МІЖНАРОДНИЙ ГУМАНІТАРНИЙ УНІВЕРСИТЕТ
Факультет кібербезпеки, програмної інженерії та комп'ютерних наук
Кафедра комп'ютерної інженерії та інноваційних технологій

Пояснювальна записка

до кваліфікаційної роботи
другого (магістерського) рівня

на тему **ДОСЛІДЖЕННЯ КРИПТОГРАФІЧНИХ МЕТОДІВ ЗАХИСТУ ІНФОРМАЦІЇ**

Виконав: студент 2 курсу, групи КТК-2.1
спеціальності 125 Кібербезпека

Аністратенко М. В.

Керівник Йона Л.Г.

Рецензент Григор'єва Т. Л.

Одеса – 2023

Д О В І Д К А

кафедри КІ та ІТ про виконану магістерську роботу

студента 2 курсу ФКПІ та КН групи КТК-2.1

Аністратенко Матвія Віталійовича

на тему Дослідження криптографічних методів захисту інформації

Висновок нормоконтролера позитивно зачислено до кваліфікаційної роботи виконано з незначними порушеннями ДСТУ. Оформлено згідно вимог внутрішнього положення НТУ

Нормоконтролер к.т.н., доцент _______ Педещ В.В.

(науковий ступінь, вчене звання, посада) (підпис, дата) (і. б. прізвище)

Висновок відповідального за наявність плагіату згідно з сертифікатом ID 1015703033 унікальність роботи підтверджено

Відповідальна особа к.т.н., доцент _______ Педещ В.В.

(науковий ступінь, вчене звання, посада) (підпис, дата) (і. б. прізвище)

Попередня експертиза (захист) магістерської роботи

(бакалаврської роботи чи магістерської роботи)

студ. Аністратенко М. В. проведена "12" грудня 2023 р.

(прізвище і.б.)

Висновки Виконання МР відповідає завданню, усі пункти виконано якісно та згідно вимог до оформлення. Оригінальність роботи: досліджено швидкодію сучасних алгоритмів шифрування та проведено порівняльний аналіз в залежності від обраних параметрів шифру. Розглянуто процес обрання оптимальних параметрів протоколу рукостискання у версіях 2.0 та 3.0 TLS. МР відповідає вимогам до ВКР за заявленою спеціальністю 125 Кібербезпека та може бути рекомендована до захисту в ДЕК.

Члени комісії _______ к.т.н., доцент Йона Л.Г.

(підпис) (науковий ступінь, вчене звання, посада, прізвище і.б.)

_______ к.т.н., доцент Педещ В.В.

(підпис) (науковий ступінь, вчене звання, посада, прізвище і.б.)

_______ викл. каф. КІ та ІТ Швець О.В.

(підпис) (науковий ступінь, вчене звання, посада, прізвище і.б.)

МІЖНАРОДНИЙ ГУМАНІТАРНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, програмної інженерії та комп'ютерних наук

Кафедра комп'ютерної інженерії та інноваційних технологій

Освітній ступінь магістр

Галузь знань 12 Інформаційні технології

Спеціальність 125 Кібербезпека

ЗАТВЕРДЖУЮ
Завідувач кафедри КТта ІТ
к.т.н., доц. Л.Г.Йона
"15" 09 2023 року

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ РОБОТУ

Аністратенко Матвію Віталійовичу

1. Тема роботи: Дослідження криптографічних методів захисту інформації
керівник роботи доцент Йона Л. Г.
затверджені наказом закладу вищої освіти від 25.09.2023 р. № 1951

2. Строк подання студентом роботи 11.12.2023

3. Вихідні дані до роботи: Дослідити сучасні методи криптографічного захисту інформації за призначенням. Привести результати аналізу криптосистем.

4. Зміст розрахунково-пояснювальної записки
Розділ 1: Засоби захисту інформації у телекомунікаціях.
Розділ 2: Сучасні симетричні алгоритми шифрування.
Розділ 3: Алгоритми ідентифікації та автентифікації.
Розділ 4: Аналіз характеристик алгоритмів криптографічного захисту інформації

5. Перелік графічного матеріалу (з зазначенням обов'язкових креслень)
Слайд 1 – Титульний слайд

Слайд 3 – Призначення криптографічних алгоритмів
Слайд 4 – Криптостійкість симетричних алгоритмів шифрування
Слайд 5 – Порівняння симетричних стандартів шифрування
Слайд10 - Порівняння швидкодії симетричних алгоритмів
Слайд12 - Порівняльний аналіз версій протоколу TLS
Слайд 13 -Протокол повного рукостискання TLS 1.3

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Завдання видав	Завдання прийняв

7. Дата видачі завдання 26.09.2023

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1	Вступ	25.09.23 – 04.10.2023	Вик
2	Розділ 1 Засоби захисту інформації у телекомунікаціях	05.10.23 – 11.10.2023	Вик
3	Розділ 2 Сучасні симетричні алгоритми шифрування	12.10.23 – 25.10.2023	Вик
4	Розділ 3 Алгоритми ідентифікації та автентифікації	26.10.23 – 04.11.2023	Вик
5	Розділ 4 Аналіз характеристик алгоритмів криптографічного захисту інформації	05.11.23 – 14.11.2023	Вик
6	Висновки та рекомендації	15.11.23 – 25.11.2023	Вик
7	Перелік джерел посилання, Додаток А	26.11.23 – 08.12.2023	Вик

Студент (підпис) М. В. Аністратенко

Керівник роботи (підпис) Л. Г. Йона

ВІДГУК

на магістерську роботу студента Аністратенко М. В.

на тему: «Дослідження криптографічних методів захисту інформації»

Тема магістерської роботи здобувача Аністратенко М. В. є актуальною та пов'язана з проблемою захисту інформації при передаванні її по каналах зв'язку. Криптографія вивчає методи захисту інформації, що передається загальнодоступними каналами. При цьому каналом зв'язку передається вже не сама інформація, а результат її перетворення за допомогою шифрування, що не дозволяє неправочинному користувачеві прочитати її без знання ключа шифрування. Проте, поява нових потужних комп'ютерів та технологій мережевих обчислень уможливила дискредитацію криптографічних систем, які ще донедавна вважалися за стійкі.

В магістерській роботі надається результат дослідження сучасних криптографічних методів та аналіз систем захисту інформації від неправочинних користувачів. Результати дослідження представлені у тезах доповіді «Дослідження методів автентифікації користувача» на IX Всеукраїнській науково-практичній конференції студентів, аспірантів та молодих вчених «Гуманітарний і інноваційний ракурс професійної майстерності: пошуки молодих вчених».

В процесі виконання магістерської роботи здобувач Аністратенко М.В. показав добру підготовку з питань статистичної теорії зв'язку, знання систем та пристроїв обробки інформації, уміння працювати з літературою.

Рівень підготовки Аністратенко М.В. заслуговує оцінки „відмінно".

Вважаю, що здобувач Аністратенко М.В. заслуговує присвоєння за заявленою спеціальністю 125 Кібербезпека кваліфікації магістр з кібербезпеки.

Керівник, к.т.н., доц. кафедри
Комп'ютерної інженерії
та інноваційних технологій Йона Л. Г.

РЕЦЕНЗІЯ

на магістерську роботу студента Аністратенко М. В.

на тему: «Дослідження криптографічних методів захисту інформації»

У магістерській роботі студента Аністратенко М. В. розглянуто сучасні алгоритми, які використовуються для захисту інформації при передаванні її телекомунікаційними каналами зв'язку.

Актуальність питання полягає в тому, що в роботі досліджуються сучасні алгоритми криптографічного захисту інформації за їх призначенням. Крім того, проведено аналіз щодо оптимального алгоритму шифрування, які можуть використовуватися в сучасних криптографічних системах. Текстова частина магістерської роботи викладена послідовно, чітко, технічно та грамотно.

Проте в роботі є деякі недоліки:

- не розглянуто аналіз асиметричних алгоритмів шифрування;
- розглянуті не всі сучасні алгоритми електронного підпису;

Але вказані недоліки не знижують цінності виконаної роботи.

Магістерська робота відповідає вимогам до випускних кваліфікаційних робіт магістрів та заслуговує оцінки «відмінно».

Здобувач Аністратенко М. В. заслуговує присвоєння за заявленою спеціальністю 125 Кібербезпека кваліфікації магістр з кібербезпеки.

Рецензент
завідувачка кафедри
Інформаційних технологій,
к.т.н., доцент Т.І.Григор'єва

UNICHECK
by Turnitin

Ім'я користувача:
Анна Серединко

ID перевірки:
1016015999

Дата перевірки:
18.12.2023 09:41:04 EET

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
18.12.2023 10:06:02 EET

ID користувача:
100001433

Назва документа: итог диплом Аністратенко

Кількість сторінок: 61 Кількість слів: 10585 Кількість символів: 79271 Розмір файлу: 1.07 MB ID файлу: 1015703033

24.9% Схожість

Найбільша схожість: 4.99% з Інтернет-джерелом (https://studfile.net/preview/5157331/page:5)

24.4% Джерела з Інтернету	962	Сторінка 63
3.05% Джерела з Бібліотеки	36	Сторінка 69

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи	11

РЕФЕРАТ

Текстова частина магістерської роботи: 62 с., 9 рисунків, 8 таблиць, 1 додаток, 13 джерел.

АВТЕНТИФІКАЦІЯ, АЛГОРИТМИ ШИФРУВАННЯ, ЗАХИСТ ІНФОРМАЦІЇ, КЛЮЧ ШИФРУВАННЯ, КРИПТОГРАФІЧНІ МЕТОДИ ЗАХИСТУ ІНФОРМАЦІЇ, КОНФІДЕНЦІЙНІСТЬ, ЦІЛІСНІСТЬ.

Об'єкт дослідження – сучасні криптографічні методи захисту інформації

Мета роботи – провести дослідження криптографічних алгоритмів, які забезпечують захист інформації у телекомунікаціях.

Метод дослідження– аналітичний з використанням комп'ютерних технологій.

У магістерській роботі проведено аналіз методів захисту інформації, дослідження швидкодії сучасних алгоритмів шифрування та обрано оптимальні параметри протоколу повного рукостискання.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ УМОВНИХ ПОЗНАК

ЕП — електронний підпис
ІКС — інформаційно-комунікаційна систем
КЗЗ — комплекс засобів захисту
КС — комп'ютерна система
КСЗІ — комплексна система захисту інформації
НСД — несанкціонований доступ
ПЗ — програмне забезпечення
ПЕОМ — персональна електронна обчислювальна машина
СЗІ — система захисту інформації
СУБД — система управління базами даних
CA — Certificate Authority (цент сертифікації)
DES — Data Encryption Standart (стандарт шифрування даних)
IP — Internet Protocol (міжмережний протокол)
KDC — Key Distribution Center (центр розподілу ключів)
PAP — Password Authentication Protocol (протокол парольної автентифікації)
PIN — Private Identification Number (персональний ідентифікаційний номер)
PKI — Public Key Infrastructure (інфраструктура відкритих ключів)
SSL — Secure Sockets Layer (рівень захищених сокетів)
SSO — Single Sign-On (система однократної автентифікації)
TCP — Transport Control Protocol (протокол управління передаванням)
VPN — Virtual Private Network (віртуальні приватні мережі)

ВСТУП

Основним чинником сьогодення, що впливає на забезпечення конфіденційності інформації є ступінь захищеності інформації та інформаційного середовища. Це сприяє процесу забезпечення захисту інформації, тобто забезпечення її конфіденційності, цілісності та доступності під час передачі та обробки у різних сферах діяльності людини (збирання, накопичування, модифікація, зберігання, видалення, реєстрація), які відбуваються в певній системі за допомогою програмних і/або технічних засобів.

Під час обробки інформації з обмеженим доступом повинен забезпечуватися її захист від несанкціонованого перегляду, змін, видалення, копіювання, поширення. У разі, коли йдеться про службову та таємну інформацію, то вона має передаватися захищеними каналами зв'язку або здійснюватися за допомогою технічного чи криптографічного захисту інформації. Криптографічний захист інформації реалізується за допомогою програмних, програмно-апаратних та апаратних засобів шляхом перетворення даних з використанням ключової інформації з метою зашифрування/розшифрування змісту, перевірки авторства, справжності, цілісності та доступності інформації.

У зв’язку з підвищенням рівня криміногенного впливу, актуальним є визначення надійного методу захисту конфіденційної інформації. З цією ціллю буде проаналізовано і проведено аналіз актуальних сучасних симетричних і асиметричних алгоритмів захисту інформації та розглянуто програмне забезпечення для надійного збереження конфіденційної інформації на пристрої користувача.

1 ЗАСОБИ ЗАХИСТУ ІНФОРМАЦІЇ У ТЕЛЕКОМУНІКАЦІЯХ

1.1 Призначення криптографічних алгоритмів

Сучасні криптографічні алгоритми мають виконувати три основні напрями захисту інформації, що передається в телекомунікаціях, а саме шифрування інформації, автентифікацію документів, та розподіл ключів. Отже, криптографічні алгоритми можна поділити за призначенням у такій спосіб:

- для забезпечення конфіденційності та доступності інформації використовують алгоритми шифрування;
- для забезпечення цілісності документу використовують алгоритми автентифікації, зокрема електронний підпис під документом;
- для забезпечення процесу управління ключами використовують алгоритми розподілу ключів.

Для забезпечення інформаційної безпеки в телекомунікаціях проводяться різноманітні заходи, що можна об'єднати поняттям - «Система Інформаційного Захисту», яка містить програмно-технічні засоби з функцію протидії загрозам та мінімізує вірогідні збитки користувача.

Способи протидії несанкціонованому витоку інформації поділяють на:

1. Технічні заходи, до яких можна віднести захист від несанкціонованого перехоплення важливих цифрових даних, несанкціонованого доступу, організацію обчислювальних мереж методом перерозподілу ресурсів:

- установка датчиків диму та пожежної сигналізації;
- захист приміщення шляхом за допомогою решіток на вікнах, замків на дверях;
- біометрична/радіочастотна автентифікація;
- системи електроживлення для уникнення втрати важливої інформації під час раптового вимкнення електроенергії;
- системи охоронної сигналізації від крадіжок.

2. Організаційні заходи, до яких можна віднести охорону серверів, підбір працівників, розподілення обов’язків, наявність плану відновлення працездатності сервера після виходу його з ладу, універсальність засобів захисту від усіх користувачів, зокрема керівництва [1].

За методами реалізації засоби захисту інформації можна розділити на три основні групи:

- програмні засоби захисту інформації – програмне забезпечення для захисту обчислювальної системи шляхом обмеження доступу користувачів до конфіденційних даних (ключі і паролі) та забезпечення багаторівневого доступу. Проте цей спосіб є доцільним тільки для локальних мереж;
- програмно-апаратний засіб захисту — це розроблені пристрої, які функціонують на спеціалізованих або універсальних мікропроцесорах, які не потребують модифікацій в схемотехніці при зміні алгоритму функціонування та забезпечують високу ступінь захисту локальної мережі, підключеної до глобальн;
- апаратними засобами називаються пристрої, які захищають від несанкціонованого доступу до інформації шляхом безпосереднього підключення технічних засобів негласного отримання інформації до каналів зв'язку і мережевих апаратних засобів. Недоліком цього методу є достатньо висока вартість завдяки точності і статичності їх роботи.

На сьогоднішній день найбільшу зацікавленість викликають наступні напрямки практичних та теоретичних досліджень:

- перевірка надійності та створення криптографічних протоколів і алгоритмів;
- процес адаптації алгоритмів до різних апаратних і програмних платформ;
- способи і методи використання вже існуючих криптографічних технологій в нових системах;
- реалізація можливості використання криптографічних технологій для забезпечення захисту інтелектуальної власності.

1.2 Класифікація методів криптографічного захисту інформації

Захист інформації, яка передається незахищеними каналами зв’язку відбувається за допомогою криптографічних алгоритмів.

Криптографія — це наука, яка вивчає методи та способи забезпечення

захисту інформації від несанкціонованих змін, втручання при передачі, оброблення та зберігання певної інформації, а також забезпечення секретності інформації. У класичній криптографії використовується тільки одна одиниця секретної інформації – ключ, знання якого дозволяє відправникові зашифрувати інформацію, а одержувачеві – розшифрувати її.

Криптографічним захистом інформації називається вид захисту, який реалізується шляхом перетворення початкової інформації за допомогою спеціальних (ключових) даних для шифрування/відновлення змісту повідомлення, підтвердження її цілісності, справжності, авторського права або ін.

Отже, захист інформації, що передається в телекомунікаціях потребує використання криптографічних алгоритмів.

По-перше, криптографічні алгоритми призначені для шифрування інформації, яка має залишатися конфіденційною.

По-друге, також необхідно захищати цілісність інформації, тобто вона має бути одержана в тому вигляді, в якому її відправили, без модифікації. В цьому випадку застосовуються алгоритми автентифікації.

В обох випадках процедури криптографічного перетворення відбуваються за допомогою використання відповідних ключів.

Криптографічні алгоритми за типом використання ключів можна поділити на два великих класи: симетричні та асиметричні.

Функції гешування (геш-функції) також можна віднести до симетричних криптографічних перетворень, оскільки в ключових функціях гешування використовуються симетричні криптографічні перетворення та ключі.

В свою чергу, криптографічні алгоритми можна класифікувати за призначенням, тобто за методами вирішення завдань криптографії.

Класифікація криптосистем за призначенням зображена на рис.1.1.

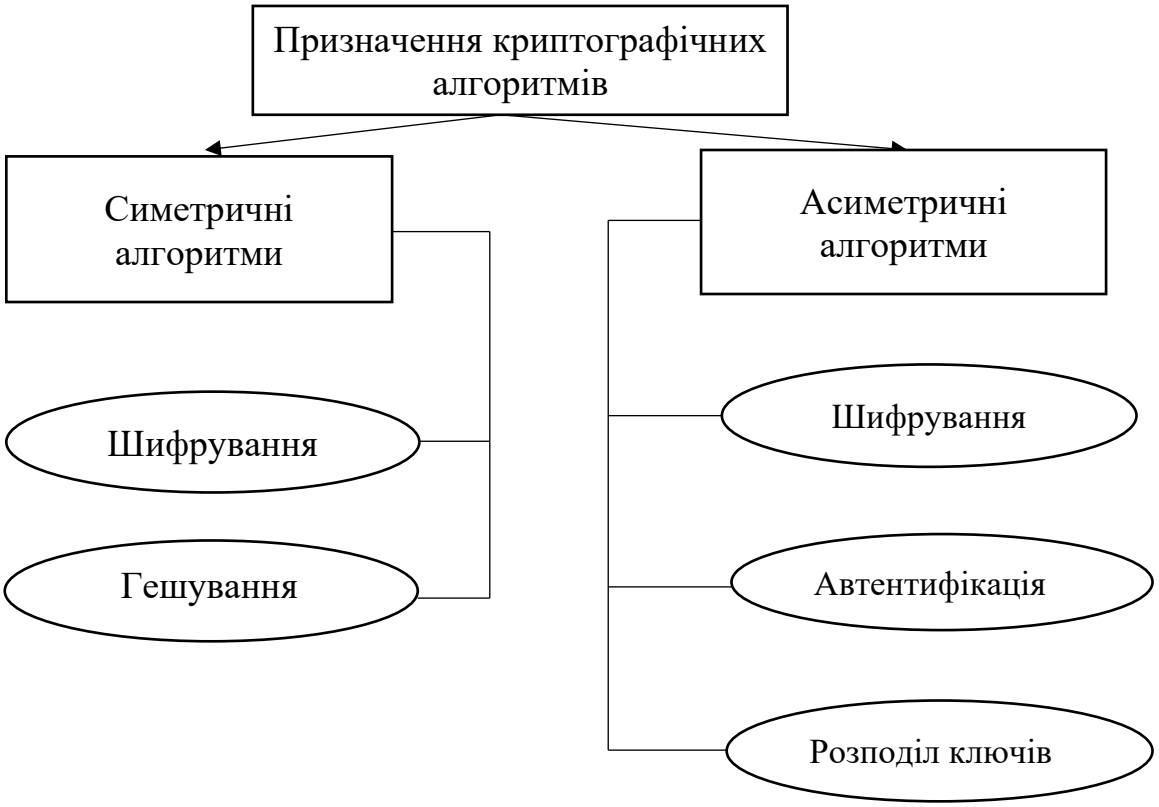


Рисунок 1.1 – Класифікація криптографічних алгоритмів за призначенням.

До завдань криптографічного захисту інформації входить забезпечення:

- конфіденційності, тобто захисту від витоку інформації (вирішується шифруванням);

- доступності, тобто інформація повинна бути доступна тільки тому користувачеві, для якого вона призначена (вирішується шифруванням);

- цілісності, тобто інформація повинна бути захищена від несанкціонованої модифікації (вирішується електронним цифровим підписом);

- автентифікації, тобто підтвердженням справжності (вирішується електронним цифровим підписом і сертифікатом);

- незаперечності, тобто неможливості відмовитися від вчиненої дії (вирішується електронним цифровим підписом і сертифікатом).

1.2.1 Алгоритми шифрування

Для забезпечення конфіденційності інформації виконується процедура шифрування. Процес шифрування поділяється на зашифровування інформації

та розшифровування криптограми. Зашифровування інформації відбувається шляхом перетворення відкритого тексту в незрозумілий текст (криптограму) за допомогою ключа шифрування (секретного параметру). Розшифровування криптограми – це процес зворотний шифруванню, тобто відновлення відкритого тексту із криптограми за допомогою ключа шифрування.

Шифрування захищає дані, які надсилає, отримує та зберігає користувач по телекомунікаційній мережі. Інформація може, наприклад, включати текстові повідомлення, що зберігаються на смартфоні, журнали та банківську інформацію, надіслану через онлайн-рахунок.

Величезна кількість особистої інформації обертається в Інтернеті та зберігається в хмарі або на серверах при постійному з'єднанні з Інтернетом. Так як більшість підприємств переходять на роботу в онлайн режимі, то ніяке підприємство не зможе функціонувати, якщо особисті дані не будуть потрапляти у мережеву комп'ютерну систему організації, саме тому важливо знати, як допомогти зберегти конфіденційність цих даних. Шифрування відіграє важливу роль у цьому процесі.

Існують різні алгоритми криптографічного перетворення інформації. Криптографічні алгоритми можна поділити на 2 класи: симетричні та асиметричні. Алгоритми, які об'єднують обидва класи називаються гібридними.

Симетричні алгоритми характеризуються тим, що шифрування та розшифровування відбувається за допомогою одного спільного секретного ключа шифрування На сьогоднішній день в основі симетричних алгоритмів частіш за все використовуються прості шифри підстановки, перестановки та комбіновані методи, які мають практичне застосування. Різні поєднання цих простих шифрів та секретність ключів, що використовуються для перетворень відкритого тексту на криптограму та навпаки, дають стійкі криптоалгоритми. Серед стійких сучасних симетричних алгоритмів можна виділити криптосистеми 3DES, Американський стандарт шифрування AES, Європейський стандарт шифрування IDEA, Японський стандарт шифрування Camelia, Український стандарт шифрування Калина і т.д.

Асиметричні алгоритми (з відкритим ключем) частіше використовуються для автентифікації документів та розподілу ключів. При цьому, несиметричні алгоритми також можуть використовуватися для процесу шифрування невеликих повідомлень (через те, що вони дуже повільно працюють). Серед асиметричних алгоритмів шифрування можна виділити криптосистеми RSA, Ель Гамаля, Меркли-Хеллмана, Шамира, тощо.

Окремим напрямом розвинення асиметричних алгоритмів є криптосистеми на еліптичних кривих, наприклад алгоритм шифрування Міллера-Кобліца.

Актуальними є гібридні криптосистеми, що поєднують обидва типи криптографічних алгоритмів. Зазвичай, текст повідомлення зашифровується з використанням симетричної криптосистеми, а секретний ключ (для подальшого застосування симетричною криптосистемою) зашифровується з використанням асиметричної криптосистеми.

Отже, можна зробити висновки щодо використання криптосистем. Основним недоліком симетричного шифрування є необхідність передавати ключі таким чином, щоби вони не були перехоплені злочинником. Недолік цей робить неможливим використання симетричного шифрування в системах з великою кількістю учасників.

Проте, у асиметричних криптосистем є свої недоліки. Перший недолік асиметричних алгоритмів шифрування – це мала швидкість виконання операцій зашифровування і розшифровування, зумовлена тим, що для виконання обчислень необхідно мати велику ємність ресурсів. Другий недолік пов’язано перш за все з задачею дискретного логарифму – поки ще не доведено, що її рішення за припустимий час неможливе. Зайві труднощі завдає і необхідність захисту відкритих ключів від підміни (зловмисник може забезпечити зашифровування повідомлення на своєму відкритому ключі і потім легко розшифрувати його власним секретним ключем).

Враховуючи всі особливості роботи криптосистем, можна зробити висновок, що найбільш придатними для захисту інформації шляхом шифрування є симетричні алгоритми. Це обумовлено тим, що вони мають

високу швидкість шифрування та забезпечують відповідну криптографічну стійкість. Проте асиметричні алгоритми рекомендовано використовувати для автентифікації та розподілу ключів.

1.2.2 Алгоритми ідентифікації та автентифікації

Ідентифікація та автентифікації застосовуються для обмеження доступу випадкових і незаконних суб'єктів (користувачі, процеси) інформаційних систем до її об'єктів (апаратні, програмні та інформаційні ресурси).

Ідентифікація — процедура розпізнавання користувача чи системи.

Автентифікація - процедура встановлення належності користувачеві інформації в системі пред'явленого ним ідентифікатора, тобто це процес підтвердження справжності користувача. За допомогою автентифікації інша сторона переконується, що користувач є дійсно тією людиною, за кого вона себе видає. Автентифікація – це процедура, яка перевіряє, чи має користувач з наданим ідентифікатором право на доступ до ресурсу.

Система автентифікації має отримати від користувача інформацію, що засвідчує його особу, потім перевірити її, та в разі справжності цієї інформації, надати допуск до системи.

Наявність процедур автентифікації або ідентифікації користувачів є необхідною умовою будь-якої захищеної системи.

При побудові систем ідентифікації і автентифікації виникає проблема вибору зони, на основі якої здійснюються процедури ідентифікації і автентифікації користувача. В якості ідентифікаторів може бути:

– пароль чи секретний ключ, або персональний ідентифікатор

– біометричні характеристики людини: відбитки пальців, голос, хода, клавіатурний почерк тощо.

Найбільш поширеними простими і звичними є методи автентифікації, засновані на паролі - конфіденційних ідентифікатори суб'єктів. В цьому випадку при введенні пароля система автентифікації порівнює його з паролем, що зберігається в базі даних. Причому всі дані зберігаються в зашифрованому вигляді. У разі збігу паролів підсистема автентифікації надає доступ до системи.

Парольні методи автентифікації за ступенем змінності паролів діляться на:

– методи, які використовують постійні (багаторазові) паролі;
– методи, які використовують одноразові паролі.

Використання одноразових або мінливих паролів є більш надійним методом парольного захисту.

Останнім часом частіше використовуються комбіновані методи ідентифікації і автентифікації, які потребують не тільки введення пароля, а й токена, що підтверджує справжність користувача.

Автентифікація, яка здійснюється за допомогою захищених механізмів двох або більше типів називається багатофакторною. Наприклад, застосування для автентифікації паролем разом із токеном або біометричної характеристики людини разом із паролем.

Певний вид інформації, що надається суб'єктом системі при його автентифікації називають фактором автентифікації.

Виділяють три фактори автентифікації, що використовуються в різних комбінаціях: на основі знання чого-небудь, володіння чимось та на основі біометричних характеристик.

Наприклад:

– перший фактор автентифікації «на основі знання» може використовувати для підтвердження справжності пароль чи PIN-код;
– другий фактор автентифікації «на основі володіння» може використовувати для підтвердження справжності якийсь фізичний ключ, пластикову картку чи OTP-токен;
– третій фактор автентифікації «на основі біометричних властивостей людини» може використовувати для підтвердження справжності відбиток пальця, рисунок сітківки ока, голос, клавіатурний почерк, тощо.

В інформаційних технологіях використовуються такі методи автентифікації:

– однобічна (одностороння) автентифікація, коли клієнт системи для доступу до інформації доводить свою автентичність;

– двобічна (взаємна) автентифікація, тобто автентичність має бути підтверджена і клієнтом, і системою;

– трибічна автентифікація, коли використовується третя сторона для підтвердження справжності кожного користувача.

Якщо в процесі процедури автентифікації справжність користувача підтверджена, то система захисту інформації повинна визначити його повноваження для контролю доступу до ресурсів.

Автентифікація за рівнем інформаційної безпеки ділиться на три види:

– статична автентифікація;

– стійка автентифікація;

– постійна автентифікація.

Статична автентифікація забезпечує захист тільки від несанкціонованих дій в системах, коли інформація не може бути перехоплена, що підтверджує справжність користувача. Прикладом такого виду автентифікації може бути пароль, що не змінюється.

Стійка автентифікація використовує динамічні дані автентифікації, що змінюються на кожний сеанс зв’язку. Прикладом системи стійкої автентифікації можуть бути одноразовий пароль чи електронний підпис. Стійка автентифікація забезпечує захист від атак, де зловмисник може перехопити інформацію і використовувати її в наступних сеансах роботи.

Проте, стійка автентифікація не може забезпечити захист від атак, коли інформація може бути модифікована зловмисником під час передачі .

Прикладом постійної автентифікації є застосування електронних підписів для кожного біта інформації.

Для підтвердження справжності документів використовують електронний підпис.

Електронний підпис призначений для автентифікації (підтвердження справжності) користувача та підтвердження цілісності документа та виконує функцію прив’язки автора до документа (неможливість відмови від

авторства).

Для формування електронного підпису кожного абонента використовують окрему пару ключів – K_1 і K_2. Накладання цифрового підпису на документ відбувається секретним ключем K_1. Перевірка підпису здійснюється одержувачем повідомлення відкритим ключем.

Секретний ключ K_2 відомий лише користувачеві, який підписує документ, а його ідентифікаційний номер *ID* і відповідний відкритий ключ K_1 розміщують у загальнодоступному каталозі для інших абонентів мережі. Це дозволяє будь-якому абонентові мережі перевіряти істинність електронного підпису документів, одержуваних від її власника.

1.2.3 Протоколи розподілення ключів

За процес безпечного розподілення ключів між користувачами або між користувачем і системою доступу відповідають криптографічні протоколи. Системи управління ключами виконують функцію перевірки послідовності використання, збереження та заміни ключів. На практиці такі системи розподіляють не самі ключі, а якийсь параметр (меншого значення та об’єму), на підставі якого користувачі можуть обчислити свої сеансові ключі.

Алгоритми розподілення ключів бувають такого типу:

1. Фізичний метод (за відправлення ключа відповідає довірений кур’єр).

2. За допомогою центра сертифікації, відбувається розподіл ключів доступу до відкритих ключів користувачам та видача секретних ключів. Так як кількість абонентів мережі збільшується, то розподілення секретних ключів по відкритому каналу зв’язку за допомогою центрів розподілення ключів, стало проблемою забезпечити надійний та довгостроковий обмін секретними ключами. Ця проблема була вирішена алгоритмом Діффі-Хеллмана, при якому використовується незахищений канал зв’язку. Абоненти отримують спільний ключ, перевіряють правильність обчислення та проходять додаткову автентифікацію, щоби бути впевненими, що відкритий ключ був відправлений саме від того користувача, хто має брати участь в цьому сеансі зв’язку. Цей процес необхідно виконувати, якщо між відправником повідомлення та

одержувачем немає можливості передачі асиметричних ключів шляхом збереження конфіденційності, тобто необхідно пам’ятати про можливість перехоплення ключа зловмисником. Щоби зловмисник не зміг скористатися перехопленим ключем, можна цей відкритий ключ захистити своїм електронним підписом.

1.2.4 Алгоритми гешування

Геш-функцією називається перетворення h інформаційного повідомлення M на послідовність фіксованої довжини $h(M)$, що є геш-кодом. Функція гешування може служити як криптографічна контрольна сума – код виявлення змін (Manipulation Detection Code – MDC) або для перевірки цілісності повідомлення (Message Integrity Check – MIC).

При цьому, функції гешування також можна віднести до симетричних криптографічних перетворень, оскільки в ключових функціях гешування використовуються саме симетричні криптографічні перетворення та ключі.

Всі існуючі функції гешування можна поділити на два великих класи:

1) безключові геш-функції, що залежать тільки від повідомлення;

2) геш-функції із секретним ключем, що залежать як від повідомлення, так і від секретного ключа.

Геш-функцій можна класифікувати за методами побудови:

1) за принципом складного математичного обчислення;

2) на підставі блокових криптосистем;

3) побудована із самого початку.

По використовуваних внутрішніх перетвореннях функції гешування можна поділити на:

– геш-функції, що застосовують бітові логічні перетворення, різні зрушення і, як правило, є багатоцикловими;

– геш-функції, що базуються на блокових симетричних шифрах;

– геш-функції, що здійснюють перетворення в групах, полях і кільцях з цілочисловими чи поліноміальним базисом;

– геш-функції, що використовують матричні перетворення.

Визначальними вимогами до функцій гешування є їхня стійкість до пошуку першого прообразу, другого прообразу, а також стійкість до колізій. Стійкість до пошуку першого прообразу – відсутність ефективного поліноміального алгоритму обчислення зворотної функції, тобто не можна відновити повідомлення M за відомою його геш-функцією $h(M)$, за реальний час (незворотність). Це властивість еквівалентна тому, що геш-функція є односторонньою функцією.

Стійкість до пошуку другого прообразу (колізій першого роду) – обчислювально неможливо, знаючи повідомлення M та його геш-функцію $h(M)$, знайти таке інше повідомлення $M \neq M$, для якого виконувалася б умова $h(M) = h(M)$.

Стійкість до колізій (колізій другого роду) – неможливість побудови двох повідомлень, для яких вироблялося б однакове значення геш-функції, тобто для заданої геш-функції h обчислювально неможливо знайти два повідомлення M і M, $M \neq M$, для яких виконувалася б умова $h(M) = h(M)$.

Найбільш відомі алгоритми отримування геш-образів повідомлень – MD2, MD4, MD5, MD6, SHA-1, SHA-2, SHA-3 (Keccak), RIPEMD, TIGER, HAVAL, Whirlpool, Tiger.

В Україні також уведено в дію та застосовуються власні стандарти, які гармонізовані з міжнародними документами:

– ДСТУ 7564-2014 «Інформаційні технології. Криптографічний захист інформації. Функція гешування»;

– ДСТУ ISO/IEC 10118-2:2015 «Інформаційні технології. Методи захисту. Геш-функції. Частина 2. Геш-функції, що використовують n-бітний блоковий шифр» (ISO/IEC 10118-2:2010; Cor 1:2011, IDT);

– ДСТУ ISO/IEC 10118-4:2015 «Інформаційні технології. Методи захисту. Геш-функції. Частина 4. Геш-функції, що використовують модульну арифметику» (ISO/IEC 10118-4:1998; Cor 1:2014; Amd 1:2014, IDT).

2 СУЧАСНІ СИМЕТРИЧНІ АЛГОРИТМИ ШИФРУВАННЯ

Криптографічні системи за типом алгоритму шифрування оділяютьсяна два великих класи: симетричні і асиметричні.

Якщо для зашифрування і розшифрування інформації використовується один ключ, то такі алгоритми називається симетричними (криптосистеми з секретним ключем).

Симетричні криптографічні перетворення поділяються на блокові шифри та потокові шифри.

У блокових шифрах символи відкритого тексту поділяються на блоки фіксованої довжини з подальшим зашифровуванням кожного блока окремо. При цьому однаковим зашифрованим блокам буде відповідати однаковий шифртекст.

У потокових шифрах кожен символ відкритого тексту зашифровується незалежно від інших і розшифровується у такий самий спосіб .

Розглянемо основні характеристики сучасних симетричних алгоритмів шифрування.

2.1 Стандарт шифрування даних США, криптосистема AES

Захист інформації, який повинен забезпечити стандарт був найвищим пріоритетом для створення алгоритму AES. У 1998 р., Національний інститут стандартів і технологій (NIST) опублікував запит на заміну стандарту шифрування, в якому описувався передбачуваний “Вдосконалений стандарт шифрування” (Advanced Encryption Standard – AES), що повинен прийти на зміну першого стандарту шифрування DES.

AES - це стандарт, який ґрунтується на симетричному блоковому алгоритмі Rijndael, використовує структуру типу SP-мережі (підстановочно-перестановочна мережа) та має архітектуру Square(квадрат). Згідно початкового алгоритму Rijndael допускається довжина блоку відкритих даних 128біт та довжина ключа від 128 до 256 біт з кроком в 32 біти [2].

Відповідність між довжиною ключа, розміром блока даних і кількістю раундів у стандарті AES показано у таблиці 2.1.

Таблиця 2.1 - Відповідність між довжиною ключа, розміром блока даних і кількістю раундів у стандарті AES.

Стандарт	Довжина ключа біт	Розмір блока даних біт	Кількість раундів
AES-128	128	128	10
AES-192	192	128	12
AES-256	256	128	14

Для позначення вибраних довжин input (введення), State (стан) і Cipher Key (ключ шифру) в 32-бітових словах використовується нотація $N_b = 4$ для input і State, $N_k = 4, 6, 8$ для Cipher Key відповідно для різних довжин ключів.

На початку шифрування input копіюється в масив State за правилом:

$$\text{State [r, c]} = \text{Input [r + 4c]}, \qquad (2.1)$$

для $0 \le r < 4$ та $0 \le c < N_b$.

Після цього до State застосовується процедура Add Round Key (додати циклічний ключ) і потім State проходить процедуру трансформації (раунд) 10, 12, або 14 разів (залежно від довжини ключа), при цьому потрібно врахувати, що останній раунд дещо відрізняється від попередніх. У результаті, після завершення останнього раунду трансформації, State копіюється в Output (вихід) за правилом:

$$\text{Output } [r + 4c] = \text{State } [r, c], \qquad (2.2)$$

для $0 \le r < 4$ та $0 \le c < N_b$.

Таким чином, в алгоритмі AES використовуються 4 функціональних перетворення:

- Sub Bytes/State (Стан);
- Shift Rows (Зміщення);
- Mix Columns (Перемішування стовпців);
- Add Round Key (Додати циклічний ключ).

Кожна функція призначена для шифрування та є оберненою, тобто розшифрування передбачає застосування обернених функцій в зворотному напрямку. Внутрішні функції визначені в кінцевому полі. Це поле складається з усіх поліномів за модулем $m = f(x)$ незвідного полінома $f(x) = x^8+x^4+x^3+x+1$ над полем F_2.

1. Функція Sub Bytes/State виконує нелінійну підстановку кожного байту (числа x), тобто функція призначена для реалізації нелінійного шифру заміни. Треба пам'ятати, що саме нелінійність – це властивість блокового шифру, яка захищає його від криптоаналізу.

У процедурі SubBytes, кожен байт в State замінюється відповідним елементом у фіксованій 8-бітовій таблиці пошуку, $S; b_{ij} = S(a_{ij})$.

Процедура SubBytes обробляє кожен байт стану, незалежно здійснює нелінійну заміну байтів, при цьому використовує таблицю замін(S - box). Така операція забезпечує нелінійність алгоритму шифрування. Побудова S - box складається з двох кроків. По-перше, робиться узяття зворотного числа в полі Галуа $GF(2^8)$. По-друге, до кожного байта b з яких складається S - box застосовується наступна операція:

$$b'_i = b_i \oplus b_{(i+4)\bmod 8} \oplus b_{(i+5)\bmod 8} \oplus b_{(i+6)\bmod 8} \oplus b_{(i+7)\bmod 8} \oplus c_i, \qquad (2.3)$$

де $0 \le i < 8$, і b_i – i-ий біт b, а c_i – i-ий біт константи $c = 63_{16} = 99_{10} = 01100011_2$.

Таким чином забезпечується захист від атак, грунтованих на простих властивостях алгебри.

2. Функція Shift Rows – це шифр перестановки по рядках, є оберненим перетворенням

У процедурі ShiftRows, байти в кожному рядку State циклічно зсуваються ліворуч. Розмір зсуву байтів кожного рядка залежить від її номеру. При цій трансформації рядка стани зміщуються на r байт по рядку, залежно від номера. Для рядка $r = 0$, для першого рядка r = 1.

3. Функція Mix Columns застосовується до кожної колонки матриці. Кожний стовпчик стану помножується на багаточлен c(x).

MixColumns обробляє стани по стовпчиках, де кожна з них є поліном четвертої міри. Над цими поліномами робиться множення в $GF(2^8)$ за модулем: $x^8+x^4+x^3+x+1$ на фіксований багаточлен $C(x) = 3x^3+x^2+x+2$.

Разом з ShiftRows, MixColumns вносить дифузію в шифр, тобто вони призначені для змішування байтів, розміщених в різних місцях блоку вихідного повідомлення. Суміш байтів, що стоять в різних позиціях блоку повідомлення, розширює розподіл повідомлень.

4. Функція Add Round Key забезпечує необхідну секретну випадковість розподілу повідомлень.

У процедурі AddRoundKey кожен байт стану об'єднується з RoundKey, використовуючи операцію складання за модулем XOR($\oplus$).

Функції застосовуються багаторазово, щонайменше 10 разів при довжині ключа 128 біт. Тобто від довжини ключа залежить кількість етапів шифрування. Так, при довжині ключа 128 біт, 64-розрядний блок відкритого тексту шифрується протягом 10 етапів шифрування. Якщо використовується довжина ключа 192 біта, то кількість етапів шифрування збільшується до 12 раундів. При довжині ключа в 256 біт, шифрування блоку відкритого тексту здійснюється впродовж 14 етапів шифрування.

Для алгоритмів блочного шифрування розроблені різноманітні режими, які забезпечують необхідні властивості блочних шифртекстів. Приміром, випадковість, можливість вирівнювання вихідних повідомлень до будь-якого розміру (щоби довжина шифртексту не була пов'язана з довжиною вихідного тексту), контроль за поширенням помилок, генерація ключа і так далі.

В криптосистемі AES використовується 5 режимів роботи.

1 Режим *Електронної книги кодів* (*electronic codebook, ECB*), він же режим простої заміни.

Повідомлення розбивається на блоки і кожен блок шифрується окремо, здійснюється присвоювання кодових слів, що взяті з шифрувальної книги.

При цьому однакові блоки відкритого тексту шифруються в такі самі блоки шифртексту.

1) Режим *Зчеплювання блоків шифру* (*cipher-block chaining, CBC*).

Усі блоки шифрованого тексту залежать не тільки від відповідного блоку вихідного тексту, а ще й від усіх блоків, які були оброблені до нього, а також від синхропосилання.

3 Режим *Зворотного зв'язку по шифртексту* (*cipher feedback, CFB*) передбачає повернення шифр тексту на вхід алгоритму. Інакше алгоритм цього режиму називають гамуванням зі зворотним зв'язком. При цьому, алгоритм блочного шифрування можна замінити на відповідну односпрямовану геш-функцію.

Так само, як і в попередньому режимі, усі блоки шифрованого тексту залежать не тільки від відповідного блоку вихідного тексту, а ще й від усіх блоків, які були оброблені до нього, а також від синхропосилання.

4 Режим *Зворотного зв'язку по виходу* (*output feedback, OFB*) можливо використовувати для шифрування повідомлень, для яких повторне передавання не є можливим. Інакше алгоритм цього режиму називають гамуванням. При цьому, алгоритм блочного шифрування можна замінити на відповідну односпрямовану геш-функцію.

5 Режим *Лічильника* (*режим CTR)* передбачає повернення на вхід відповідного алгоритму значення лічильника, що є накопиченням з моменту старту. До поточного ключу і блокам вихідного повідомлення застосовується операція складання за модулем 2. Якщо зворотний звя'зок відсутній, алгоритм шифрування та розшифрування можуть виконуватися паралельно.

Режим Лічильника перетворює блочний шифр в потоковий. Прості лічильники, що на кожному кроці збільшуються на одиницю використовуються найчастіше.

Щодо криптостійкості Advanced Encryption Standard, то вважається, що використовуваний в AES ключ в 128 біт – це надійний захист проти атаки повного перебору.

Обчислення кількості можливих варіантів ключа можна обчислити за формулою:

$$N = 2^n, \qquad (2.4)$$

де n – довжина ключової послідовності.

У Таблиці 2.2 надано можливе число комбінацій з урахуванням розміру ключа, згідно (2.4).

Таблиця 2.2 – Кількість можливих комбінацій з урахуванням розміру ключа

Розмір ключа, біт	Кількість можливих комбінацій
1	2
2	4
4	16
8	256
16	65536
32	4.2×10^9
56 (DES)	7.2×10^{16}
64	1.8×10^{19}
128 (AES, IDEA)	3.4×10^{38}
192 (AES)	6.2×10^{57}
256 (AES)	1.1×10^{77}

У міру збільшення розміру ключа кількість комбінацій зростає експоненціально. Математичні числення доводять, що розмір ключа в 128 біт надійнішим чином захищає від лобової атаки. У таблиці 2.3 показано час, необхідний на злом ключа з урахуванням можливостей сучасних комп’ютерів.

Таблиця 2.3 – Час, що необхідно витратити на злом ключа

Розмір ключа, біт	Необхідний час для злому шифру
56	399 секунд
128	1.02×10^{18} років
192	1.872×10^{37} років
256	3.31×10^{56} років

Тобто розмір ключа в 128 біт забезпечує криптостійкість сучасних алгоритмів шифрування, проте рекомендовано секретну інформацію

шифрувати ключем, довжиною 256 біт.

2.2 Алгоритм шифрування Японії Camellia

Camellia – це алгоритм симетричного блокового шифрування, за параметрами схожий з алгоритмом AES у якого розмір блока - 128 біт, а ключі можуть бути 128, 192, 256 біт.

Camellia являється розробкою японських компаній Nippon Telegraph and Telephone Corporation і Mitsubishi Electric 2000 року. Організація CRYPTREC провела успішну сертифікацію цього алгоритма.

Структура алгоритму Camellia побудована на мережі Фейстеля. Циклова функція використовує нелінійне перетворення S-блоків, блок лінійного розсіювання, операцію додавання за модулем 2 і байтову перестановку.

Шифрування за алгоритмом відбувається згідно схеми Фейстеля з різною кількістю етапів в залежності від розміру ключа:

- 18 етапів, якщо ключ складає 128 біт;

- 24 етапи, якщо ключі - 192 і 256 біт.

В алгоритм шифрування Camellia входять наступні компоненти:

1. F-функція

F-функція обчислюється наступним чином, де S-функція та P-функція зображені на рис.2.1:

$$F : L \times L \rightarrow L$$

$$(X_{(64)}, k_{(64)}) \rightarrow Y_{(64)} = P\,(S\,(X_{(64)} \oplus k_{(64)})) \qquad (2.5)$$

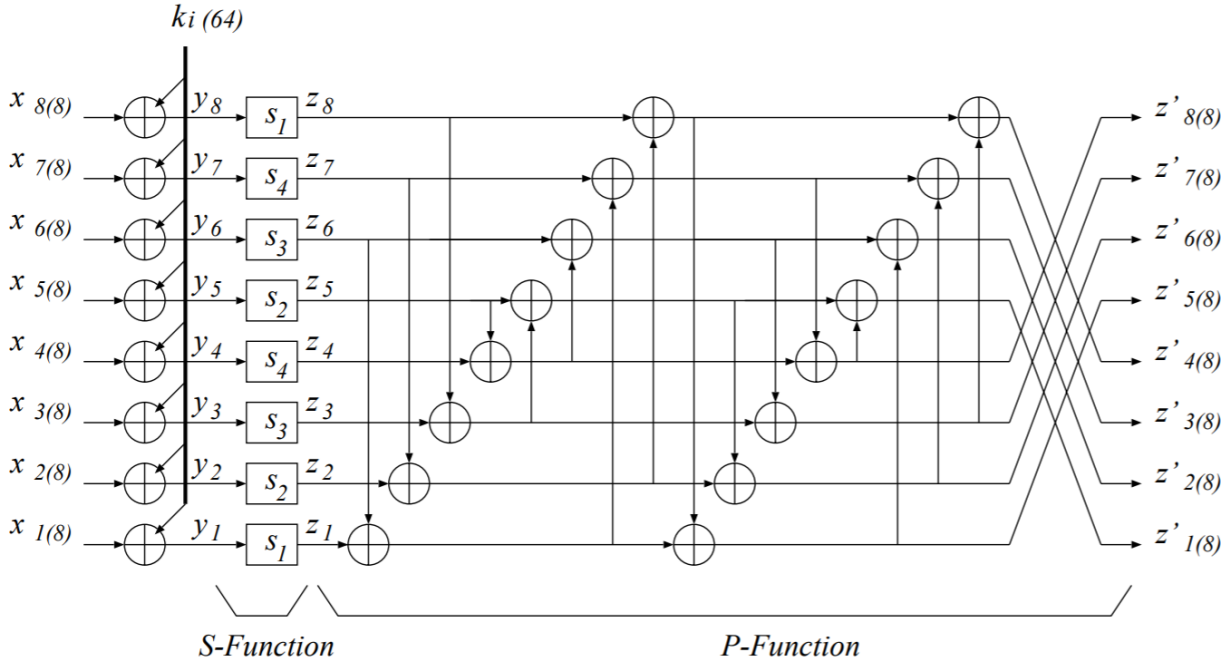


Рисунок 2.1 - S-функція та P-функція

2. FL-функція

FL-функція обчислюється згідно FL : L × L → L

3. FL^{-1}-функція

FL^{-1}-функція обчислюється згідно FL^{-1} : L × L → L

4. P-функція

P-функція є частиною F-функції і обчислюється згідно P : L → L

5. S-функція

S-функція є частиною F-функції і обчислюється згідно S : L → L

Алгоритм розшифрування Camellia ідентичний шифруванню. Різниця зберігається в допоміжних ключах, які змінюють своє місце в залежності від розміру ключа.

Під час криптоаналізу алгоритму Camellia була підтверджена стійкість алгоритму до диференціального і лінійного криптоаналізу, а також до зсувних атак використання зрізаних і неможливих диференціалів, методу інтерполяції та методу бумеранга.

Camellia являється одним з найбільш стійких сучасних алгоритмів шифрування. Відзначається висока швидкість шифрування, зокрема на серверних платформах та швидка процедура розширення ключів.

Але у данного алгоритма є недоліки:

- Camellia істотно програє у швидкості алгоритму Rijndael стандарту AES;

- має високі вимоги до оперативної і енергонезалежної пам’яті [3].

2.3 Стандарт шифрування ДСТУ 7624:2014 «Калина»

Державний стандарт України ДСТУ 7624:2014 «Калина» [4] визначає сучасний алгоритм симетричного блокового шифрування для забезпечення конфіденційності й цілісності інформації при її обробці та встановлює режими його роботи.

Даний алгоритм шифрування враховує досвід і результати проведення міжнародних та відкритих національних конкурсів криптографічних алгоритмів. Алгоритм ДСТУ 7624:2014 забезпечує досить високий рівень криптостійкості порівняно з міжнародним стандартом AES (ISO/IEC 18033-3:2010), оскільки дає можливість застосовування блоку даних і ключа шифрування розміром до 512 біт. На сьогодні це єдиний у світі стандарт блокового шифрування, що підтримує 512-бітові симетричні ключі та дозволяє забезпечити надвисокий рівень захисту навіть секретної інформації, що може дати змогу стати йому основою для створення ефективних засобів криптографічного захисту майбутніх поколінь.

Національний стандарт підтримує змінні величини розміру блоку і довжини ключа шифрування, а саме 128, 256 і 512 біт в різних поєднаннях. Основні параметри шифру, такі як довжина ключа k і блоку даних l, кількість раундів t та кількість стовпців матриці стану c пов’язані залежностями, які представлено у таблиці 2.4.

Таблиця 2.4 – Основні параметри шифру «Калина»

Довжина ключа k, біт	Довжина блоку відкритого тексту, l, біт	Кількість етапів шифрування одного блоку відкритого тексту t	Кількість стовпців матриці стану c

128, 256	128	10	2
256, 512	256	14	4
512	512	18	8

Стандарт ДСТУ 7624:2014 «Калина» забезпечує нормальний, високий та надвисокий рівень стійкості.

При розробці національного стандарту було прийнято рішення забезпечити прозорість проектування та використовувати консервативний підхід з застосуванням добре досліджених конструкцій, які мають забезпечувати запас міцності для можливості використання алгоритму в умовах криптоаналітичних технік, що швидко розвиваються. Даний стандарт розроблено з урахуванням існуючих та потенційних загроз, подальшого інтенсивного розвитку інформаційних технологій та необхідності активного використання протягом кількох наступних десятиліть.

В алгоритмі шифрування даних «Калина» використовуються криптографічні перетворення, які відповідають сучасним вимогам до рівня криптостійкості та швидкодії. Стандарт блокового симетричного шифрування ДСТУ 7624:2014 має десять різних режимів роботи, що широко поширені відповідно до міжнародних стандартів ISO/IEC 10116:2006. Ефективність реалізації систем, засобів та протоколів криптографічного захисту інформації в інформаційно-телекомунікаційних системах різного призначення може бути забезпечена саме наявністю такої кількості режимів роботи алгоритму.

Національний стандарт шифрування ДСТУ 7624:2014 «Калина» належить до SP-мережевих, байт-орієнтованих шифрів.

До блокового шифру «Калина» ставляться такі вимоги:

- високий рівень криптографічної стійкості з достатнім запасом у разі появи нових атак протягом тривалого часу;
- висока швидкість програмної реалізації на сучасних та перспективних платформах;
- прозорість проектування, консервативний підхід до забезпечення стійкості;

- вища (або однакова) ефективність порівняно з найкращими світовими рішеннями.

Оскільки в алгоритмах симетричного блокового шифрування «Калина» та AES («Rijndael») використовуються аналогічні криптографічні перетворення, то можна порівняти їх за основними параметрами [5].

Основні відмінності стандартів «Калина» та AES :

- кількість етапів шифрування (до 18 при довжині ключа 512 біт);
- застосування операції додавання за модулем 2^{64} і за модулем 2 для введення ключів;
- застосування чотирьох S-блоків заміни замість одного (додатковий захист від алгебричних атак, збільшення результату операцій розсіювання – покращені статистичні властивості, відповідно, збільшення рівня стійкості до диференціального та лінійного криптоаналізів тощо);
- використання випадково сформованих чотирьох блоків, відібраних критеріями стійкості до диференціального та лінійного криптоаналізів, ступені нелінійності булевих функцій (на відміну від S-блоку Rijndael/Camellia та інших шифрів, що використовують звернення в полі та, відповідно, квадратичні залежності між входом і виходом, – захист від алгебричних атак);
- принципово нова схема створення підключів (захист від усіх відомих атак на схеми створення підключів);
- досить висока продуктивність;
- можливість відновлення сеансового ключа за окремим підключем (додатковий захист від атак, що виконують відновлення підключів).

Крім того, стандарт «Калина» має аналогічну або навіть більш високу швидкодію на сучасних і перспективних програмних та програмно-апаратних платформах. Усі поліпшення алгоритму спрямовані на збільшення криптостійкості та запобігання потенційним вразливостям відносно алгоритму Rijndael. Основними перевагами шифру порівняно з іншими міжнародними

аналогами є можливість застосовувати блок даних і ключ шифрування розміром до 512 біт.

Алгоритм ДСТУ 7624:2014 «Калина» забезпечує досить високий рівень криптостійкості порівняно з міжнародним стандартом ISO/IEC 18033-3:2010, оскільки дає можливість застосування блока даних і ключа шифрування розміром до 512 бітів. На сьогодні це єдиний у світі стандарт блокового шифрування, що підтримує 512-бітові симетричні ключі та дозволяє забезпечити надвисокий рівень захисту навіть секретної інформації [5].

2.4 Міжнародний алгоритм шифрування IDEA.

Остаточна назва IDEA розшифровується як *International Data Encryption Algorithm* (Міжнародний Алгоритм Шифрування Даних).

IDEA є блоковим симетричним алгоритмом шифрування та передбачає попередній поділ вхідної послідовності M на 64-розрядні блоки, що зашифровуються за допомогою 128-бітового ключа

Алгоритм побудовано на базі модифікованої мережі Фейстеля (рис.2.2). Метою його розробки було створення стійкого криптоалгоритму з простою реалізацією. Алгоритм шифрування є обернений.

Загальною засадою шифрування IDEA, як і у більшості блочних алгоритмів, є змішування та розсіювання. Алгоритм передбачає поєднання операцій основних алгебричних груп, до яких належать:

- XOR (додавання за модулем 2);
- додавання за модулем 2^{16};
- перемножування за модулем $2^{16} + 1$.

Всі перелічені операції виконуються з 16-бітовими підблоками.

Спочатку 64-розрядний блок розбивається на чотири 16-розрядних підблоки X_1, X_2, X_3 та X_4, які є вхідними даними для алгоритму. Усього алгоритм передбачає виконання восьми етапів. На кожному етапі чотири підблоки піддаються операціям додавання за модулем 2, додавання за модулем 2^{16} та множення за модулем $2^{16} + 1$ кожен з кожним та з шістьома 16-бітовими

підключами.

Кожний етап передбачає виконання таких операцій:

1 помножуються X_1 та перший підключ K_1;

2 виконується додавання за модулем 216 X2 та другого підключа К2;

3 виконується додавання за модулем 216 X3 та третього підключа К3;

4 помножуються X4 та четвертий підключ К4;

5 виконується додавання за модулем 2 результатів 1 та 3 кроків;

6 виконується додавання за модулем 2 результатів 2 та 4кроків;

7 помножуються результати кроку 5 та п`ятий підключ К5;

8 виконується додавання за модулем 216 результатів 6 та 7 кроків;

9 помножуються результати кроку 8 та шостий підключ К6;

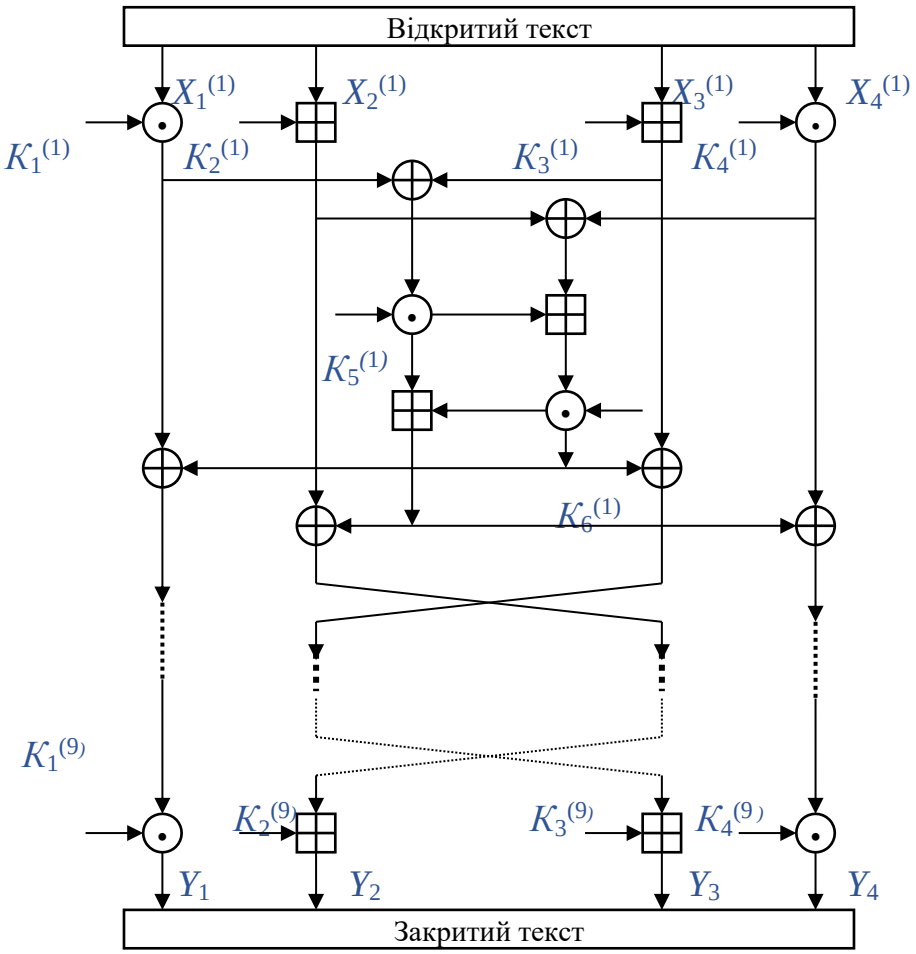
10 виконується додавання за модулем 216 результатів 7 та 9 кроків;

11 виконується додавання за модулем 2 результатів 1 та 9 кроків;

12 виконується додавання за модулем 2 результатів 3 та 9 кроків;

13 виконується додавання за модулем 2 результатів 2 та 10 кроків;

14 виконується додавання за модулем 2 результатів 4 та 10 кроків.



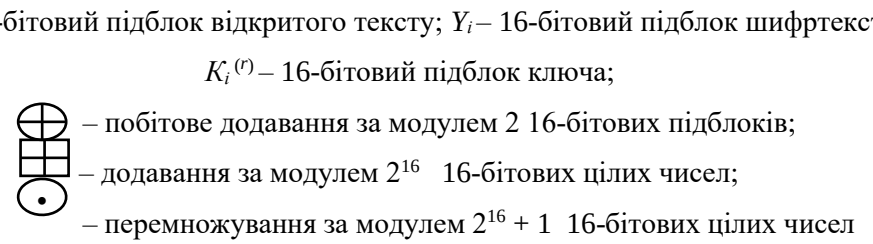


Рисунок 2.2 – Схема шифрування одного блоку вихідного тексту IDEA

Виходом цикла є чотири підблоки, які здобуваються як результат виконання останніх чотирьох кроків алгоритму шифрування. Цикл шифрування завершується переставлянням двох внутрішніх блоків (за винятком останнього циклу). Отже, формуються входи для наступного циклу.

По завершенні останнього, восьмого циклу, виконується вихідне перетворювання:

1 помноження першого підблока X_1 та першого підключа K_1;

2 додавання за модулем 2^{16} другого підблока X_2 та другого підключа K_2;

3 додавання за модулем 2^{16} підблока X_3 та третього підключа K_3;

4 помноження четвертого підблока X_4 та першого підключа K_4.

Здобуті після виконання цих чотирьох операцій результати Y_1, Y_2, Y_3 та Y_4 знову сполучуються, утворюючи шифртекст. Аналогічно шифруються наступні 64-бітові блоки відкритого тексту.

Алгоритм IDEA для шифрування кожного 64-бітного блоку відкритого тексту передбачає використовування 52-х ключів (по шість для кожного з восьми циклів та ще чотирьох для формування вихідного блока).

Спочатку 128-бітовий ключ розбивають на вісім 16-бітових підключів, шість із яких використовуються в першому циклі, а два залишаються для наступного циклу ($K_1^{(1)}$, $K_2^{(1)}$, $K_3^{(1)}$, $K_4^{(1)}$, $K_5^{(1)}$, $K_6^{(1)}$, $K_1^{(2)}$, $K_2^{(2)}$).

Потім 128-бітовий ключ зсовують на 25 біт ліворуч і знову поділяють на вісім ключів ($K_3^{(2)}$, $K_4^{(2)}$, $K_5^{(2)}$, $K_6^{(2)}$, $K_1^{(3)}$, $K_2^{(3)}$, $K_3^{(3)}$, $K_4^{(3)}$). . Перші чотири підключа використовують у другому циклі, останні чотири – у третьому циклі.

Формування наступних ключів виконується аналогічно до завершення

алгоритму, тобто до тих пір, поки не будуть згенеровані 52 16-бітових підключа. Результат формування підключів приведено у таблиці 2.5.

Процес розшифровування супроводжується генерацією ключів у зворотному порядку, причому певні з них змінюються оберненими величинами.

Таблиця 2.5 – Сформовані підключі для кожного етапу шифрування одного 64-бітного блоку даних

Номер етапу шифрування	Сформовані 16-бітові підключі	Лічільник підключів
1	$(K_1^{(1)}, K_2^{(1)}, K_3^{(1)}, K_4^{(1)}, K_5^{(1)}, K_6^{(1)})$	6
2	$(K_1^{(2)}, K_2^{(2)}, K_3^{(2)}, K_4^{(2)}, K_5^{(2)}, K_6^{(2)})$	12
3	$(K_1^{(3)}, K_2^{(3)}, K_3^{(3)}, K_4^{(3)}, K_5^{(3)}, K_6^{(3)})$	18
4	$(K_1^{(4)}, K_2^{(4)}, K_3^{(4)}, K_4^{(4)}, K_5^{(4)}, K_6^{(4)})$	24
5	$(K_1^{(5)}, K_2^{(5)}, K_3^{(5)}, K_4^{(5)}, K_5^{(5)}, K_6^{(5)})$	30
6	$(K_1^{(6)}, K_2^{(6)}, K_3^{(6)}, K_4^{(6)}, K_5^{(6)}, K_6^{(6)})$	36
7	$(K_1^{(7)}, K_2^{(7)}, K_3^{(7)}, K_4^{(7)}, K_5^{(7)}, K_6^{(7)})$	42
8	$(K_1^{(8)}, K_2^{(8)}, K_3^{(8)}, K_4^{(8)}, K_5^{(8)}, K_6^{(8)})$	48
Вихідне перетворення	$(K_1^{(9)}, K_2^{(9)}, K_3^{(9)}, K_4^{(9)})$	52

Алгоритм IDEA може працювати в будь-якому режимі, передбачуваному для блочного шифрування. Він має низку переваг відносно алгоритму DES. По-перше, він є безпечніше, оскільки його ключ є удвічі довше. По-друге, його внутрішня структура є більш складна й тому є більш стійка до криптоаналізу. По-третє, його програмна реалізація удвічі швидша, ніж DES. Завдяки перетворенням, що використовуються у алгоритмі, його можна реалізувати апаратно на підставі інтегральних схем, що забезпечує набагато більшу швидкодію, ніж при програмній реалізації. Саме цю особливість часто використовують при апаратному шифруванні потоку даних у мережах. Алгоритм IDEA застосувався в пакеті програм шифрування PGP (Pretty Good Privacy).

Апаратна реалізація алгоритму IDEA має такі переваги: істотне підвищення швидкості шифрування за рахунок використання паралелізму при виконанні операцій та менше енергоспоживання. Алгоритм IDEA був запатентований у багатьох країнах, однак останній патент закінчився у 2012 році [5].

2 АЛГОРИТМИ ІДЕНТИФІКАЦІЇ ТА АВТЕНТИФІКАЦІЇ

Ефективними системами криптографічного захисту даних є асиметричні криптосистеми. Такі системи також називають криптосистемами з відкритим ключем чи двоключовими системами, тобто в таких системах використовується пара ключів (один публічний, та другий – секретний чи приватний)

Концепція асиметричних криптосистем базується на використовуванні однонапрямлених функцій. Головним критерієм віднесення функції *f* до класу однонапрямлених функцій є відсутність ефективних алгоритмів обернених перетворювань $Y \rightarrow X$.

Другим класом функцій, використовуваних при побудові криптосистем з відкритим ключем, є такі, що називаються однонапрямленими функціями з секретом. Функція називається однонапрямленою з секретом тоді, якщо вона є однонапрямленою й, окрім того, існує змога ефективного обчислювання оберненої функції, коли є відомий „таємний хід” (секретне число, рядок чи інша інформація, яка асоціюється з цією функцією). Саме такою функцією є модульна експонента з фіксованим модулем та показником степені, яку використовують в криптосистемі RSA, яка може працювати як в режимі шифрування, так і в режимі електронного підпису.

Електронний підпис (ЕП) використовується для автентифікації текстів, які передаються комунікаційними каналами. Функційно він є аналогом звичайного рукописного підпису й має основні його переваги:

- засвідчує, що підписаний текст виходить від імені користувача, котрий поставив підпис;
- не надає саме цьому користувачеві можливості відмовитися від зобов’язань, пов’язаних з підписаним текстом;
- гарантує цілісність підписаного тексту.

Електронний підпис являє собою відносно невелику кількість додаткової цифрової інформації, яка передається разом з текстом, що підписується.

Система Електронного підпису складається з двох процедур:

1 формування підпису;

2 перевірка підпису.

Для постановки підпису необхідно застосувати приватний ключ відправника повідомлення, а для перевірки підпису – публічний ключ відправника.

При формуванні ЕП відправник по-перше обчислює геш-функцію $h(M)$ тексту M, який він підписує. Значення геш-функції $h(M)$ - це короткий блок інформації m, який однозначно є залежним від тексту M. Результат гешування m зашифровується приватним ключем користувача, що підписує відкритий текст. Здобувана пара чисел являє собою електронний підпис для тексту M.

3.1 Ідентифікація та автентифікація користувача

Кожний суб'єкт комп'ютерної системи має бути однозначно ідентифікований. Це може бути число, рядок символів чи алгоритм, який ототожнює даний суб'єкт. Цю інформацію називають ідентифікатором.

Суб'єкт – це користувач (людина, програма, процес), який має доступ до об'єкта КС.

Об'єкт – це компонент (комп'ютер, принтер, банкомат), до якого суб'єкт має доступ.

Існує три процедурами захисту при виконанні операцій ініціалізації, що належать до одного об'єкта КС.

Перша процедура, яка виконується в процесі ініціалізації системи, - це ідентифікація, тобто розпізнавання суб'єкта.

Друга процедура – це автентифікація, тобто підтвердження справжності суб'єкта комп'ютерної системи.

Третя процедура - це авторизація, тобто надання певних повноважень суб'єкту комп'ютерної системи.

Автентифікація – це процедура встановлення фактичної належності

ідентифікатора, який надає системі суб’єкт.

При проходженні процедури автентифікації, інформаційна система перевіряє відповідність наданих користувачем даних з тим, що зберігається в її базі даних. По-перше, система має з’ясувати чи існує користувач з таким ім’ям. По-друге, перевіряється збіг введеного користувачем паролю з його обліковим записом. Далі система може затребувати сертифікат, IP-адресу чи додатковий код верифікації. Після отримання правильних відповідей користувача, надається допуск до системи.

Проте процедура автентифікації не обмежується тільки перевіркою справжності користувача. Крім цього на сьогодні є потреба в можливості перевірити справжність електронного документу за допомогою протоколів автентифікації.

2.1Автентифікація документів за допомогою електронного підпису

За обміну електронними документами істотно знижуються витрати на опрацьовування й зберігання документів, зростає швидкість їхнього пошуку. Виникає необхідність автентифікації автора документа й самого документа, тобто встановлення правочинності автора й відсутності змін у здобутому документі. У звичайній (паперовій) формі документообігу ці проблеми розв’язуються за рахунок того, що текст в документі й рукописний підпис автора пов’язані з аркушем паперу, шляхом написання чорнилами. В електронних документах при опрацьовуванні документів такого зв’язку немає. Завданням електронного підпису є поєднати автора і документ, щоби не було можливості відмовитися від авторства та провести процедуру автентифікації з метою підтвердження справжності документу.

Електронний підпис являє собою додаткову інформацію, що представлена в цифровому вигляді, яка формується саме з цього документу секретним ключем та геш-функцією $h(M)$, яка потім надсилається разом з текстом M, що підписується.

Метою автентифікації електронних документів є захист від можливих зловмисних дій, якими є:

- *активне перехоплювання*, порушник, який долучився до мережі, перехоплює документи (файли) і змінює їх;
- *маскарад*, абонент *C* надсилає документ абонентові *B* від імені абонента *A*;
- *ренегатство*, абонент *A* заявляє, що не надсилав повідомлення абонентові *B*, хоча насправді й надсилав;
- *підміна*, абонент *B* змінює чи формує новий документ і заявляє, що одержав його від абонента *A*;
- *повторювання*, абонент *B* повторює раніше переданий документ, що його абонент *A* надсилав абонентові *B*.

Ці зловмисні дії можуть завадити підприємствам, організаціям та особам, які застосовують у своїй діяльності інформаційні технології [5].

Класифікація протоколів автентифікації приведена на рис.3.1

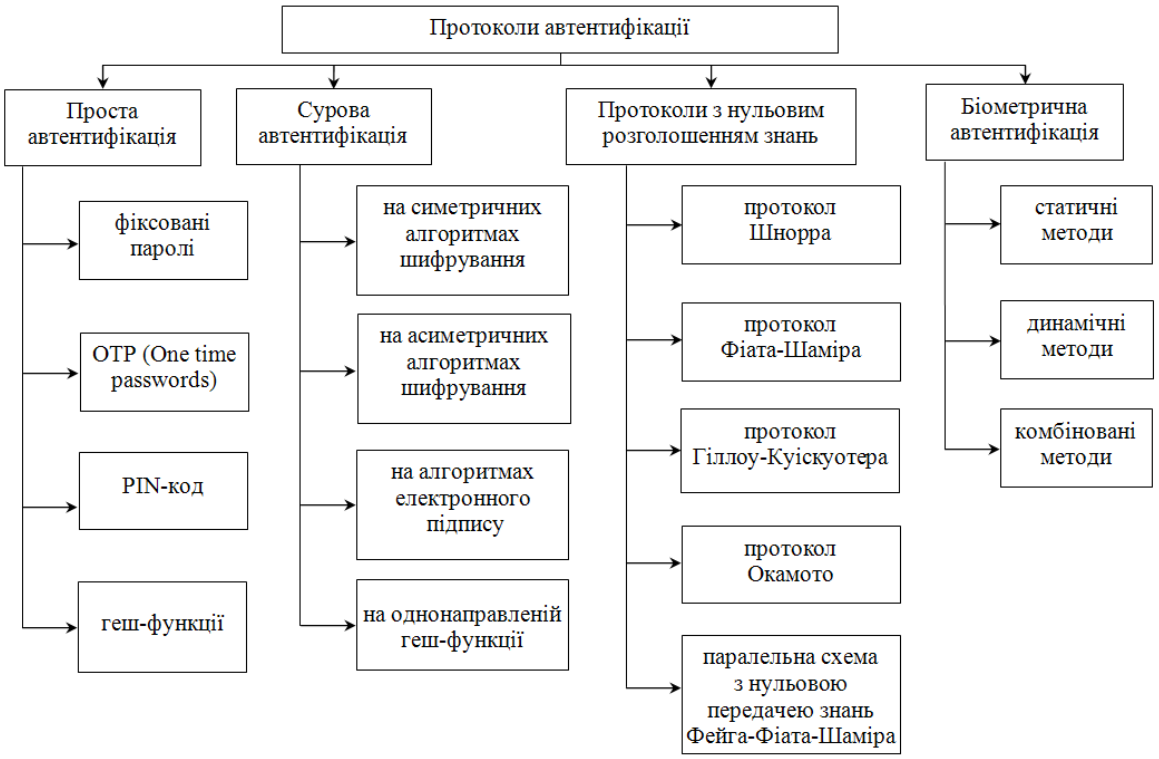


Рисунок 3.1 - Класифікація протоколів автентифікації.

Технологія застосовування системи ЕП припускає наявність мережі абонентів, які надсилають один одному підписані електронні документи. Для кожного абонента генерується пара ключів: секретний і відкритий. Для генерування пари ключів (секретного й відкритого) в алгоритмах ЕП, як і в асиметричних системах шифрування, використовуються різні математичні схеми, що побудовані на застосовуванні однонапрямлених функцій. Ці схеми поділяються на дві групи. В підґрунті такого поділу лежать відомі складні обчислювальні завдання:

- факторизація (розкладання на множники) великих цілих чисел;
- дискретне логарифмування.

Секретний ключ зберігається абонентом у таємниці й використовується ним для формування ЕП. Відкритий ключ призначений для перевірки електронного підпису одержувачем документа та дозволяє перевірити чинність електронного документа та автора. Знаючи відкритий ключ не можна обчислити секретний ключ.

Головним у системі ЕП є те, що неможливо підробити підпис користувача. Підписаний документ утворюється з відкритого файлу, до якого додається електронний підпис, який формується саме з цього файлу.

Кожний підпис містить таку інформацію:

- дата підпису;
- термін завершення дії ключа даного підпису;
- інформація про користувача, який підписав файл (П.І.Б., посада, назва фірми);
- ідентифікатор особи, що поставила підпис (відповідний відкритий ключ);
- власне електронний підпис.

3.2.1 Алгоритм електронного підпису RSA

Алгоритм RSA став першою в світі системою електронного підпису

Щодо процедури обчислення, то спочатку необхідно обрати пару ключів (публічний і приватний ключ). Для цього відправник електронних документів

обирає два простих великих числа P та Q, потім обчислює їхній добуток

$$N = P \cdot Q$$

та функцію Ейлера

$$\varphi(N) = (P-1)(Q-1).$$

Потім відправник обчислює число E з умов

$$E \le \varphi(N), \quad \text{НОД}(E, \varphi(N)) = 1$$

й число D з умов

$$D < N, \quad E \cdot D \equiv 1(\text{mod}\varphi(N)).$$

Пара чисел (E, N) є публічним ключем, який автор передає тому, хто бажає перевірити його підпис. Число D – це приватний ключ для підписування.

Загальну схему формування й перевірки електронного підпису RSA подано на Рис. 3.2.

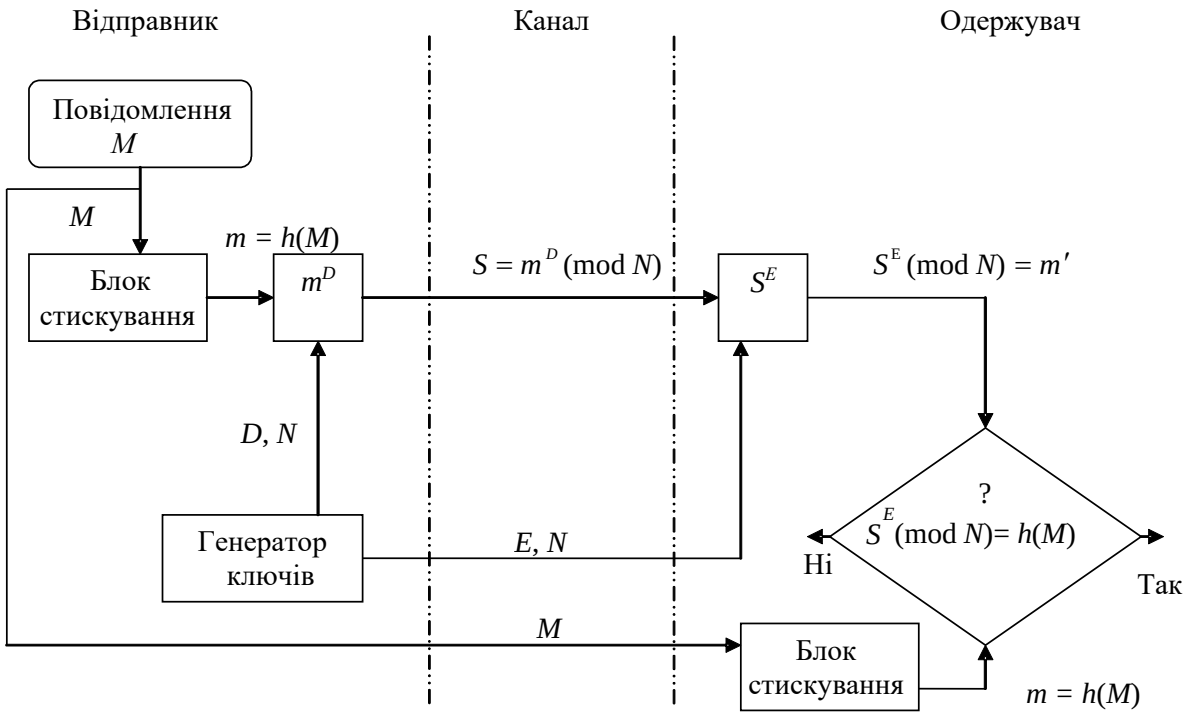


Рисунок 3.2 – Узагальнена схема електронного підпису RSA

Для передачі повідомлення M , що має бути підписане, стискають за допомогою геш-функції $h(\cdot)$ в ціле число m:

$$m = h(M).$$

Електронний підпис S обчислюють під з документу M, шляхом застосування гешу m та приватного ключа D:

$$S = m^D (\text{mod } N).$$

Після прийняття пари (M, S), одержувач повідомлення M обчислює геш-значення двома способами.

Спочатку він відновлює геш m, публічним ключем E:

$$m' = S^E (\text{mod} N).$$

Потім він віднаходить результат гешування прийнятого повідомлення M за допомогою такої самої геш-функції $h(\cdot)$:

$$m = h(M).$$

Якщо виконується рівність обчислених значень, тобто

$$S^E (\text{mod } N) = h(M),$$

то одержувач визнає пару (M, S) чинною. Доведено, що лише власник секретного ключа D може сформувати цифровий підпис S з документа M, а визначити секретне число D завдяки відкритому числу E є не легше, аніж розкласти модуль N на множники.

В літературі ключ E ще називають ідентифікатором особи, що поставила підпис.

3.2.2 Алгоритм електронного підпису Ель-Гамаля

Більш зручний для реалізації на персональних комп'ютерах алгоритм електронного підпису, що був обраний як підґрунтя для національного стандарту США, розроблено Ель Гамалем.

Ель Гамалю вдалося уникнути недоліку алгоритму електронного підпису RSA, пов'язаного з можливістю підробки ЕП під певними повідомленнями без знання секретного ключа. Назва EGSA походить від слів El Gamal Signature Algorithm (алгоритм цифрового підпису Ель Гамаля). Ідея за системою EGSA базовано на тім, що для обґрунтування практичної неможливості фальсифікації електронного підпису може бути використано більш складне обчислювальне завдання, чим розкладання на множники великого цілого

числа, – завдання дискретного логарифмування.

Розглянемо докладніше алгоритм електронного підпису Ель Гамаля. Для того, щоби генерувати пару ключів (відкритий ключ – секретний ключ), спочатку обирають певні великі прості цілі числа P та G, причому $G < P$. Відправник і одержувач підписаного документа використовують при обчислюваннях однакові за довжиною великі цілі числа P та G, що не є секретними.

Відправник обирає випадкове ціле число X, $1 < X \leq (P - 1)$, і обчислює

$$Y = G^X \bmod P.$$

Число Y є відкритим ключем, використовуваним для перевірки підпису відправника.

Число X є секретним ключем відправника для підписування документів і має зберігатися в секреті.

Для того щоб підписати повідомлення M, спочатку відправник гешує його за допомогою геш-функції $h(\cdot)$ в ціле число m:

$$m = h(M), \ 1 < m < (P - 1)$$

і генерує випадкове ціле число K, $1 < K < (P - 1)$, таке що K і $(P - 1)$ є взаємно простими. Потім відправник обчислює ціле число a:

$$a = G^K \bmod P$$

і, застосовуючи розширений алгоритм Евкліда, обчислює за допомогою секретного ключа X ціле число b з рівняння

$$m = X \cdot a + K \cdot b(\bmod(P - 1)).$$

Пара чисел (a, b) є електронним підписом:

$$S = (a, b),$$

який ставиться під документом M.

Три числа (M, a, b) передаються одержувачеві, тоді як пара чисел (X, K) тримається в секреті.

Після прийняття підписаного повідомлення (M, a, b) одержувач повинен перевірити, чи відповідає підпис $S = (a, b)$ повідомленню M. Для цього

одержувач спочатку обчислює за прийнятим повідомленням M число

$$m = h(M),$$

тобто гешує прийняте повідомлення M.

Потім одержувач обчислює значення

$$A = Y^a a^b (\mathrm{mod}\ P)$$

і визнає повідомлення M чинним, лише якщо

$$A = G^m (\mathrm{mod}\ P).$$

Інакше кажучи, одержувач перевіряє слушність співвідношення

$Y^a a^b (\mathrm{mod}\ P) = G^m (\mathrm{mod}\ P)$ Слід зазначити, що обрання кожного підпису за допомогою методу Ель Гамаля потребує нового значення K, причому це значення має обиратися випадково. Якщо злочинник розкриє коли-небудь значення ключа K, вдруге використовуваного відправником, то він зможе розкрити секретний ключ X відправника.

4 АНАЛІЗ ХАРАКТЕРИСТИК АЛГОРИТМІВ КРИПТОГРАФІЧНОГО ЗАХИСТУ ІНФОРМАЦІЇ

4.2 Порівняльний аналіз блокових алгоритмів шифрування

Основними перевагами шифру Калина, порівняно з іншими міжнародними аналогами, є можливість застосовувати блок даних і ключ шифрування розміром аж до 512 біт, збільшена кількість циклів шифрування та принципово нова схема створення підключів, що забезпечують захист від усіх відомих атак на схеми їх створення. Тому введення в дію нових національних стандартів ДСТУ 7624:2014 і ДСТУ 7564:2014 дозволить суттєво удосконалити показники ефективності систем захисту, засобів і протоколів криптографічного захисту інформації, які розробляються в Україні, а в деяких випадках поліпшити їх порівняно з існуючими та перспективними світовими практиками.

Основні відмінності "Калина" від "Rijndael"(AES):

- ступені нелінійності булевих функцій (на відміну від S-блоку Rijndael/Camellia та інших шифрів, що використовують звернення в полі та, відповідно, квадратичні залежності між входом і виходом, – захист від алгебричних атак);

- кількість етапів шифрування збільшена;

- операції додавання за модулем 2^{64} і за модулем 2 для введення ключової інформації;

- застосування 4 блоків S заміни замість одного;

- випадково сформовані 4-блоки;

- нова схема створення підключів (захист від всіх відомих атак на схеми створення підключів);

У наступному графіку, який зображено на рис. 4.1 різницю у швидкодії (Мб/с) двох алгоритмів враховуючи різні довжини ключів. Дослідження проводились на операційній системі Linux 64 bit (компілятор gccversion 4.9.2), процесор IntelCorei5-4670 CPU @ 3.40GHz.

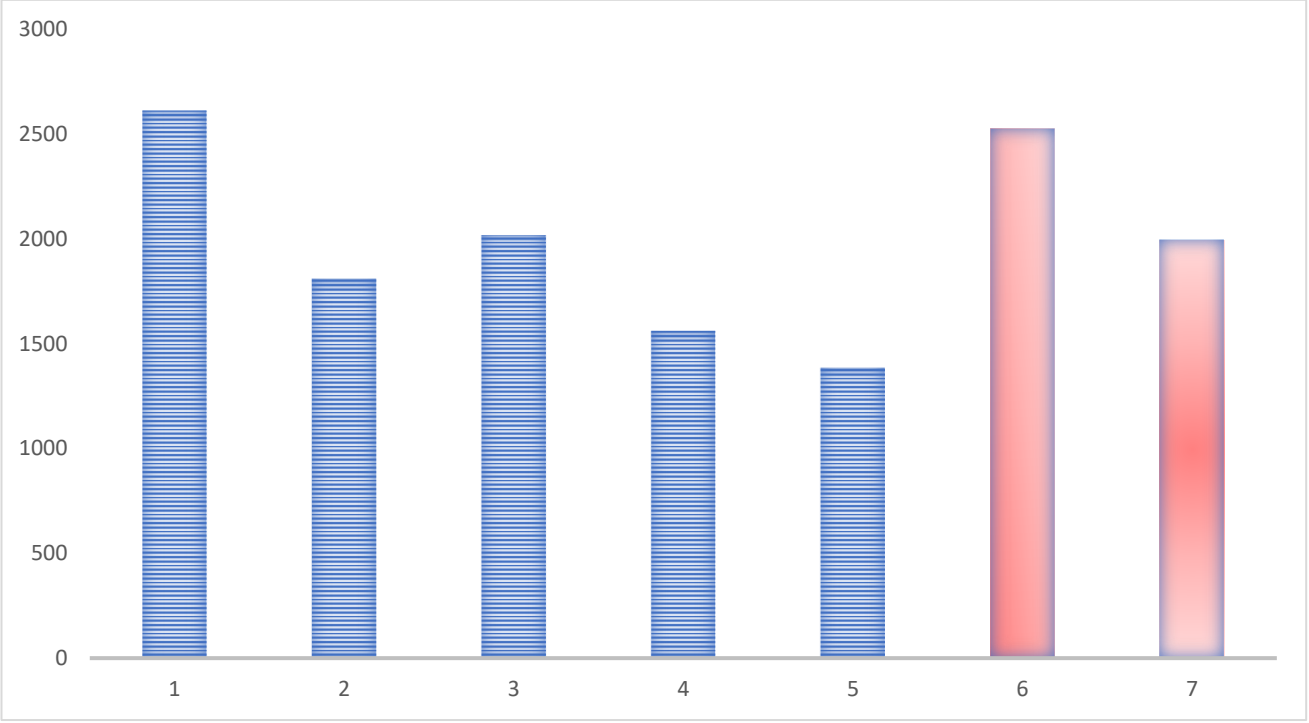


Рисунок 4.1 – Порівняння швидкодії алгоритмів блочного шифрування

1 — Калина(К) 128/128 — 2611.77 Мб/с;
2 — К 128/256 —1809.70 Мб/с;
3 — К 256/256 — 2017.97 Мб/с;
4 — К 256/512 — 1560.89 Мб/с;
5 — К 512/512 — 1386.46 Мб/с;
6 — AES 128 – 2525.89 Мб/с;
7 — AES 256 – 1993.53 Мб/с.

Дане дослідження підтверджує, що алгоритм Калина має найбільшу швидкодію при використанні блока даних 128 біт і ключа 128 біт, але поступається алгоритму AES при розмірі блоку 128 біт і ключа 256 біт. При довжині блоку та ключа 256 біт швидкість алгоритму Калина більше за швидкість алгоритму AES.

Для того, щоб прийняти рішення щодо правильного вибору алгоритму для шифрування конфіденційних даних проведемо додаткове порівняння алгоритмів AES та Camelllia (шифр-фіналіст на конкурсі AES).

З технічної точки зору, Camellia має як високий рівень безпеки, так і добру ефективність та практичні характеристики. Цей алгоритм може бути реалізованим на різноманітних платформах з високою продуктивністю за допомогою програмного забезпечення. Що стосується апаратної реалізації, то підтверджується низьке енергоспоживання. Виходячи з цих технологічних переваг, Camellia отримала міжнародне признання. Проект відбору за європейською рекомендацією сильних криптографічних примітивів NESSIE описав даний алгоритм так: "Camellia дуже подібний до AES, тому більша частина аналізу AES також застосовується до Camellia ". На сьогоднішній день Camellia - єдиний міжнародно визнаний шифр, який має такий самий рівень безпеки та продуктивності, як AES, і відібраний для багатьох міжнародних стандартних і/або рекомендованих шифрів. Зокрема, це перший випадок, коли японський вітчизняний алгоритм шифрування був затверджений як стандартний шифр IETF (міжнародне співтовариство, яке займається розвитком протоколів і архітектури Інтернету).
NTT також забезпечує Camellia з відкритим кодом, що дає можливість перегляду, вивчення та змін. Зараз Camellia завантажена багатьма основними

міжнародними програмами з відкритим кодом, наприклад, OpenSSL, Firefox, Linux та FreeBSD.

NTT та Mitsubishi Electric Corporation надають ліцензії на безкоштовне надходження основних патентів на Camellia з метою встановлення провідної ролі у напрямку досягнення некоштовного безпечного розвиненого телекомунікаційного суспільства шляхом розповсюдження та просування Camellia, що сприяє побудові середовища, в якому різні продукти та послуги безпеки можуть широко використовуватися.

AES та Camellia є блочними шифрами з розміром блоку 128 біт і трьома різними розмірами ключів: 128, 192 і 256 біт.

В обох випадках, коли використовується більший розмір ключа, наприклад 192 або 256, для шифрування будуть використовуватися додаткові раунди. AES виконує 10, 12 і 14 раундів для 128, 192 і 256 бітових ключів, тоді як Camellia виконує 18 раундів для 128-бітових ключів і 24 раундів для 192 і 256-бітних ключів. Тоді ми розглянемо різницю часу шифрування від розміру файлу.

З цією метою було використано 5 файлів для обох алгоритмів розміром від 10 Кб до 50 Мб, час шифрування яких зображено у табл.4.1. Для кожної ситуації були запущені програми, і була використана довжина ключа 256 біт. Для дослідження була використана відкрита бібліотека SSL, включена в Ubuntu Linux 14.04 TLS. Операційна система Linux Ubuntu 14.04 TLS вимагає мінімального обслуговування. Використаними обчислювальними системами були два ноутбуки Asus, обидва засновані на мікропроцесорах Intel, Core i5 і Core процесори i7 [5].

Таблиця 4.1 – Час шифрування файлів за алгоритмами AES та Camellia

Розмір файлу	Час шифрування AES, сек	Час шифрування Camellia, сек
10KB	0,005	0,005
20KB	0,005	0,0052
30KB	0,005	0,0052
40KB	0,005	0,0056
50KB	0,0052	0,0056

100KB	0,006	0,0068
200KB	0,0064	0,008
300KB	0,0074	0,009
400KB	0,0076	0,0104
500KB	0,008	0,0122
1 MB	0,0114	0,019
2 MB	0,0168	0,0338
3 MB	0,0226	0,047
4 MB	0,0276	0,0602
5 MB	0,032	0,0742
10MB	0,0566	0,1506
20MB	0,1102	0,294
30MB	0,161	0,437
40MB	0,2136	0,5828
50MB	0,2658	0,7258

Наступний графік, зображений на рис.4.2, демонструє порівняння алгоритмів шифрування цілого набору файлів.

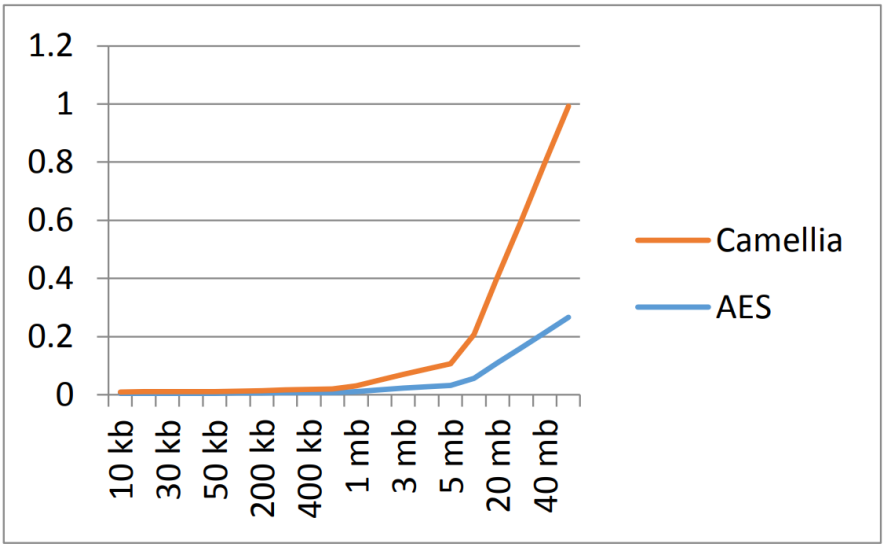


Рисунок 4.2 - Порівняння алгоритмів шифрування

Враховуючи середні показники швидкостей алгоритмів Калина, Camellia та AES при розмірі ключа 256 біт, і розмірі блока даних 128 біт ми отримуємо результати, зображені на рис.4.3

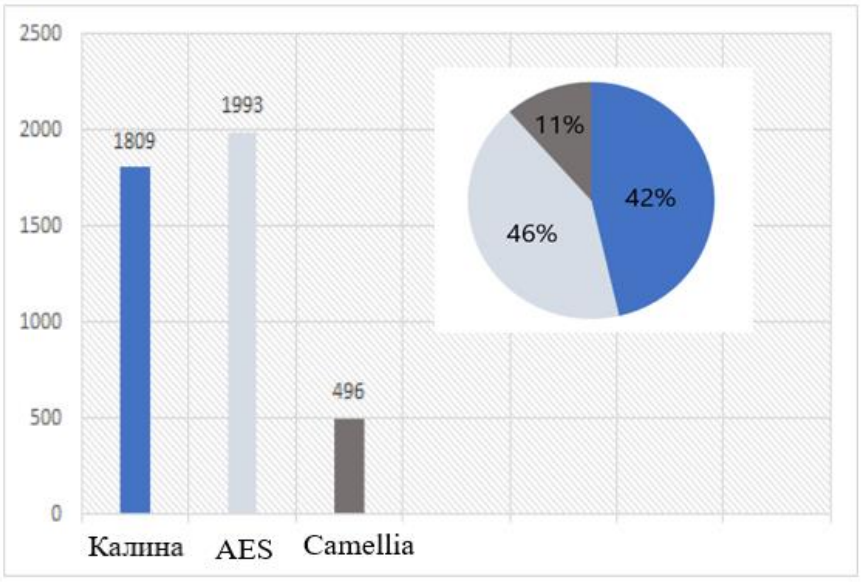


Рисунок 4.3 – Результат порівняння алгоритмів Калина, AES та Camellia

Особливе значення має порівняння основних двох шифрувальних алгоритмів для того, щоб точно знати, який алгоритм є більш ефективним залежно від розміру файлу**.**

Алгоритм шифрування AES забезпечує дуже високий рівень безпеки та дуже швидке впровадження програмного та апаратного забезпечення.

Під час порівняльного аналізу алгоритмів AES та Калини, було вирішено, що Калина, по-перше, дозволяє обирати рівень криптостійкості за рахунок можливості збільшення довжини ключа. По-друге, він є швидким та продуктивним алгоритмом шифрування при використанні ключа, довжиною 128 біт. Але так як Калина не входить в перелік запроронованих алгоритмів шифрування криптоконтейнерів і в порівнянні з AES має нижчий показник швидкодії при розмірі ключа 256 біт, можна зробити висновки, що алгоритм AES, надає кращі показники швидкості шифрування, ніж алгоритм Camellia.

4.3 Порівняльний аналіз алгоритмів електронних підписів

Стійкість схеми електронного підпису залежить від стійкості використовуваних криптоалгоритмів та геш-функцій.

Класифікація атак на схеми електронного підпису:

– *атака на основі відомого відкритого ключа*– майже завжди доступна

для зловмисника;

– *атака на основі відомих підписаних повідомлень*– у розпорядженні зловмисника є число пар (M, S), де M – певне повідомлення, а S – припустимий підпис для нього;

– *спрямована атака з вибором повідомлень*– обираючи підписані повідомлення, зловмисник знає відкритий ключ;

– *адаптивна атака з вибором повідомлень* – зловмисник знає відкритий ключ

Найбільш надійними є схеми, стійкі проти найслабкішої із загроз на базі найпотужнішої з атак, тобто проти екзистенційної підробки на базі атаки з вибором підписаних повідомлень.

4.3 Вибір оптимальних параметрів протоколу захисту електронних транзакцій TLS

Важливою складовою інформаційної безпеки в Інтернеті є захист даних за допомогою протоколу захисту транспортного рівня TLS (Transport Layer Security). Цей протокол забезпечує криптографічний захист інформації між вузлами комп'ютерної мережі.

Протокол TLS базується на основі протоколу SSL (Secure Socket Layer або Рівень Захищених Сокетів).

Спочатку в 1995 був опублікований протокол SSL (одразу з версії 2.0, тому що версія 1.0 мала серйозні недоліки з безпеки).

У 1996 році вийшов SSL 3.0 — повністю перероблена версія протоколу.

У 1999 році для версії SSL 3.0 Інженерною радою Інтернету (IETF) було запропоновано оновлення, яке почало носити назву протоколу TLS 1.0 .

У 2006 році було створено оновлену версію протоколу TLS 1.1.

У 2008 році було створено покращену версію TLS 1.2.

У 2018 році – остання оновлена покращена версія протоколу TLS 1.3.

В даний час чинними є лише версії TLS 1.2 и 1.3.

Протоколи SSL версій 2.0 та 3.0 були визнаними застарілими у 2011 та 2015 роках відповідно.

Протоколи TLS версій 1.0 та 1.1 були визнаними застарілими у 2020 році.

Процес розвинення протоколу TLS наведено на Рис. 4.4.

TLS протокол використовується для наступних завдань:

- здійснює взаємну автентифікацію за допомого асиметричної криптографії;
- забезпечує конфіденційність даних за допомогою симетричної криптографії;
- гарантує цілісність повідомлення шляхом алгоритмів автентифікації.

TLS - протоколи використовуються в мережі для різного призначення. Про це можна дізнатися, коли наприкінці відповідного позначення протоколу

є літера S. Так, наприклад, веб-сайти, які мають позначку HTTPS (HTTP, HyperText Transfer Protokol, Протокол передачі гіпертексту) захищені саме протоколом TLS. Для безпечного передавання файлів використовується FTPS (FTP, File eXchange Protokol, Протокол обміну файлами) протокол. Захист протоколу електронної пошти SMTPS (SMTP, Simple Mail Transfer Protokol, Простий протокол пересилання пошти) також здійснюється протоколом TLS.

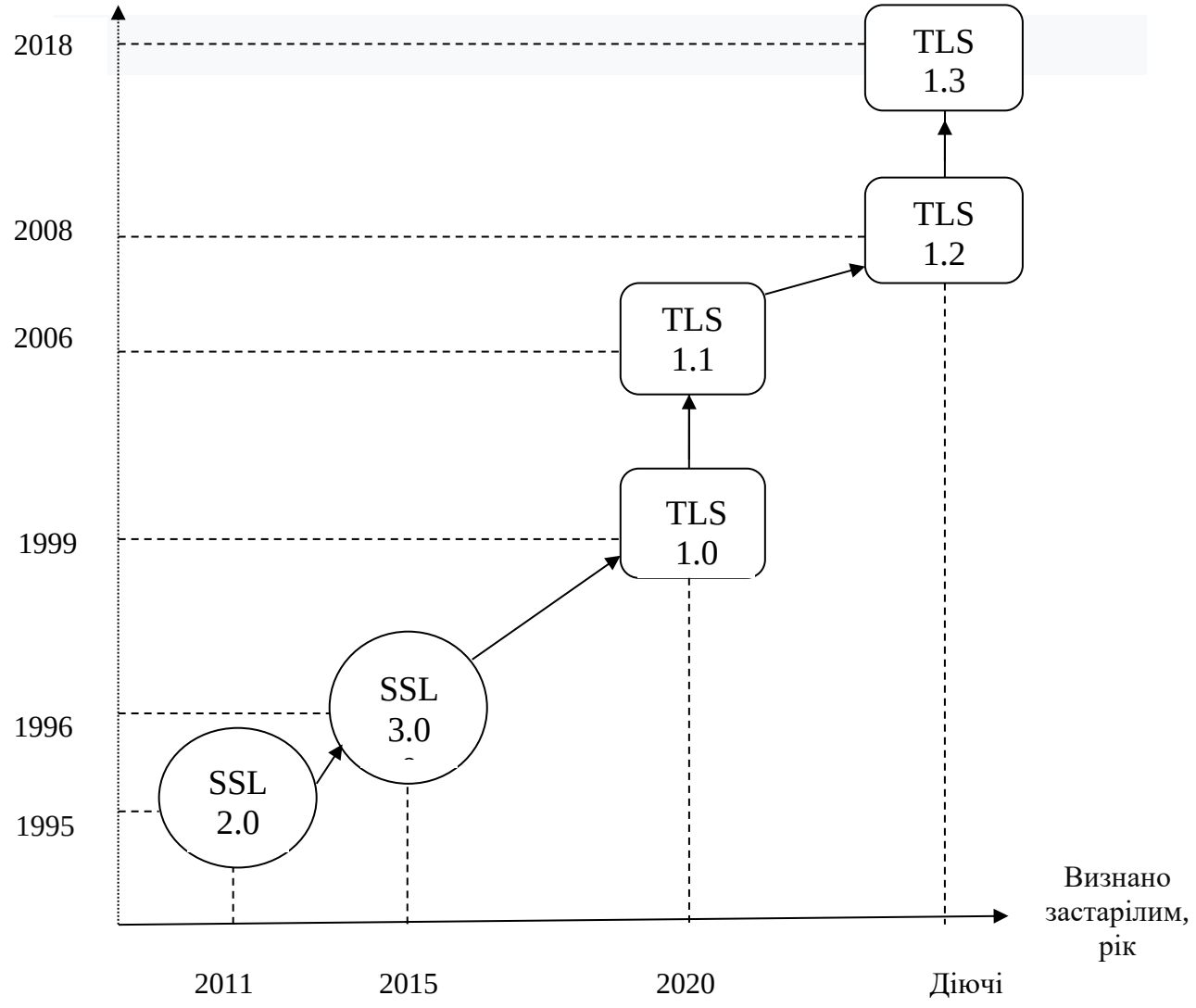


Рисунок 4.4 - Процес розвинення протоколу TLS

TLS – протокол складається з двох етапів:

1 Протокол рукостискання. Метою цього етапу є взаємна автентифікація користувача і сервера та обмін ключами.

На початку сеансу зв'язку, сторони мають домовитися про:

- версію протоколу;
- криптографічний алгоритм шифрування;
- провести взаємну автентифікацію;
- обрати спільний секретний ключ для подальшого шифрування.

2 Протокол запису. Метою цього етапу є забезпечення цілісності та конфіденційності інформації.

На цьому етапі:

- всі вихідні повідомлення зашифровуються за допомогою симетричного алгоритму та спільного секретного ключа, що були обрані на першому етапі;
- зашифровані повідомлення передаються на протилежний бік;
- на приймальному боці криптограми перевіряються на цілісність;
- в разі підтвердженні цілісності криптограми, її розшифровують за допомогою того самого симетричного алгоритму і спільного секретного ключа.

Для обрання оптимальних параметрів протоколу, по-перше необхідно розібратися з діючими версіями протоколу TLS.

Версія протоколу TLS 1.3. працює з більш надійними криптографічними алгоритмами. Так , для розподілу ключів видалено алгоритм RSA та залишився тільки ефемерний алгоритм Діффі-Хеллмана та алгоритм Діффі-Хеллмана на еліптичних кривих (EC)DH. Щодо алгоритмів симетричного шифрування, то в останній версії протоколу TLS використовується шифрування AEAD

У таблиці 4.2 показано порівняльний аналіз характеристик протоколу версій TLS 1.2 та TLS 1.3.

Таблиця 4.2 - Порівняльний аналіз характеристик протоколу версій TLS 1.2 та TLS 1.3.

№	**TLS 1.2**	**TLS 1.3.**	**Параметри протоколу**
1	RSA; (EC)DH; (EC)DHE	(EC)DHE	Механізм обміну ключами
2	2-RTT	1-RTT; 0-RTT	Прискорене рукостискання
3	AEAD, CBC, RC4, 3DES	AEAD	Алгоритм симетр. шифрування
4	DHE; RSA; AES256; SHA256	AES256; SHA384	Набір шифрів
5	Підпис частини рукостискання	Підпис всього рукостискання	Електронний підпис
6	ECDSE (P-256; P-384)	EdDSA (Ed25519;Ed448)	Алгоритм ЕП на еліптичних кривих

У таблиці 4.3 показано алгоритм повного рукостискання протоколу TLS 1.3.

Таблиця 4.3 - Протокол повного рукостискання TLS 1.3

Клієнт (пропонує)	Версія протоколу, що пропонується	TLS 1.3; TLS 1.2
	Набори шифрів	TLS_ AES-256_GCM_ SHA384; TLS_CHACHA20 _POLY1305_SHA256
	Протокол обміну ключами	В кінцевому полі (DHE) На еліптичних кривих (ECDHE)
	Ключі	DHE відкритий ключ клієнта ECDHE відкритий ключ клієнта
	Протоколи електронного підпису	RSA RSS ECDSA
Сервер (обирає)	Вибрана версія протоколу	TLS 1.3;
	Набор шифрів	TLS_ AES-256_GCM_ SHA384;
	Протокол обміну ключами	В кінцевому полі (DHE)
	Ключ, що передається	DHE відкритий ключ сервера
	Запит сертифіката	Запит сертифіката клієнта
	Сертифікат сервера	Сертифікат сервера
	Поле для перевірки справжності сертифіката	Підпис всього сертифіката
	Server Finish	MAK всього рукостискання
Клієнт	Сертифікат клієнта	Сертифікат клієнта
	Поле для перевірки справжності сертифіката	Підпис всього сертифіката
	Client Finish	MAK всього рукостискання

Висновки

У магістерській роботі одержані такі результати:

- досліджено сучасні криптографічні методи захисту інформації, що використовуються в системах телекомунікації;
- проаналізовано криптоалгоритми за призначенням;
- досліджено методи забезпечення конфіденційності та цілісності електронних документів при передачі інформації;
- проведено порівняння характеристик симетричних криптосистем за основними параметрами та зроблено висновки щодо обрання кращого криптографічного алгоритму для поставлених завдань;
- досліджено методи ідентифікації та підтвердження справжності клієнта та даних ;
- обрано оптимальні параметри протоколу захисту TLS.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1 Про затвердження Положення про порядок розроблення, виробництвата експлуатації засобів криптографічного захисту інформації.

[Електронний ресурс]. – Режим доступу: https://zakon.rada.gov.ua/laws/show/z0862-07#Text

5 Daemen J., Rijmen V. Design of Rijndael. AES – The Advanced Encryption Standard. – Berlin: Springer–Verlag. – 2002. – 238 p.

6 Camelliaand SEED [Електронний ресурс]. – 1212. – Режим доступу до ресурсу: https://www.jmest.org/wp-content/uploads/JMESTN42351298.pdf.

7 ДСТУ 7624:2014. Інформаційні технології. Криптографічний захист інформації.

Алгоритм симетричного блокового перетворення. [Алгоритм шифрування Калина]. Мінекономрозвитку України, 2016. 228с. https://www.kmu.gov.ua/news/247952015.

5 Криптографічний захист інформації: навч.посібн. з дисципліни «Криптографічний захист інформації» / О.В.Онацький, Л.Г.Йона, Ю.В.Бєлова; Держ.ун-т інтелект.технологій і зв'язку.- Одеса: Астропринт, 2023.-252с.