота ключова	Прогін IV	Прогін ключовий
1	0.5959	0.3768	0.8399	2.0000
2	0.4795	0.1573	1.6028	1.8940
3	0.4795	0.8597	0.6897	0.1109
4	0.7237	0.1573	0.0750	0.1508
5	0.7237	0.5959	0.1088	0.0199
6	0.5959	0.7237	0.0724	0.0503
7	0.2888	0.5959	1.2023	1.9773
8	0.8597	0.3768	1.6219	0.1356

Рис.3.10 Тест на частоті NIST та прогоновий тест

Атаки попереднього обчислення

Окрім атаки грубої сили, є ще два ефективніших атаки, що ґрунтуються на попередньому обчисленні великої бази даних перед атакою на алгоритм. Ці атаки називаються атакою зіткнення ключів та атакою компромісу часу та пам'яті Геллмана. Ці атаки забезпечують коротший час порівняно з атакою грубої сили. Крім того, для попереднього обчислення потрібно більше пам'яті. Атаки не вимагають жодних знань про відкритий текст під час попереднього обчислення.

У атаці зіткнення ключів кількість бітів секретних ключів становить n. Ефективність атаки зіткнення ключів може вгадувати розмір ключа n-lgM, де M є кількістю секретних ключів. З іншого боку, атака Геллмана має важливий адекватний розмір 2n/3. Запропонований алгоритм AE має ефективність 256-lgM для зіткнення ключів та 170 біт для атаки Геллмана порівняно з 128-lgM для зіткнення ключів та 85 для атаки Геллмана для алгоритму AES-CTR.

Однак блочний шифр може захистити атаки попереднього обчислення, залежність AES-CTR від IV значно ефективна проти атак попереднього обчислення. Зловмисник обчислює значення за допомогою IV на етапі попереднього обчислення, а шляхом рандомізації IV створює обчислювальні труднощі для зловмисника вдвічі. Також ініціалізація IV секретним вектором або використання непередбачуваного алгоритму генерації може захистити від атаки попереднього обчислення. Подібний підхід інтегровано в запропонований алгоритм AE шляхом використання

секретного IV, спочатку такого ж, як і ключ. Отже, атаки попереднього обчислення неефективні проти запропонованого алгоритму AE.

Відома або вибрана атака відкритим текстом

Припустимо, що зловмисник має відомий відкритий текст P1 і P2, зашифрований за допомогою ключа K1. Тоді згенерований шифротекст за допомогою загального алгоритму AES-CTR буде:

(P1 XOR K1), (P2 XOR K1)

Зловмисники можуть виконати XOR вибраної послідовності S1 і S2, щоб отримати потрібну інформацію.

(S1 XOR (P1 XOR K1)), (S2 XOR (P2 XOR K1)) (S1 XOR P1) XOR K1), (S2 XOR P2) XOR K1) (S1 XOR P1), (S2 XOR P2)

Хоча запропонований алгоритм AE також має повторне ключове унікальне ключове XOR з відкритим текстом, тому зловмиснику буде важко порушити безпеку.

(S1 XOR (P1 XOR K1 XOR Ki)), (S2 XOR (P2 XOR K1 XOR Ki)) (S1 XOR P1 XOR Ki) XOR K1), (S2 XOR P2 XOR Ki) XOR K1) (S1 XOR P1 XOR Ki), (S2 XOR P2 XOR Ki)

Отже, запропонований алгоритм AE є безпечним проти атаки з відомим або обраним відкритим текстом.

Відома або атака компроментованим ключем

Припустимо, що зловмисник знає значення оригінального ключа, тому, знаючи алгоритм розширення ключа, він може вгадувати згенеровані ключі для алгоритму AES-CTR. Але навпаки, зловмисник не знає початкового секретного IV, тому вихід алгоритму переключення ключів і вихід алгоритму генерації IV йому невідомі, що внутрішньо зберігає безпеку запропонованого алгоритму AE. Крім того, зловмисник не знає ключів з алгоритму переключення ключів, що робить атаку слабкою, оскільки, знаючи ключ, зловмисник не може вгадувати ключі переключення ключів за допомогою шифротексту. 2128 комбінація ключів з алгоритму переключення ключів і 2128 комбінація IV з загальною кількістю 2256 комбінацій вхідних значень невідомі зловмиснику. Отже, знання одного ключа не допоможе зловмиснику вгадати відкритий текст. Тоді як запропонований алгоритм

використовує ключ у генерації IV, а IV у генерації ключа, робить проблеми для атаки вдвічі.

Атака бічного каналу

Атаки на побічні канали - це атаки, що здійснюються на електромагнітний шаблон переданого шифротексту або на аналіз переданої потужності шифротексту, наприклад, атака DPA. Хоча існує багато побічних каналів атак, окрім двох обговорюваних методів, згадані методи є найефективнішими побічними каналами атак. Сила атак побічних каналів ґрунтується на повторенні шаблону, щоб вгадати схожість шаблону. Запропонований алгоритм AE рандомізував IV і ключі з алгоритму переключення ключів, що робить шифротекст випадковим, відмінним від інших згенерованих шифротекстів. Тому побічні канали атаки, такі як атака DPA, не будуть можливими для порушення безпеки запропонованого алгоритму AE. Явище надання нових сеансових ключів використовується в запропонованому алгоритмі AE, що є суттєвою вимогою для запобігання атакам побічних каналів.

Атака зловживання нонсом

Атака зловживання нонсом включає використання одного й того ж нонсу в IV для генерації шифротексту. У загальному алгоритмі AES-CTR операція інкременту виконується для генерування унікального IV для шифротексту. Однак, як і раніше, нонс залишається незмінним, а збільшується лише поле лічильника. Запропонований алгоритм AE використовує секретний IV для початкового IV, а наступні IV генеруються за допомогою алгоритму генерації IV. Алгоритм генерації IV робить IV випадковими для генерації шифротексту. Також той самий нонс не використовується в IV для генерації шифротексту. Тому атака зловживання нонсом не ефективна для запропонованого алгоритму AE. Також помилка переповнення поля лічильника для повторення IV, яка існує в загальному алгоритмі, не застосовується до запропонованого алгоритму AE. Крім того, атака відтворення, наприклад, повторне надсилання шифротексту запропонованому алгоритму AE, не спрацює, оскільки ключ та IV змінюються щоразу.

ВИСНОВКИ

В першому розділі було докладно проаналізовано сучасні супутники та їхній широкий спектр застосувань у різних галузях, включаючи телекомунікації, навігацію та наукові дослідження космосу. З'ясовано, що архітектура супутникових систем є складною мережею, яка об'єднує різноманітні компоненти, такі як супутники, базові станції, датчики, енергопостачання та засоби зв'язку.

Детально проаналізовано можливі загрози для супутникових систем у вигляді кібератак та конфліктів. Виявлено, що супутники є вразливими до кіберзагроз, особливо атак на їхні електронні компоненти. Зокрема, зазначено важливість розробки та впровадження ефективних заходів з кіберзахисту для забезпечення безпеки та цілісності супутникових систем.

Виокремлено необхідність вдосконаленого шифрування для ефективного захисту конфіденційності та цілісності передаваних даних у супутникових системах. Зазначено, що стандарти шифрування є ключовим елементом в системах забезпечення безпеки, і їхнє використання важливо для захисту інформації в супутникових мережах.

У другому розділі запропоновано паралельний AE-алгоритм із захистом від зловживання нонсом та стійкими до побічних каналів атаками. Запропонований алгоритм використовує алгоритм генерації IV для захисту від зловживання нонсом шляхом рандомізації секретного IV. Також атаки побічних каналів, такі як атака DPA, неефективні для алгоритму через використання алгоритму переключення ключів. Тим часом, також запропоновано три оптимізації для досягнення високої пропускної здатності даних, легкої реалізації та захисту від космічних випромінювань. Також обговорюється вплив комбінації оптимізацій на апаратну реалізацію. Запропонований алгоритм порівнюється з попередньою реалізацією алгоритмів AES-GCM та AES-GCM-SIV для порівняння ефективності запропонованого алгоритму. Представлено аналіз безпеки запропонованого алгоритму для валідації запропонованого алгоритму.

У даній роботі був представлений і детально розглянутий новаторський паралельний AE-алгоритм, який володіє важливими перевагами в області безпеки та ефективності. Запропонований алгоритм вирішує проблему зловживання нонсом шляхом використання алгоритму генерації IV та рандомізації секретного IV, що забезпечує надійний захист від атак, пов'язаних із використанням нонсів.

Додатково, запропонований алгоритм стійкий до побічних каналів атак, таких як атака DPA, завдяки використанню алгоритму переключення ключів. Це робить його ефективним рішенням для застосувань, де безпека та захист від різноманітних атак є критично важливими.

Три запропоновані оптимізації значно підвищують пропускну здатність даних, роблячи алгоритм ефективним у використанні. Легкість реалізації та захист від космічних випромінювань роблять його привабливим для реальних застосувань в космічних системах. Оптимізації були спрямовані на максимізацію функціональності та стійкості алгоритму.

Порівняння із попередніми реалізаціями алгоритмів AES-GCM та AES-GCM-SIV підтверджує високу ефективність та безпеку запропонованого алгоритму.

Загальний висновок полягає в тому, що запропонований алгоритм є добре збалансованим рішенням, яке об'єднує в собі високу захищеність, ефективність та готовність до використання в космічних умовах. Результати аналізу безпеки підтверджують, що алгоритм може бути використаний в критичних застосуваннях, де надійний захист від різноманітних загроз є високопріоритетною задачею.

ПЕРЕЛІК ПОСИЛАННЬ

1.Fritz, J. (2013). Satellite hacking: A guide for the perplexed. In Culture Mandala: The Bulletin of the Centre for East-West Cultural and Economic Studies, 10(1), 21-50. Retrieved October 25, 2015 from http://epublications.bond.edu.au/cgi/viewcontent.cgi?article=1131&context=cm

2.LAWS. (2015). Easy definition of hacking. Retrieved October 26, 2015 from http://cyber.laws.com/hacking

3. Rooker, J. W. (2008). Satellite vulnerabilities. Defense Information Technical Center. Retrieved October 25, 2015 from http://www.dtic.mil/dtic/tr/fulltext/u2/a507952.pdf

4. Coffed, J. (2014). The threat of GPS jamming: the risk to an information utility. Retrieved October 26, 2015 from http://www.exelisinc.com/solutions/signalsentry/Documents/ThreatOfGPSJamming_February2014.pdf

5.Paganini, P. (2013, December 3). Satellite infrastructures: Principal cyber threats [PowerPoint]. Presented at CYBERSECURITY in Air and Satellite Navigation conference. Retrieved October 27, 2015 from http://www.aofs.org/wpcontent/uploads/2013/12/131203-Paganini-Satellite-infrastructures-Principal-cyberthreats_Final.pdf

6.Advanced Encryption Standard (AES) available at: https://www.geeksforgeeks.org/advanced-encryption-standard-aes/7.Murtaza, S. J. Hussain Pirzada, L. Jianwei, and T. Xu, "Air traffic surveillance using IP-based space information network," in Proc. 28th Wireless Opt. Commun. Conf. (WOCC), Beijing, China, May 2019, pp. 1–6.

8. CSDS 350.0-G-2. (Jan. 2006). The Application of CCSDS Protocols to Secure Systems. [Online]. Available: https://public.ccsds.org/Pubs/ 350x0g2.pdf

9.B. H. A. Zauner, S. Devlin, J. Somorovsky, and P. Jovanovic, "Noncedisrespecting adversaries: Practical forgery attacks on GCM in TLS," in Proc. USENIX WOOT, 2016, pp. 1–11.

10.E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with aleakage model," Lect. Notes Comput. Sci., vol. 2004, vol. 3156, pp. 16–29.

11.M. Zaba, N. Jamil, M. Rohmad, H. Abdul, and S. Shamsuddin, "The CiliPadi family of lightweight authenticated encryption (version 1)," NIST Lightweight Cryptogr. Round-1, Gaithersburg, MD, USA, 2019

12. C. I. Underwood and M. K. Oldfield, "Observations on the reliability of cots-device-based solid state data recorders operating in low Earth orbit," in Proc. 5th Eur. Conf. Radiat. Effects Compon. Syst., Fontevraud, France, 1999, pp. 387–393.

13. J. Vliegen, O. Reparaz, and N. Mentens, "Maximizing the throughput of threshold-protected AES-GCM implementations on FPGA," in Proc. IEEE 2nd Int. Verification Secur. Workshop (IVSW), Thessaloniki, Greece, Jul. 2017, pp. 140–145.

14.S. Koteshwara, A. Das, and K. K. Parhi, "FPGA implementation and comparison of AES-GCM and deoxys authenticated encryption schemes," in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), May 2017, pp. 1–4.

15.A. S. Bader and A. M. Sagheer, "Modification on AES-GCM to increment ciphertext randomness," Int. J. Math. Sci. Comput., vol. 4, no. 4, pp. 34–40, Nov. 2018.

16. S. Koteshwara, A. Das, and K. K. Parhi, "Architecture optimization and performance comparison of Nonce-Misuse-Resistant authenticated encryption algorithms," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 27, no. 5, pp. 1053–1066, May 2019.

17.S. J. H. Pirzada, S. of Cyber Science, China. TechnologyBeihang UniversityBeijing, A. Murtaza, L. Jianwei, and T. Xu, "Single event effects tolerant AES-CTR implementation for authentication of satellite communication," Int. J. Comput. Commun. Eng., vol. 8, no. 4, pp. 178–183, 2019.

18. IEEE Standard for Verilog Hardware Description Language, Standard IEEE Std 1364-2005 and Revision of IEEE Std 1364-2001, Apr. 2006, pp. 1–590.

19.S. Chhabra and K. Lata, "Hardware-software co-simulation of obfuscated 128-bit AES algorithm for image processing applications," in Proc. IEEE Int. Symp. Smart Electron. Syst. (ISES), Hyderabad, India, Dec. 2018, pp. 191–194

20. NIST, Special Publication 800-38A. (Dec. 2001). Recommendation for Block Cipher Modes of Operation: Methods and Techniques. [Online]. Available: https://csrc.nist.gov/publications/detail/sp/800-38a/final

21. J. Daemen, M. Lamberger, N. Pramstaller, V. Rijmen, and F. Vercauteren, "Computational aspects of the expected differential probability of 4-round AES and AES-like ciphers," Computing, vol. 85, nos. 1–2, pp. 85–104, May 2009.

22. D. Canright, "A very compact S-box for AES," in Proc. Int. Workshop Cryptograph. Hardware Embedded Syst., 2005, pp. 441–455

23. D. A. McGrew. (Nov. 2002). Counter Mode Security: Analysis and Recommendations. Accessed: Nov. 1, 2019. [Online]. Available: http://www.mindspring.com/-dmcgrew/ctr-security.pdf

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

Магістерська робота на тему,

«МЕТОДИ ЗАПОБІГАННЯ КІБЕРЗАГРОЗАМ ТА СУПУТНИКОВИМ КОНФЛІКТАМ У КОСМІЧНОМУ ПРОСТОРІ»

Виконав: студент 6 курсу, групи АРДМ-62

Клименко Микита





Актуальність – Однією з ключових причин, що підкреслює актуальність теми, є зростаюча залежність нашого сучасного суспільства від космічних технологій. Така значуща роль космічних технологій створює важливий контекст для розуміння, як кіберзагрози можуть впливати на ці системи. Інфраструктура космічних апаратів та супутників стає об'єктом зацікавлення для атак, оскільки їхній незахищений функціонал може призвести до серйозних наслідків. Постає питання у захищеному алгоритмі шифрування що буде спроможний обробити великі об’єми інформації достатньо забезпечивши захист інформації.

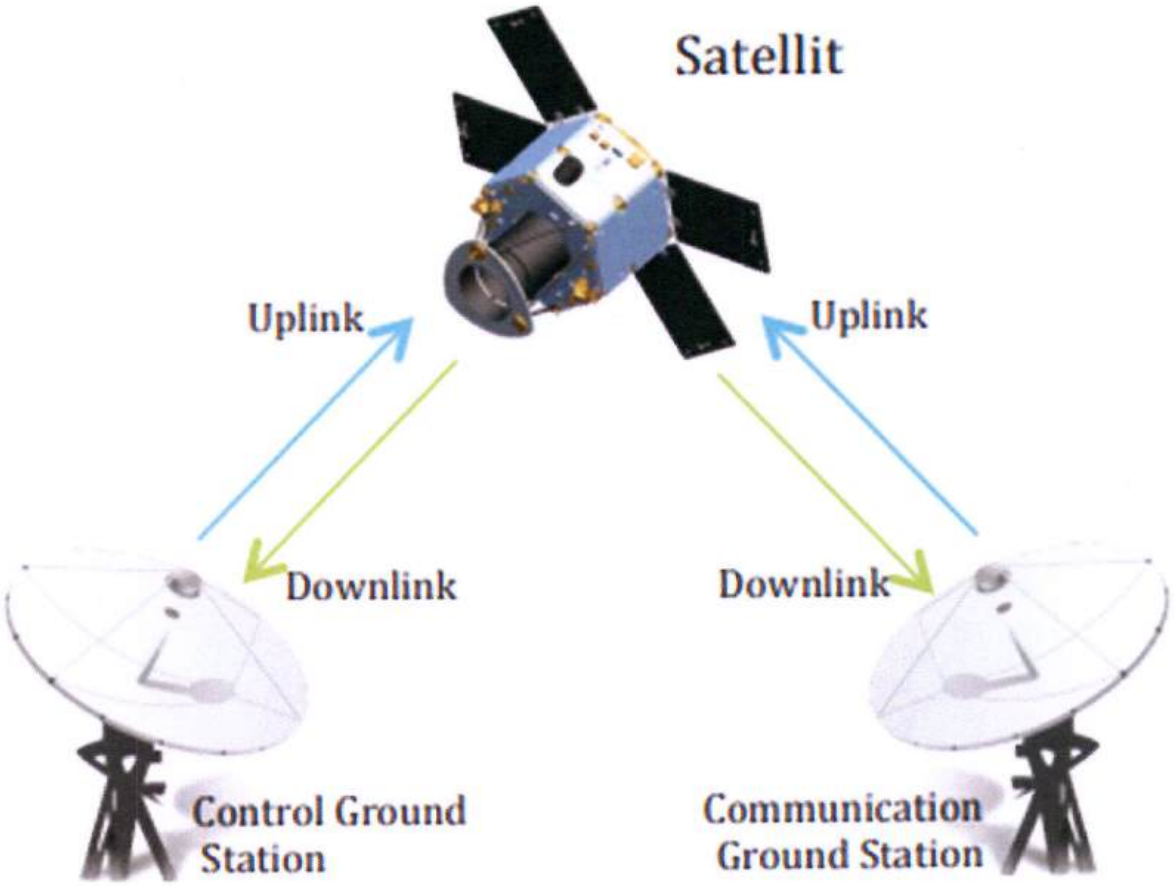
Тому, теоретичні та практичні дослідження направлені на вирішення даних завдань є актуальними і своєчасними.

Об’єкт дослідження – алгоритм шифрування супутникової системи комунікацій

Предмет дослідження – методи визначення ефективності алгоритму аутентифікованого шифрування для супутникових систем

Мета роботи – огляд кіберзагроз супутникових систем та розробка архітектурної оптимізації паралельного алгоритму аутентифікованого шифрування для супутникових застосувань.

Супутники та супутникові системи



Кіберзагрози та їх види

Злом

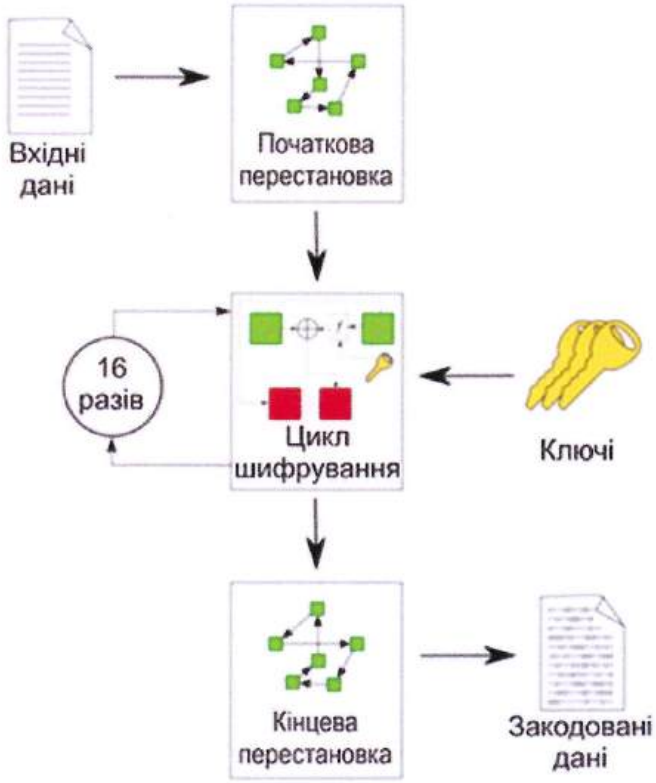
Підробка



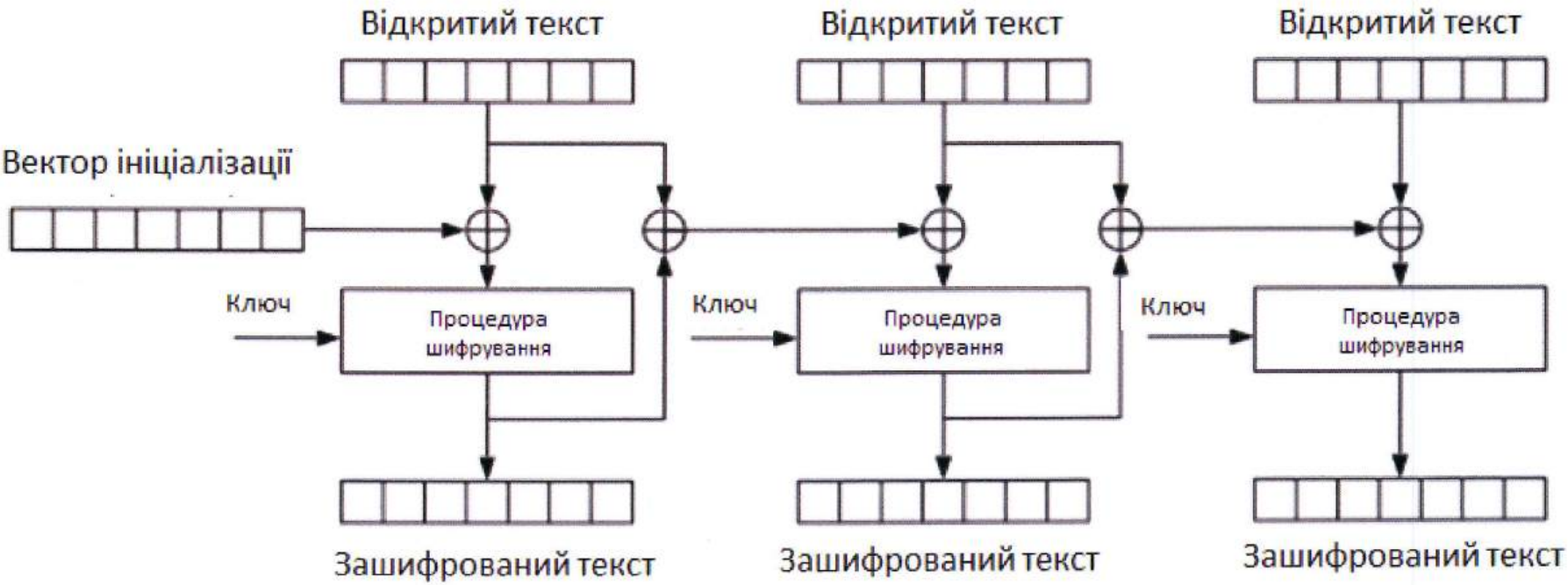
Заглушення

Викрадення

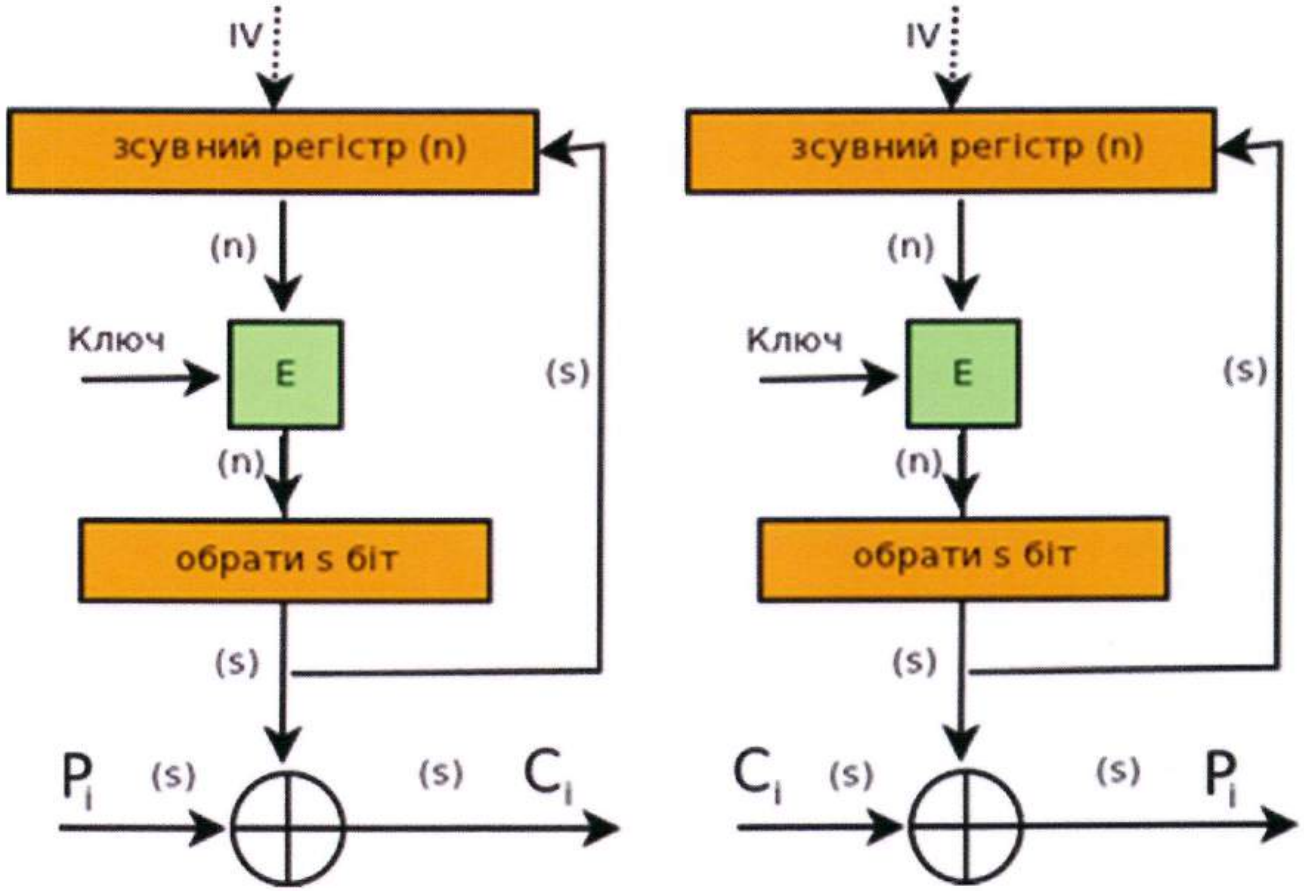
AES – алгоритм симетричного шифрування



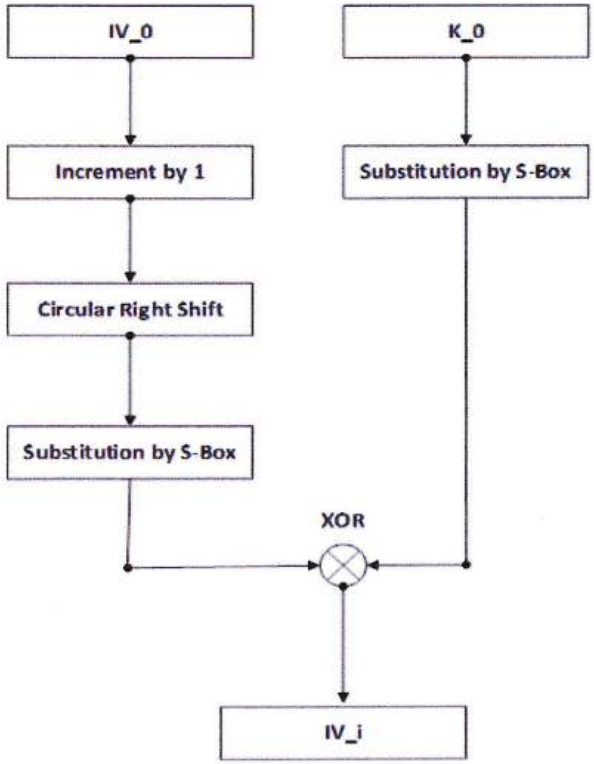
Алгоритм IV генерації



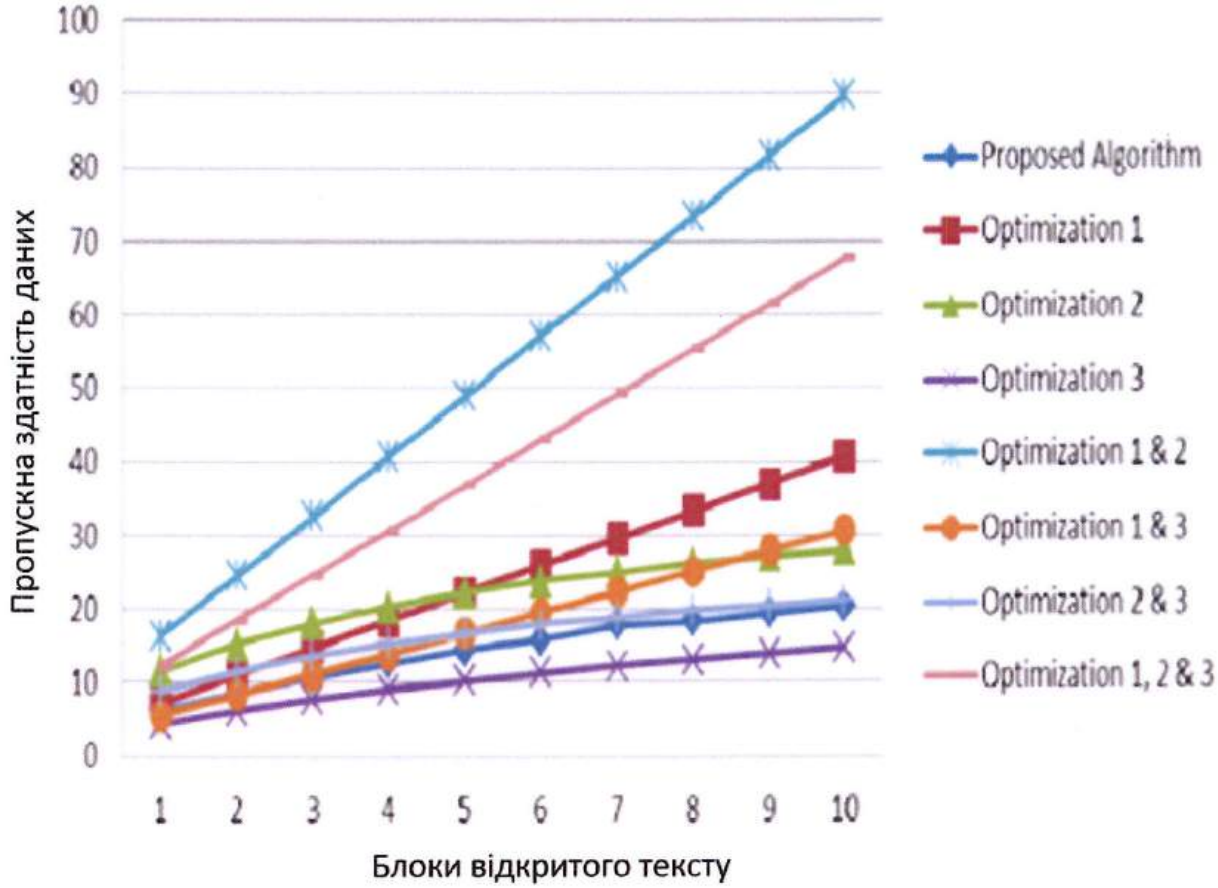
Алгоритм блокового шифру



Алгоритм перемикання ключів



Результати оптимізацій



Порівняння з іншими алгоритмами

Імплементація	Тактова частота (MHz)	Ділянок	Блокова RAM	Пропускна здатність (Gb/s)	Ефективність (Mbps/Ділянку)
B.Yang	271	463328	0	34.7	0.07
S.Lemsitzer	120	27800	0	15.3	0.55
J.Vliegen	119	38241	0	15.2	0.39
G.Zhou	305	8077	0	39.0	4.83
L.Hanzen	233	18505	0	48.8	2.63
K.M. Abdellatif	264	7475	0	31.4	4.19
K.M. Abdellatif 2	232	5512	0	29.7	5.38
Y. Zhang	381	6482	0	48.8	7.54
Koteshwara GCM	50	4087	0	4.1	1.02
Koteshwara GCM-SIV	50	4262	0	3.9	0.93
Запропонований метод	318	7136	55	20.3	2.85
Оптимізація 1 & 2	318	7136	55	89.7	12.57
Оптимізація 1, 2 & 3	240	15458	0	67.6	4.37

Висновки

У даній роботі був представлений і детально розглянутий новаторський паралельний AE-алгоритм, який володіє важливими перевагами в області безпеки та ефективності. Запропонований алгоритм вирішує проблему зловживання нонсом шляхом використання алгоритму генерації IV та рандомізації секретного IV, що забезпечує надійний захист від атак, пов'язаних із використанням нонсів.

Додатково, запропонований алгоритм стійкий до побічних каналів атак, таких як атака DPA, завдяки використанню алгоритму переключення ключів. Це робить його ефективним рішенням для застосувань, де безпека та захист від різноманітних атак є критично важливими.

Три запропоновані оптимізації значно підвищують пропускну здатність даних, роблячи алгоритм ефективним у використанні. Легкість реалізації та захист від космічних випромінювань роблять його привабливим для реальних застосувань в космічних системах. Оптимізації були спрямовані на максимізацію функціональності та стійкості алгоритму.

Апробація результатів магістерської роботи

- Основні наукові результати магістерської роботи опубліковано на п’ятій всеукраїнській науково-практичній конференції «TELECOMMUNICATION: PROBLEMS AND INNOVATION», Київ: ДУІКТ, 20 грудня 2023 р.
- Основні наукові результати магістерської роботи опубліковано в науковому журналі “Наукові записки Державного університету інформаційно-комунікаційних технологій, – 2023, – №2”.

Протокол аналізу звіту подібності науковим керівником

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Микита Клименко

Назва: Методи запобігання кіберзагрозам та супутниковим конфліктам у космічному просторі

Координатор: Анатолій МАКАРЕНКО

Підрозділ: ННІТ

Коефіцієнт подібності 1:1.3

Коефіцієнт подібності 2:0.2

Тривога: 2

Після аналізу Звіту подібності констатую наступне:

☑ виявлені в роботі запозичення є сумлінними і не мають ознак плагіату. Тому робота визнається самостійною і допускається до захисту;

☐ виявлені в роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і самостійності її автора. Роботу направити на доопрацювання;

☐ виявлені в роботі запозичення є недобросовісними і мають ознаки плагіату або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень. У зв'язку з чим, робота не допускається до захисту.

....

Дата *Підпис Наукового керівника*