Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет *Інфокомунікацій*
(повна назва)
Кафедра *Інформаційно-мережної інженерії*
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження криптокомпресійних методів обробки інтерактивного відео на базі поліадичного коду з нефіксованими вагами
(тема)

Виконав:
студент _2_ курсу, групи *ІМІм-20-1*
Ольховський В.І.
(прізвище, ініціали)

Спеціальність *172 Телекомунікації та радіотехніка*
(код і повна назва спеціальності)
Тип програми *освітньо-професійна*
(освітньо-професійна або освітньо-наукова)
Освітня програма *Інформаційно-мережна інженерія*
(повна назва освітньої програми)

Керівник *ст.викл. Твердохліб В.В.*
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ *Безрук В.М.*
(підпис) (прізвище, ініціали)

2021 р.

Не містить відомостей, заборонених
до відкритого публікування

Керівник _____ / *В.В. Твердохліб*.

Студент _____ / *В.І. Ольховський*

Харківський національний університет радіоелектроніки

Факультет *Інфокомунікацій*

Кафедра *Інформаційно-мережної інженерії*

Рівень вищої освіти другий (магістерський)

Спеціальність *172 Телекомунікації та радіотехніка*

(код і повна назва)

Тип програми *Освітньо-професійна*

(освітньо-професійна або освітньо-наукова)

Освітня програма *Інформаційно-мережна інженерія*

(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«_8_» *листопада* 2021 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові *Ольховському Володимиру Івановичу*

(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження криптокомпресійних методів обробки інтерактивного відео на базі поліадичного коду з нефіксованими вагами*

затверджена наказом університету від _8_ листопада 2021 р. № 1674 Ст

2. Термін подання студентом роботи до екзаменаційної комісії _17_ грудня 2021 р.

3. Вихідні дані до роботи *Довести, що саме процес шифрування інтерактивного відео за умови його повного закриття потенційно здатен внести найбільшу затримку пакетів; розглянути шляхи зменшення такої затримки. Показати доцільність модифікації існуючої схеми кодоутворення застосуванням поліадичного коду на етапі стиснення без втрат. Дослідити принципи функціонування ряду алгоритмів шифрування та обґрунтувати який з них є найбільш доцільним для шифрування інтерактивного відео. Виконати дослідження процесу формування шифрограм відео, попередньо кодованого з використанням поліадичного коду. Обґрунтувати переваги такого підходу.*

4. Перелік питань, що потрібно опрацювати в роботі *Вступ*

1. Вимоги до процесу обробки відеоінформації

2. Вибір та обґрунтування підходу, що дозволяє мінімізувати кількість обчислювальних операцій у ході шифрування ключових кадрів

3. Дослідження поширених алгоритмів шифрування даних

4. Застосування поліадичний кодів нефіксованої ваги у базисі jpeg для зменшення обчислювального навантаження у ході формування шифрограм

Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) _____
слайди презентації в форматі Power Point (назва та мета роботи, проблематика процесу обробки відеоінформації, шляхи мінімізації кількості обчислювальних операцій у ході шифрування ключових кадрів, дослідження поширених алгоритмів шифрування, застосування поліадичний кодів нефіксованої ваги у базисі jpeg для зменшення обчислювального навантаження у ході формування шифрограм)

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Вступ	12.11.2021	Виконано
2	Розділ 1	14.11.2021	Виконано
3	Розділ 2	17.11.2021	Виконано
4	Розділ 3	19.11.2021	Виконано
5	Розділ 4	21.11.2021	Виконано
6	Висновки	24.11.2021	Виконано
7	Оформлення пояснювальної записки	26.11.2021	Виконано

Дата видачі завдання *8 листопада* 2021 р.

Студент _____
(підпис)

Керівник роботи _____ *ст.викл. Твердохліб В.В.*
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 72 с., 16 рис., 4 табл., 28 джерел, 2 додатка

AES, ВІДЕОКОДУВАННЯ, RSA, КРИПТОГРАФІЧНИЙ АЛГОРИТМ, КОДУВАННЯ З НЕФІКСОВАНИМИ ВАГАМИ, H.26*, КЛЮЧОВИЙ КАДР

Об'єкт дослідження – методи обробки відеоінформації на базі криптокомпресійного підходу для зменшення часу шифрування.

Мета роботи – дослідження методів та умов обробки інтерактивної відеоінформації, які забезпечують скорочення часу побудови шифрограм.

Здійснено аналіз умов, за яких забезпечується ефективне передавання відеоданих у реальному часі за умови застосування відносно них методів шифрування. Досліджено поширені алгоритми шифрування, які можливо застосувати для захисту відеопотоку. Виконано дослідження методу поліадичного кодування як базису для подальшого шифрування відеоінформації.

THE ABSTRACT

Explanatory note: 72 p., 16 fig., 4 tabl., 28 sources, 2 app.

AES, VIDEO CODING, RSA, CRYPTOGRAPHIC ALGORITHM, UNFIXED WEIGHT CODING, H.26 *, KEY FRAME

The object of research - methods of processing video information based on a cryptocompression approach to reduce encryption time.

The purpose of the work is to study the methods and conditions of processing interactive video information, which reduce the time of construction of ciphers.

An analysis of the conditions under which the effective transmission of video data in real time provided the application of encryption methods. Common encryption algorithms that can be used to protect the video stream are investigated. A study of the method of polyadic coding as a basis for further encryption of video information.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

SSH – (Secure Shell) – мережевий протокол безпеки прикладного рівня;
ДЗЗ – дистанційне зондування землі;
GIS – (Geoinformation System) – геоінформаційна система;
VR – (virtual reality) – віртуальна реальність;
P2PE – (Piont-to-Point Encryption) – стандарт захисту даних;
E2EE – (End-to-End Encryption) – наскрізне шифрування;
UHD – (Ultra-High Definition) – формат відео надвисокої роздільної здатності;
HD-ready – формат відеокадру, що передує роздільній здатності FullHD;
FullHD – (full-high definition) – повноформатне відео високої роздільної здатності;
VoD – (video on demand) – група сервісів, що надають послуги доставки потокового відео на запит користувача;
QoS – технологія контролю якості послуг, які надає інформаційно-комунікаційна мережа;
H.26* – група стандартів кодування відео контенту, що базується на сімействі MPEG;
H.264/AVC – (advavced video coding) — технологія високоефективного кодування відео H.264;
H.265/HEVC – (high-efficiency video coding) — технологія високоефективного кодування відео H.265;
JPEG – (Joint Photographic Expert Group) – технологія кодування статичних зображень;
ДКП – дискретне косинус не перетворення;
RGB – адитивна трьохкомпонентна колірна модель;
YCbCr – яскравісно-хроматична колірна модель;
YUV – яскравісно-хроматична колірна модель;
RLE – (Run-Length Encoding) – алгоритм кодування довжин серій;
GOP – (group of pictures) – упорядкована група кадрів різних типів, структурна одиниця відеопотоку;
MPEG – (Motion Pictures Expert Group) – сімейство стандартів кодування відеоінформації;
RSA – (Rivest, Shamir, Adleman) – асиметричний алгоритм шифрування;

AES – (Advanced Encoding System) – симетричний алгоритм блочного шифрування;

GCD – (Greatest Common Divisor) – найбільший спільний дільник;

TLS – (Transport Layer Security) – протокол захисту транспортного рівня;

ROCA – (Return of Coppersmith's Attack) – криптографічна вразливість;

XSL – (eXtended Sparse Linearization) – алгебраїчна атака, метод криптографічного аналізу.

ВСТУП

Удосконалення існуючого технологічного базису, збільшення продуктивності клієнтського та мережевого обладнання, а також зростання пропускної спроможності мережевих каналів стали тими чинниками, що зумовили розширення формату надання інформаціно-комунікаційних мережевих послуг майже в усіх сферах діяльності людини.

За таких умов, окрім суттєвого розширення існуючого функціоналу тих чи інших сервісів, виникли також принципово нові класи мережевих послуг.

У підсумку це, з одного боку, сприяло розвитку бізнесу та надало додаткові інструменти для керування ним, зумовило розширення можливостей користувачів щодо пошуку інформації, замовлення послуг та товарів.

З іншого боку, суттєво збільшилася кількість інформаційних ризиків, зумовлених високою ймовірністю крадіжки, руйнування або заміни інформації [1]. Це пояснюється як тим, що в умовах загального технологічного розвитку суттєво збільшилися можливості зловмисників, так і тим, що за останні роки значно зріс обсяг інформації обмеженого доступу, яку необхідно надсилати мережевими каналами, або зберігати на фізичних серверах або хмарних сховищах.

Для того, щоб убезпечити найбільш критичні дані, сьогодні використовуються:

- протоколи безпеки обміну даними;
- захищені канали;
- маскування даних на базі стегоалгоритмів;
- криптографічний захист.

При цьому, перелічені засоби сьогодні реалізовано або як самостійні інструменти, або як вбудовані модулі у складі тих чи інших програмних засобів – месенджерів, систем електронного документообігу та платежів, тощо.

У загальному випадку, якщо говорити про безпечне передавання різнорідної інформації мережевими каналами, сьогодні найбільш широко використовується створення безпечних каналів типу «point-to-point» (наприклад, на базі SSH), або надсилання відкритими каналами попередньо

шифрованих чи маскованих даних, а також практикується спільне використання обох способів.

Разом з тим, в умовах збільшення відсотку відеоінформації у мережі, постає питання захисту також і цього типу даних.

При цьому, найбільш гострим це питання є для випадків криптозахисту відео надвисоких роздільних здатностей та відео інтерактивного типу. Це пояснюється необхідністю забезпечення побудови шифрограм у реальному часі, що накладає додаткові вимоги як на апаратну частину клієнтських терміналів, так і на безпосередньо алгоритми шифрування.

Тому у зазначених умовах дослідження, спрямовані на збільшення ефективності шифрування критичних відеоданих у реальному часі є актуальними.

1. ВИМОГИ ДО ПРОЦЕСУ ОБРОБКИ ВІДЕОІНФОРМАЦІЇ

1.1 Сфери застосування відеоінформації

Відповідно до щорічного звіту [2], який формується за результатами досліджень компанії Cisco, за останні роки відео перетворилося на домінантний тип трафіку. Так, зараз відео відповідає найбільша відсоткова частка серед усіх існуючих типів трафіку. При цьому, у загальній масі даних, що передаються мережею щосекунди, у загальносвітовому масштабі відео займає понад 80%. Надалі спостерігається виражена тенденція щодо постійного збільшення відсоткової частки відеотрафіку.

За даних умов відео виконує роль як ключового трафіку тих чи інших сервісів, або додаткового, що дублює та/або доповнює трафік інших типів, таких, як трафік даних, аудіо та ін. Інакше кажучи, перелік сфер застосування відео та обсяг завдань, рішення яких передбачає пряме або опосередковане застосування відео контенту, також розширюється [3]. Ряд сфер застосування відео приведено рис.1.1. При цьому, слід зазначити наступне:

- наведений перелік сфер застосування відео є далеко не вичерпним та постійно збільшується;
- ряд галузей (наприклад, ДЗЗ, Smart City, комп’ютерний зір та ін.) мають велику кількість за стосунків відео;
- умовно відеоконтент може бути розподілено на відкритий та закритий.

Так, відео відкритого типу являє собою дані, що однаково та без обмежень доступні широкій аудиторії на постійній основі. При цьому, доступ до таких даних може бути вільним незалежно від типу відео контенту, або надаватися за умови внесення користувачем відповідної платні.

Разом з тим, сегмент відеоданих закритого типу, який на сьогодні є досить суттєвим, утворює відео контент, доступ до якого має обмежене коло осіб. Це, зазвичай, такі дані, як [4, 5]:

- інформація, пов’язана з питаннями національної та громадської безпеки;
- стратегічно важлива інформація, та інформація, що являє собою об’єкт комерційної таємниці;
- конфіденційні відеодані.

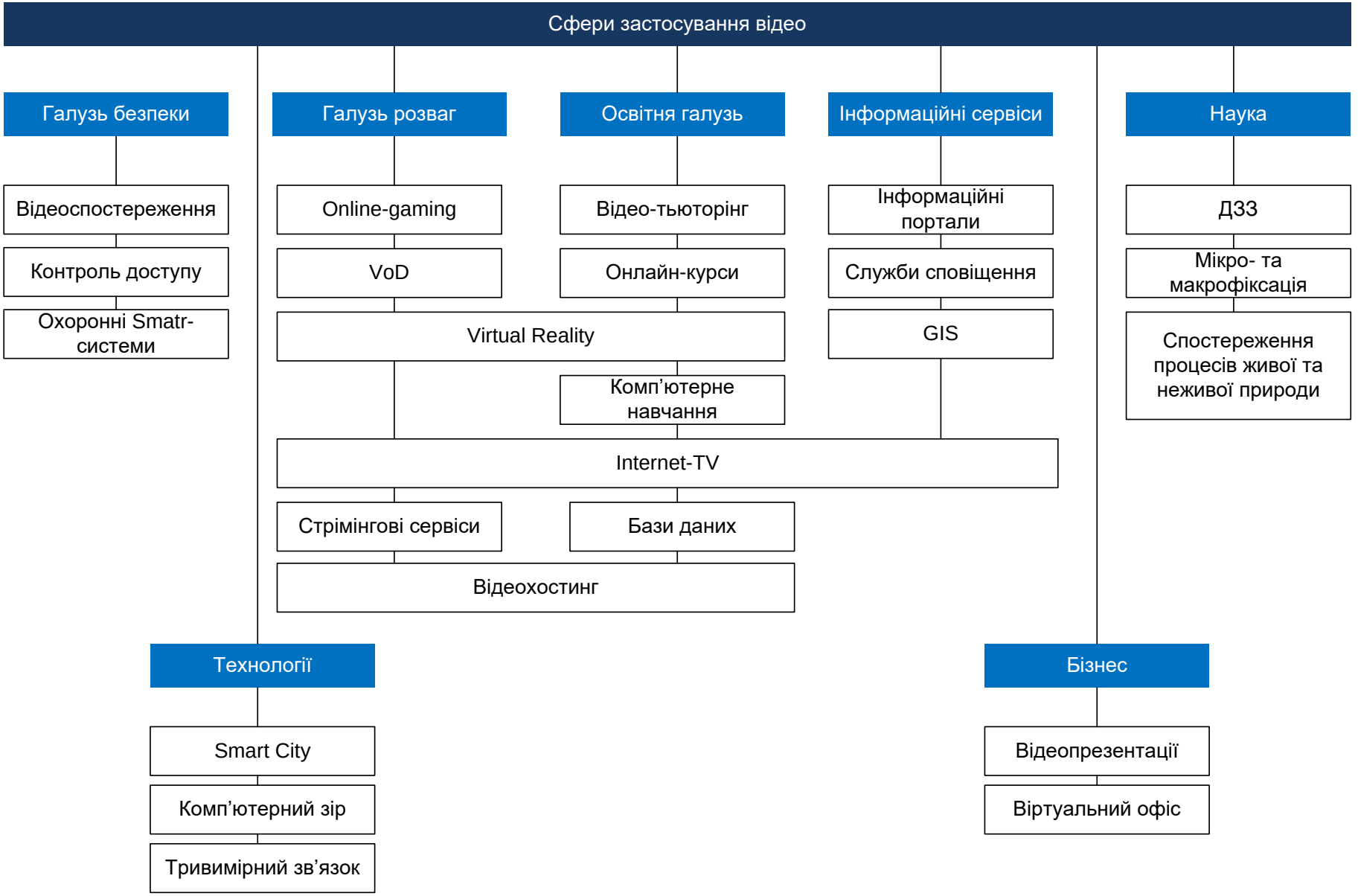

Рисунок 1.1 – Сфери застосування відео

1.2 Підходи до забезпечення захисту відеоінформації з обмеженим доступом

Для того, щоб створити умови безпечного передавання відео закритого типу, та мінімізувати ймовірність його перехоплення зловмисником з подальшою модифікацією чи видаленням, на сьогодні використовуються такі ключові підходи, як (рис.1.2) [4]:

- використання окремих фізичних каналів для надсилання даних закритого типу;
- шифрування даних з обмеженим доступом на базі криптографічних алгоритмів.

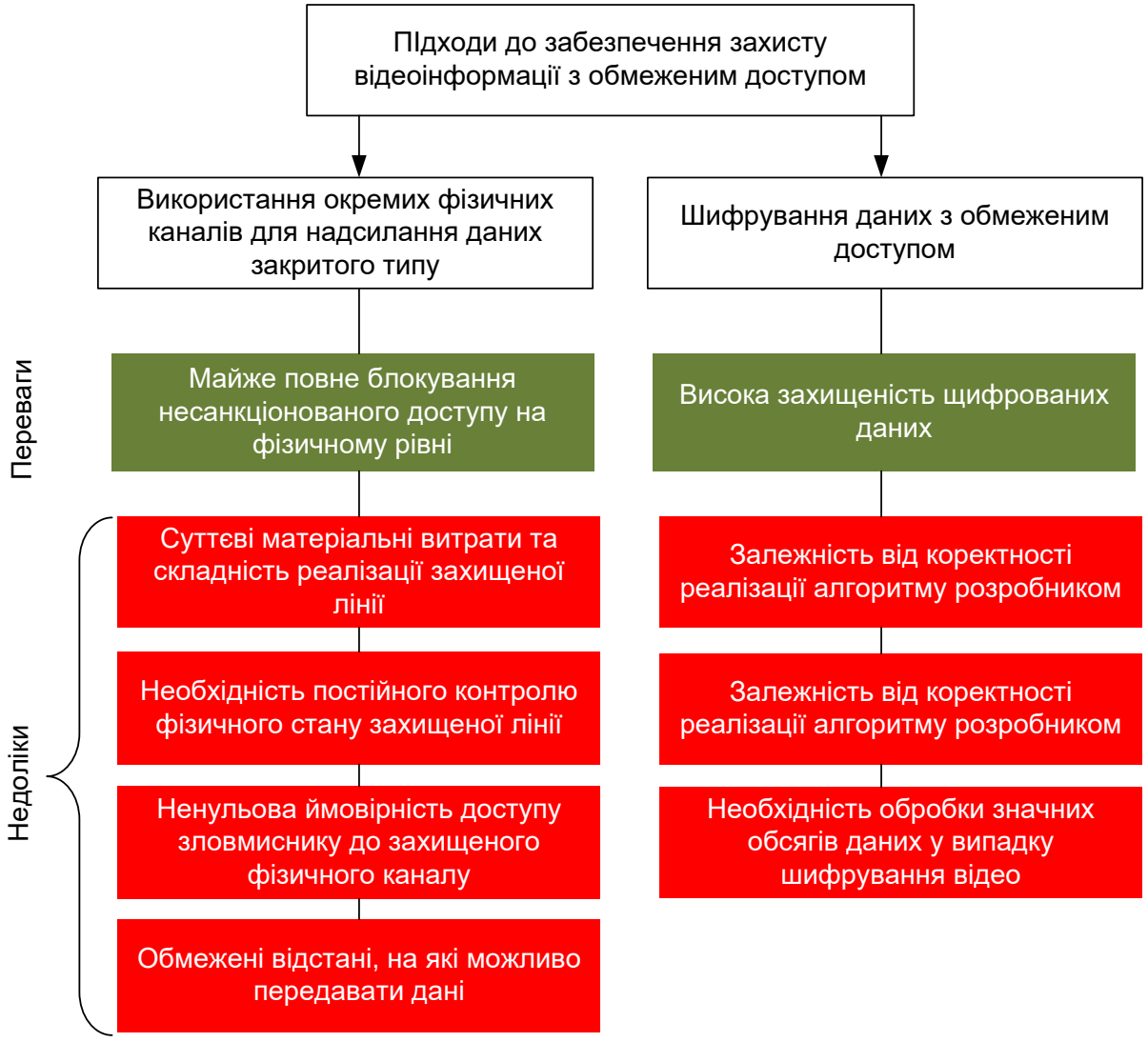

Рисунок 1.2 – Переваги та недоліки підходів до забезпечення захисту відеоінформації з обмеженим доступом

Перший підхід передбачає повну фізичну ізоляцію мережевої інфраструктури від будь-яких користувачів окрім тих, що мають відповідні права доступу до неї. Це може бути:

- спеціалізована intranet-мережа, обмежена територією об’єкту з особливим режимом доступу;

- захищений фізичний канал між окремими сегментами мережі, без використання спільного розподіленого середовища.

У випадку реалізації захищеного зв’язку в означений спосіб теоретично повністю блокується можливість несанкціонованого доступу зловмисника до даних, у т.ч. і відео.

Недоліками даного підходу є:

- суттєві матеріальні витрати та складність реалізації захищеної лінії у випадку її значних фізичних габаритів;
- необхідність постійного контролю фізичного стану захищеної лінії у реальному часі;
- ненульова ймовірність доступу зловмиснику до захищеного фізичного каналу;
- обмежені відстані, на які можливо передавати дані за такого способу реалізації захищеного каналу.

У свою чергу, підхід, який передбачає використання засобів криптографічного захисту даних, з одного боку, не гарантує повної відсутності ймовірності відкриття даних.

З іншого боку, кожен криптографічний алгоритм, що використовується для шифрування, має певний рівень стійкості, що може бути інтерпретовано як час, необхідний зловмиснику для дешифрування інформації [6]. Так, злам ряду сучасних криптографічних алгоритмів може зайняти у зловмисника щонайменше кілька днів, та навіть значно більше часу. При цьому, протягом даного часу інформація знеціниться і її дешифрування не матиме сенсу.

Проте, даному підходу також властивий ряд недоліків, а саме:

- реальна криптографічна стійкість суттєвим чином залежить від коректності реалізації алгоритму розробником;
- шифрування відео у загальному випадку передбачає обробку значних обсягів даних.

У загальному випадку за сукупністю переваг та недоліків більш доцільним для забезпечення захищеності даних вважається підхід на базі застосування алгоритмів криптографії.

Разом з тим, відео, як один з типів трафіку, має ряд характерних відмінностей.

1.3 Відмінності відео потокового та інтерактивного типів з позиції обробки на базі алгоритмів шифрування

По-перше, як раніше зазначалося, на відео сьогодні приходиться більш, ніж 80% усього трафіку, а сегмент відеоданих з обмеженим доступом постійно зростає.

При цьому, відеодані умовно можуть бути поділені на [7]:

- потокові;
- інтерактивні.

Перший тип відеоданих являє собою інформацію, яку попередньо було сформовано та кодовано, а також оброблено з використанням технологій шифрування. У загальному випадку потокове відео зберігається у файлосховищах, а доступ до нього надається за вимогою користувача.

Отже, оскільки потокове відео не генерується у реальному часі, як це справедливо для відео інтерактивного типу, та вже зразу зберігається у шифрованому вигляді, відповідно, процес його трансляції нічим не відрізняється від випадку трансляції відеоінформації відкритого типу.

У свою чергу, інтерактивне відео формується джерелом у реальному часі, та у реальному часі також має доставлятися отримувачеві. З цієї точки зору, умови обробки відео цього типу є суттєво жорсткішими, ніж потокового, що зумовлено наступним [8]:

- кодування відеопотоку повинно виконуватися у реальному часі;
- процедура шифрування відеоряду не повинна вносити суттєвої додаткової затримки обробки;
- загальний допустимий час $t_{st}(int)$ затримки пакетів відео інтерактивного типу має бути майже у 33 рази нижчим, ніж відповідний показник $t_{st}(str)$ для потокового відео (4-5 секунд проти 150 мсек).

Так, загальна затримка t доставки пакетів відео визначається на базі виразу:

$$t = t_{gen} + t_{enc} + t_{sh} + t_{ch} + t_{nt} \le t_{st}, \qquad (1.1)$$

де t_{gen} - час безпосередньо формування відеоінформації джерелом;

t_{enc} - час, який витрачається на кодування сформованих відеоданих;

t_{sh} - час шифрування відео з застосуванням криптографічних алгоритмів;

t_{ch} - час загальної обробки, куди входить пакетування кодованих даних, переміщення їх до вихідного буферу тощо;

t_{nt} - час, який займає передавання відео мережею;

t_{st} - стандартизована величина затримки, що є допустимою за QoS.

Необхідно зазначити, що для випадку потокового відео, яке є попередньо обробленим, справедливо наступне [7, 9]:

$$t_{gen} = t_{enc} = t_{sh} = 0. \tag{1.2}$$

Отже, для відео потокового типу загальний вираз для опису сумарної затримки наближено буде таким, як:

$$t_{st}(str) \approx t_{ch} + t_{nt} \tag{1.3}$$

При цьому, також пам’ятаємо, що за існуючими вимогами до затримок також справедливою є нерівність:

$$t_{st}(int) < t_{st}(str) \tag{1.4}$$

Звідси виходить, що за даних умов, щоб забезпечити процес обробки інтерактивної відеоінформації на рівні джерела з використанням шифрування відповідно до вимог QoS, має виконуватися умова мінімізації часу кодування та шифрування, а саме:

$$t_{enc},\ t_{sh} \rightarrow \min. \tag{1.4}$$

На сьогодні існуючі технології кодування дозволяють виконувати кодування відео у реальному часі, та за штатних умов - без внесення

критичної додаткової затримки, тобто, умова $t_{enc} \to \min$ у переважній більшості випадків виконується.

Разом з тим, додавання до загального каскаду перетворень відеоінформації на рівні джерела технологічного етапу шифрування потенційно створює умови, коли виконання вимог QoS щодо затримки передавання пакетів не гарантується.

Таким чином, умова (1.4) наближено може бути подана у наступному вигляді:

$$t_{sh} \to \min \tag{1.5}$$

1.4 Існуюче протиріччя між рівнем захищеності відеоданих та швидкістю формування шифрограм

При цьому, у ході передавання відео формату FullHD (роздільною здатністю 1920x1080 пікселів) може створюватися потік з інформаційною інтенсивністю до 2,5-3 Мбіт/с.

Відповідно, цей самий обсяг даних протягом секунди необхідно шифрувати у відповідності до вимог умови (1.5).

Разом з тим, за існуючими статистичними даними [2] сьогодні у мережі спостерігається зростання відсотку відео роздільної здатності UHD, яке має значно вищу інформаційну інтенсивність порівняно з 2К-форматом, таким, як FullHD та HD-ready. Наприклад, відео формату 4К може утворювати потік з інформаційною інтенсивністю 35 Мбіт/с і вище.

Зрозуміло, що процес шифрування відео навіть 2К-формату на рівні клієнтського терміналу створює суттєве обчислювальне навантаження. При цьому, за таких умов для шифрування потоку 4К виконання умов (1.5) не гарантується.

Водночас, якщо систему шифрування реалізовано за принципом P2PE, тобто, point-to-point encryption, у процесі шифрування може брати участь також сервер додатку, що деяким чином зменшує навантаження на клієнтський термінал [10].

У той же час, P2PE-підходу характерними є ряд недоліків, зокрема:

- оскільки ключі шифрування зберігаються на сервері додатку, шляхом його зламу зловмисник може отримати до них доступ, за результатами чого матиме доступ безпосередньо до відео контенту;

- існує ризик крадіжки ключів співробітниками хмарного провайдеру або самої платформи, що забезпечує захищені канали трансляції відео.

Свого часу на базі P2PE функціонувала система захисту ZOOM.

Проте, з огляду на виявлені численні уразливості [11], принцип побудови шифрування було змінено.

Тому на сьогоднішній час платформа ZOOM використовує E2EE (end-to-end encrypting) архітектуру побудови шифрування [12].

У цьому випадку ключі шифрування зберігаються безпосередньо на терміналів кінцевих користувачів, відповідно, шифрограма повністю формується також на рівні кінцевих пристроїв.

За умови застосування наскрізного (E2EE) шифрування гарантується, що можливим дешифрування інформації є лише для її відправника та особи, якій дані інформацію було адресовано.

Такий підхід, з одного боку, суттєво збільшує рівень захищеності шифрованих даних, зокрема, майже повністю блокує можливість атак MITM [13], а з іншого – накладає додаткові вимоги щодо обчислювальної потужності кінцевих пристроїв, що найбільш гострим є для випадків обробки UHD-відеотрафіку.

За розглянутим матеріалом можна зазначити, що:

- для забезпечення високого ступеню захищеності даних доцільним для використання є E2EE-архітектура побудови шифрування;
- шифрування відео у реальному часі вимагає значних обчислювальних ресурсів для того, щоб затримка на побудову шифрограми не була критичною для загального часу затримки, як показано виразом (1.1), тобто, має забезпечуватися умова (1.5). Для випадку обробки трафіку надвисокої роздільної здатності забезпечення виконання означених умов не гарантується.

Отже, спостерігається протиріччя між рівнем захищеності відеоданих та швидкістю формування шифрограми. В розглянутих умовах ймовірними підходами до подолання даного протиріччя є:

- екстенсивний підхід, у рамках якого виконання умови (1.5) досягається за рахунок збільшення апаратної потужності кінцевих терміналів, або реалізація алгоритмів шифрування на базі спеціалізованих апаратно-програмних модулів;

- Інтенсивний підхід, який передбачає реалізацію шифрування з урахуванням особливостей побудови відеопотоку та принципів кодування відео.

1.5 Огляд особливостей побудови потоку кодованих відеокадрів

У рамках найбільш поширеної сьогодні платформи MPEG передбачається, що потік кодованих кадрів являє собою неоднорідну у часі структуру, для якої поточний обсяг $\eta(t)$ біт визначається функціоналом f, тобто [14]:

$$\eta(t) = f(Q(t)_f ; Q((t))_{type}), \tag{1.6}$$

де $Q(t)_f$ - особливості вмісту Q-го кадру відеопотоку, що обробляється у поточний час t;

$Q((t))_{type}$ - фактор належності Q-го кадру відеопотоку до одного з типів, прийнятих у MPEG.

Так загальний принцип MPEG-кодування відео передбачає, що потік формується на базі груп G_i кадрів. У свою чергу, кожна така група G_i утворюється поєднанням кадрів 3-х типів, а саме:

- т.з. ключових, опорних або базових кадрів – I-кадрів;
- двонаправлено передбачених кадрів (тип B, або bi-predicted);
- передбачених кадрів (P-тип, або predicted).

При цьому, групу G_i складає 1 ключовий кадр та певна кількість кадрів P та B.

Тобто, у цьому випадку кількість $N(G_i)$ біт, що містить група G_i кадрів, визначатиметься як:

$$N(G_i) = \eta(Q_I) + \sum_{k=1}^{K} \eta(Q_P^{(k)}) + \sum_{\ell=1}^{L} \eta(Q_B^{(\ell)}), \tag{1.7}$$

де $\eta(Q_I)$, $\eta(Q_P^{(k)})$ та $\eta(Q_B^{(\ell)})$ - кількість біт, що вносяться кадрами I, P та B типу відповідно;

K та L – кількість P та B-кадрів у групі.

Водночас, у групі G_i існує чіткий взаємозв'язок між кадрами, принцип якого ілюструють рисунки 1.3. та 1.4.

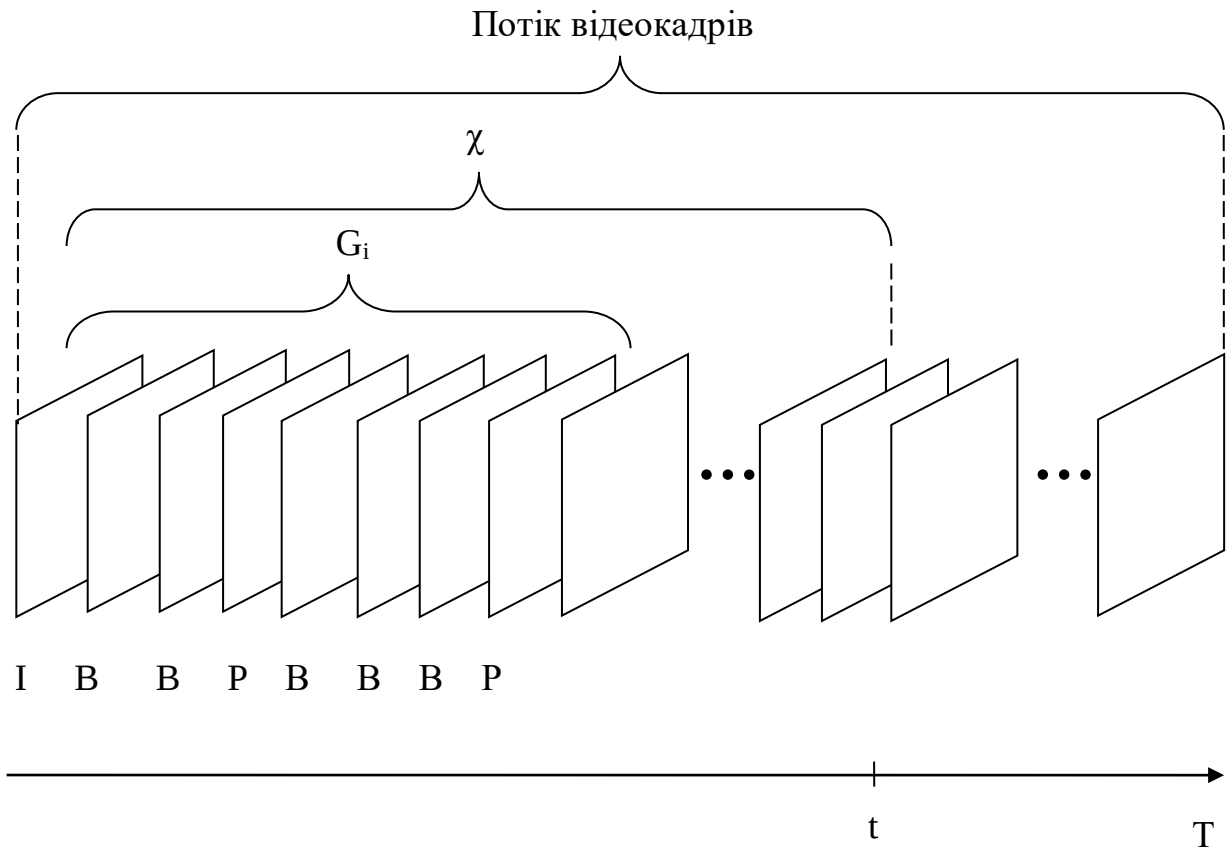

Рисунок 1.3 – Група кадрів у структурі відеопотоку

На рис. 1.3 величина χ – частота слідування кадрів у потоці. Якщо брати до уваги те, що розмірність групи G_i може варіюватися у діапазоні від 8 да 32, а стандартизоване значення χ для HD-відео – 25 або 30, та до 60 і, у окремих випадках, до 120, стає зрозумілим, що за одиницю часу t може бути надіслано або менше однієї групи G_i, або кілька груп.

При цьому, як свідчить аналіз схем на рис.1.3 та 1.4, кадри P-типу та частина B-кадрів групи залежать від I-кадру, у свою чергу, інша частина B-кадрів є залежними від P-кадрів.

Така залежність зумовлюється тим, що P та B-кадри у групі G_i, у сутності, несуть у собі різницю зміни відеосцени між I-кадрами у сусудніх групах G_i та G_{i+1}. Ця різниця відповідає складовій $\sum_{k=1}^{K}\eta(Q_P^{(k)})+\sum_{\ell=1}^{L}\eta(Q_B^{(\ell)})$ виразу (1.7).

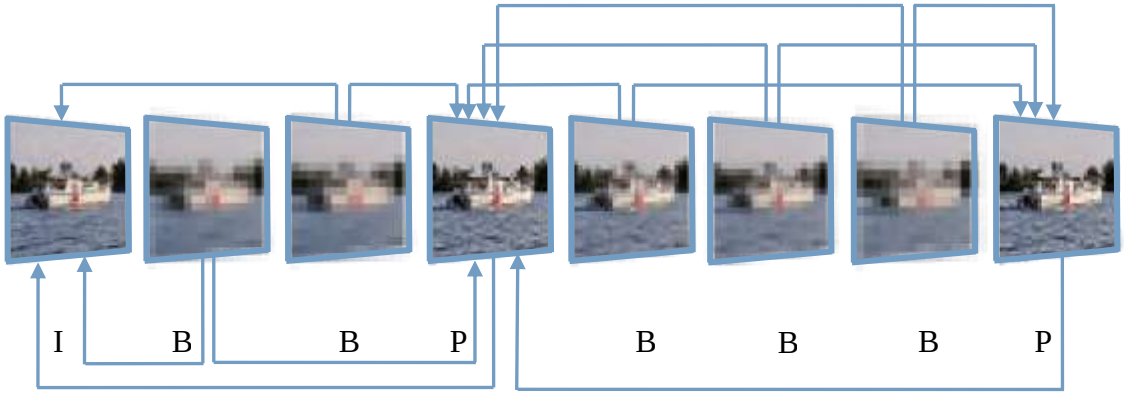

Рисунок 1.4 – Схематичне зображення типового взаємозв’язку між кадрами у групі

Тобто, як видно з аналізу виразу (1.7) та рис. 1.3 і 1.4, B та P-кадри, у сутності, є похідними об’єктами від I-кадру.

Отже, з огляду на зазначене, немає сенсу виконувати шифрування кожного кадру у групі. Для закриття відеопотоку достатньо шифрувати ключові кадри, у свою чергу, формування P та B-кадрів буде виконуватися відносно вже закритих I-кадрів. Таким чином навіть за умови, що зловмисник виконав по кадрове захоплення відеоряду, відновити його за P-кадрами, як найбільш інформативними після ключових кадрів, він не зможе.

1.6 Загальний сценарій шифрування відеоконтенту з урахуванням особливостей формування відеопотоку у MPEG-базисі

Припустимо, що існує джерело відеоінформації. Дані, які генерує це джерело, надходять до кодеру.

Початково потік некодованих кадрів може сприйматися як масив I-кадрів. Далі, керуючись опціями кодування, зокрема, частотою χ слідування кадрів у потоці та розмірністю G групи, кодер обирає опорні кадри. Після цього різниця зміщення відеосцени між I-кадрами груп G_i та G_{i+1} у потоці компенсується [14-16]. Даний технологічний етап кодування має назву компенсації руху. У результаті його виконання якраз і утворюється множина P та в-кадрів групи. Отже, шифрувати ключові кадри слід ще до етапу

компенсації руху, що веде до повного закриття відеопотоку без обов’язкової обробки усіх кадрів у групі.

1.7 Висновки за розділом

Було виявлено, що однією з ключових вимог до шифрування відео контенту є забезпечення мінімізації внесеної затримки. Дана затримка зумовлюється тим, що саме шифрування являє собою окремий та додатковий технологічний процес, що має виконуватися послідовно з процесом кодування відео.

При цьому, найбільш складно досягти мінімізації внесеної затримки і тим самим – створити умови для можливості трансляції відео у реальному часі для випадку відео інтерактивного типу.

Разом з тим, архітектура E2EE, що забезпечує вищу захищеність даних, ніж для випадку використання P2PE, спричинює вище обчислювальне навантаження на кінцеві пристрої, так як шифрування повністю здійснюється виключно на їх рівні. Це веде до того, що шифрування відеоконтенту UHD-формату, а для кінцевих вузлів низької обчислювальної потужності- формату HD перетворюється на складне завдання. При цьому, виконання умов мінімізації часу шифрування не гарантується.

Водночас, існуючий принцип кодування відео, що використовується платформою MPEG, розглядає потік кадрів як гетерогенну структуру, яка утворюється поєднанням кадрів різних типів, зокрема:

- опорних I-кадрів;
- прогнозованих P та B-кадрів.

Ураховуючи існуючу закономірність, а саме, те, що B та P кадри у групі утворюються у наслідок процедури компенсації руху між сусідніми I-кадрами у потоці, та, фактично, є різницею між ними, для того, щоб повністю закрити відеопотік від зловмисника, достатньо шифрувати ключові, або опорні кадри, без обов’язкового шифрування усіх кадрів потоку. Для цього необхідно шифрувати кадри, що кодер буде використовувати у якості опорних, ще до етапу компенсації руху.

Такий підхід дозволяє:

- суттєво скоротити час шифрування відеопотоку та заощадити обчислювальні ресурси;

- забезпечити можливість шифрування трафіку відео надвисокої роздільної здатності навіть за умови, що кінцевий вузол, який бере участь у захищеному обміні відеоданими, не характеризується збільшеною продуктивністю.

Разом з тим, у ході кодування UHD-відео порядку 8К та вище, за умов збільшеного показника частоти слідування кадрів означеного підходу недостатньо для того, щоб побудувати процес шифрування інтерактивних відеоданих згідно з вимогами (1.5). За цих умов необхідно вирішити такі завдання, як:

- вибір та обґрунтування підходу, що дозволяє мінімізувати кількість обчислювальних операцій у ході шифрування ключових кадрів;

- вибір алгоритму шифрування, який відзначається відносно високою швидкодією та може бути ефективним навіть для умов, коли апаратна платформа, у рамках якої передбачається його реалізація, не відзначається збільшеною продуктивністю.

2 ВИБІР ТА ОБҐРУНТУВАННЯ ПІДХОДУ, ЩО ДОЗВОЛЯЄ МІНІМІЗУВАТИ КІЛЬКІСТЬ ОБЧИСЛЮВАЛЬНИХ ОПЕРАЦІЙ У ХОДІ ШИФРУВАННЯ КЛЮЧОВИХ КАДРІВ

2.1 Огляд існуючої схеми кодоутворення на рівні окремих кадрів потоку, прийнятої у MPEG

У загальному випадку, обробка кожного окремого кадру, як ключового, так і будь-яких інших, у рамках MPEG виконується на основі алгоритму JPEG, до якого, у залежності від конкретного стандарту, можуть додатково включатися ті чи інші додаткові механізми.

Проте, незалежно від конкретної використовуваної MPEG-специфікації, загальний процес обробки кадрів на базі JPEG містить у собі такі технологічні етапи, як (рис.2.1) [15, 17]:

- конвертація колірного представлення з наступним етапом субдискретизації хроматичних компонент;
- сегментація;
- трансформація блоків $\lambda_{x,y}^{(Q)}$ кадру Q на базі дискретного косинусного перетворення (ДКП);
- квантування і округлення;
- кодування без втрат на основі методу Хафмана чи арифметичного кодування.

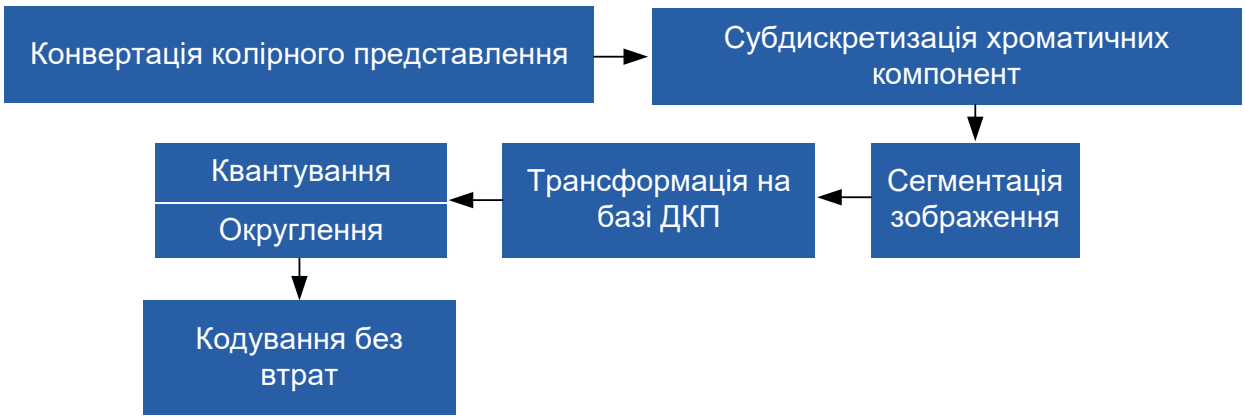

Рисунок 2.1 – Загальний каскад JPEG-перетворень у ході кодування окремих кадрів відеопотоку незалежно від специфікації MPEG

2.2 Сутність технологічних етапів JPEG-перетворення у ході кодування кадру

Так, на етапі конвертації колірного представлення здійснюється перехід від трьохкомпонентної колірної палітри RGB до колірно-різницевого опису YUV або YCbCr.

Фізичний сенс такого переходу пояснюється низьким рівнем сприйняття зором людини градацій колірності і навпаки, високим – градацій яскравості. У даному ж випадку замість компонент R, G та B утворюються компонента яскравості Y та дві хроматичних компоненти [15, 18, 19].

Далі обробка здійснюється таким чином, що хроматичні компоненти зазнають більш суттєвого ущільнення, ніж яскравісна, що створює умови для забезпечення високих рівнів компресії баз візуально помітних викривлень у зображенні, хоча і вносить при цьому незворотні втрати.

У свою чергу, такі незворотні втрати вносяться, у першу чергу, у наслідок виконання субдискретизації хроматичних компонент.

Сутністю даного процесу є співставлення вихідній кількості ε компонент яскравості зменшеного обсягу хроматичних компонент (табл. 2.1).

Таблиця 2.1 – Приклади пропорційного розподілу компонент яскравості та хроматичних у процесі їх субдискретизації

Y	Cb	Cr
ε	$\frac{\varepsilon}{2}$	$\frac{\varepsilon}{4}$
	$\frac{\varepsilon}{2}$	$\frac{\varepsilon}{2}$
	$\frac{\varepsilon}{4}$	$\frac{\varepsilon}{4}$

У свою чергу, технологічний етап трансформування блоків $\lambda_{x,y}^{(Q)}$ кадру Q відеопотоку передбачає їх переведення у частотний (спектральний) формат представлення.

Необхідність виконання даного технологічного етапу зумовлюється тим, що така трансформація дозволяє окремо групувати низькочастотні та високочастотні компоненти $\nu_{i,j}^{(\lambda)}$. Це, у свою чергу, далі дозволяє квантувати

окремо зазначені складові. При цьому, низькочастотні компоненти $\nu_{i,j}^{(\lambda)}$ пригнічуються суттєво менше, ніж високочастотні. Таким чином, на етапі квантування досягаються високі показники стиснення за умови незначної втрати даних та взагалі непомітної візуально. Така залежність пояснюється тим, що високочастотні компоненти не несуть у собі достатнього семантичного змісту [15, 17, 19, 20].

Реалізується процес квантування поділом компонент $\nu_{i,j}^{(\lambda)}$ блоку $\lambda_{x,y}^{(Q)}$ на відповідні коефіцієнти матриць квантування. Отже, даний технологічний процес також вносить незворотні викривлення у кодований кадр Q.

Разом з тим, хоча етап дискретного косинусного перетворення використовується виключно для розкладання компонент $\nu_{i,j}^{(\lambda)}$ блоку $\lambda_{x,y}^{(Q)}$ за частотною ознакою, та не передбачає спрямованого внесення будь-яких змін у оброблювані дані, такі зміни все ж таки вносяться. Їх існування пояснюється тим, що косинус не є цілочисельною функцією. У результаті цього піксель $\mu_{a,b}'^{(Q)}$, який відновлено після JPEG кодування, навіть за умови, що на етапі квантування та субдискретизації компонент жодних змін у кадр не вносилося, не збігатиметься повністю з вихідним $\mu_{a,b}^{(Q)}$ пікселем, тобто, $\mu_{a,b}^{(Q)} = \mu_{a,b}'^{(Q)}$.

Етап кодування без втрат, окрім, безпосередньо, побудови кодограми $E(\lambda_{x,y}^{(Q)})$ блоку кадру Q містить етапи лінеаризації та RLE, що також не вносять незворотніх змін у дані, що кодуються.

Таким чином, у результаті розгляду принципів кодування кадрів відеопотоку на засадах JPEG, було виявлено, що такі технологічні етапи їх кодування, як субдискретизація хроматичних компонент, квантування, округлення та трансформація на базі ДКП вносять незворотні зміни.

Звідси виходить, що шифрування опорних кадрів потоку необхідно виконувати в один з наведених далі способів, а саме (рис.2.2):

- шифрувати блоки кадру на етапі після виконання квантування та округлення, але до етапу кодування без втрат;

- формувати шифрограму після етапу кодування без втрат.

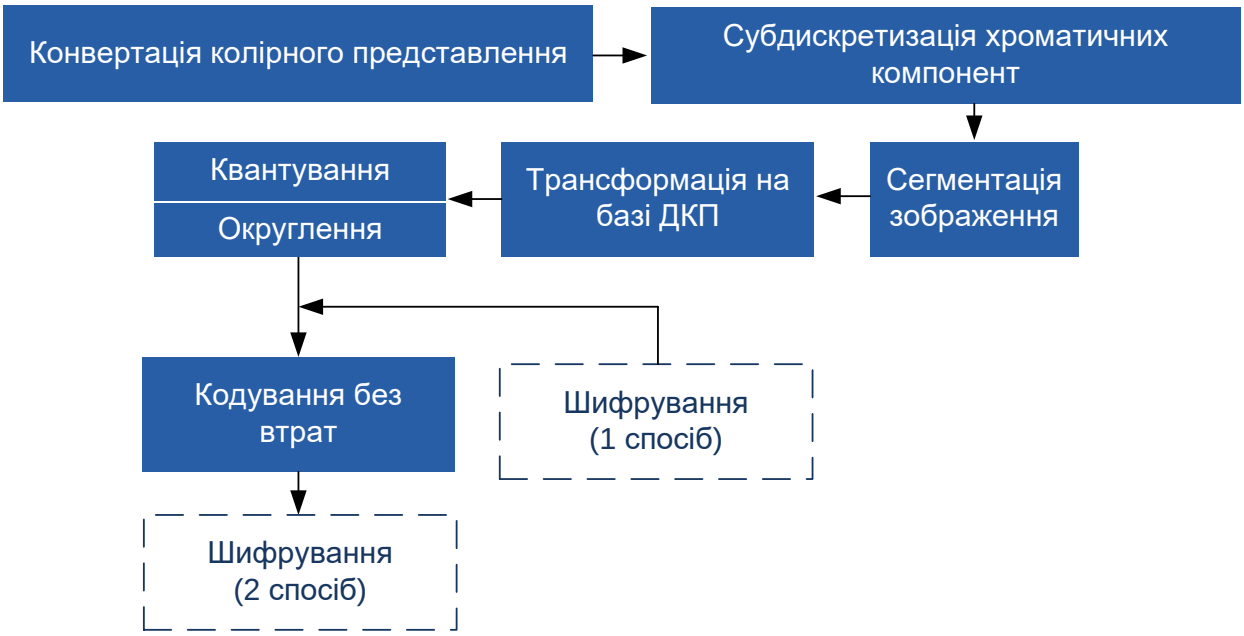

Рисунок 2.2 – Способи організації технологічного етапу шифрування даних на базі JPEG-платформи

У будь-якому випадку з зазначених, шифруванню буде підлягати вектор десяткових величин (компонент). При цьому, на опис кожної компоненти $\nu_{i,j}^{(\lambda)}$ у загальному випадку резервується 8 біт.

Розмірність такого вектору у випадку шифрування блоків $\lambda_{x,y}^{(Q)}$ кадру Q одрзу після операцій квантування та округлення теоретично знаходиться у діапазоні від 64 до 1, проте найчастіше це діапазон значень від приблизно 50-55 до 30-25 [17].

У свою чергу, якщо шифрування виконується після етапу кодування без втрат, тоді обробці буде підлягати менша кількість десяткових величинг, певний обсяг яких залежить від особливостей змісту блоку $\lambda_{x,y}^{(Q)}$ після етапу округлення.

Таким чином, існуючий каскад JPEG-перетворень вимагає виконання наступного:

- застосування сторонніх алгоритмів шифрування;

- включення до складу шифрограми усіх 64 компонент $\nu_{i,j}^{(\lambda)}$ блоку $\lambda_{x,y}^{(Q)}$ за умови шифрування перед етапом стиснення без втрат;

- шифрування множини γ десяткових чисел, для яких є справедливим наступне (коли шифрування виконується після етапу кодування без втрат):

$$\gamma \le 64. \tag{2.1}$$

Зрозуміло, що з точки зору забезпечення мінімізації загального часу на обробку відеоданих, доцільнішим є шифрування даних після того, як їх було стиснено без втрат.

2.3 Оцінка потенційно можливого обсягу даних, який необхідно шифрувати, на випадок застосування сторонніх алгоритмів разом з базовою платформою JPEG

Оцінку ймовірного обсягу даних для шифрування виконаємо для випадків потоку відеокадрів форматів FullHD (2K) та UHD (4K).

При цьому, у загальному випадку кадр містить у собі множину ξ блоків $\lambda_{x,y}^{(Q)}$, яка розраховується як [16]:

$$\xi = \frac{H}{n} \times \frac{W}{n}, \tag{2.2}$$

де H та W - висота та ширина кадру у пікскелях;

n - розмірність блоку.

У першому випадку, за умови роздільної здатності FullHD (1920x1080 пікселів), та розиірності блоку 8x8, кадр міститиме у собі $\frac{1920}{8} \times \frac{1080}{8} = 32400$ блоків.

Відповідно, у більш раціональному випадку, тобто, якщо при цьому шифруються блоки $\lambda_{x,y}^{(Q)}$, що були попередньо стиснені без втрат, шифрограма буде охоплювати такий обсяг σ даних, як:

$$\sigma = \xi \times \bar{\gamma},\ \bar{\gamma} < 64, \tag{2.3}$$

де $\bar{\gamma}$ - усереднене значення множини γ десяткових чисел, що підлягають шифруванню.

На відміну від виразу (2.1), величина $\bar{\gamma}$ є меншою, ніж 64, так як робиться припущення, що за найгірших умов принаймні у складі одного блоку $\lambda_{x,y}^{(Q)}$ міститиметься кількість γ десяткових величин, що буде меншою, ніж 64.

Проте, оскільки розрахунки ведуться для найгірших умов, надалі вважаємо, що $\bar{\gamma} = 63$.

Отже, для одного FullHD кадру I-типу за виразом (2.3) шифруванню підлягатиме щонайбільше $\sigma = 32400 \times 63 = 2041200$ біт, (0,25515 Мб).

Далі оцінимо, який саме об'єм даних необхідно зашифрувати за секунду для випадку транслювання інтерактивного відео. Для цього попередньо визначимо, яка кількість N_{key} ключових кадрів належите до даного часового інтервалу.

У свою чергу, для цього може бути використано наступний вираз [17]:

$$N_{key} = \text{trunc}\left(\frac{\chi}{G}\right). \tag{2.4}$$

Тоді, відповідно, щосекунди шифруванню буде підлягати $\sigma(t)$ біт, що визначається як:

$$\sigma(t) = \sigma \times N_{key}. \tag{2.5}$$

Таким чином, якщо мова йде про потік FullHD, для якого частота слідування кадрів χ=30, отже за умови, що розмірність блоку приймається рівною $n = 8x8$, для групи розмірністю $G = 8$ маємо $N_{key} = \text{trunc}\left(\frac{30}{8}\right) = 3$.

Тобто, $\sigma(t) = \sigma \times N_{key} = 2041200 \times 3 = 6123600$ біт (0,76545 Мб).

За тих же самих умов, тобто, при χ=30, але з групою максимальної розмірності $G = 32$ за секунду підлягає шифруванню $N_{key} = \text{trunc}\left(\frac{30}{32}\right) \approx 1$ ключовий кадр. Отже, у цьому випадку шифрується $\sigma(t) = \sigma \times N_{key} = 2041200 \times 1 = 2041200$ біт (0,25515 Мб).

Далі виконаємо оцінку необхідної для шифрування кількості біт для випадку обробки опорних кадрів роздільної здатності 4К. При цьому

розрахунки ведуться для випадку т.з. повнокадрового 4К-формату з роздільною здатністю 4096х3072 пікселів.

Так, кадр даної роздільної здатності міститиме у собі $\xi = \frac{4096}{8} \times \frac{3072}{8} = 196608$ блоків розміром 8х8 піскелів.

У свою чергу, для $\bar{\gamma} = 63$ на рівні одного ключового кадру шифруванню за виразом (2.3) підлягає $\sigma = 196608 \times 63 = 12386304$ біт (1,548288 Мб).

Відповідно, якщо для групи мінімальної розмірності, тобто, при $G = 8$, необхідно шифрувати $N_{key} = 3$ ключових кадри, тоді протягом однієї секунди шифруватися має $\sigma(t) = \sigma \times N_{key} = 12386304 \times 3 = 37158912$ біт (4,644864 Мб).

У свою чергу, коли мова йде щодо грипи з $G = 32$, величина $\sigma(t)$ згідно з виразом (2.5) дорівнюватиме $\sigma(t) = 12386304$ біт, або 1,548288 Мб.

Результати оцінки ймовірного обсягу даних для шифрування у випадку відеокадрів форматів FullHD (2K) та UHD (4K), кодованих у класичному базисі JPEG наведено табл.2.2.

Таблиця 2.2 – Потенційно можливий об’єм даних, які підлягають шифруванню, для випадку відеокадрів форматів 2K та 4K за умови, що χ=30

Роздільна здатність	Окремий ключовий кадр	$G = 8$	$G = 32$
2K	0,25515 Мб	0,76545 Мб	0,25515 Мб
4K	1,548288 Мб	4,644864 Мб	1,548288 Мб

2.4 Висновки за розділом

Досліджено ключові етапи кодування окремих кадрів з використанням загального касдкаду перетворень JPEG.

Виявлено, що ураховуючи ключову особливість JPEG, а саме – можливість внесення незворотніх змін у кодовані дані, шифрування доцільно виконувати як окремий технологічний етап, що або передує етапу стиснення без втрат, або виконується після нього. Проте, з точки зору скорочення

обсягу операцій у ході шифрування, більш доцільним можна вважати перенесення етапу шифрування після стиснення без втрат.

Виконано оцінку потенційно можливого об’єму даних у межах опорних кадрів відеопотоку, який необхідно шифрувати для випадку, коли шифрограма формується відносно даних, які попередньо було стиснено без втрат на останньому етапі JPEG-кодування. При цьому показано, що найбільший обсяг даних при сталому стандартизованому показникові частоти кадрів відповідає випадку групи мінімального розміру. Водночас, для груп максимального розміру обсяг даних, що шифруються щосекунди, буде таким, як і для єдиного ключового кадру.

Разом з тим, навіть за умови, що шифруванню підлягають виключно ключові кадри групи, обсяг даних, які при цьому підлягають обробці, все ще може залишатися значним, особливо для випадку формату 4К. Відтак, зберігається ризик того, що загальна затримка передавання пакетів у наслідок значного часу формування шифрограм може перевищувати допустимі показники.

3. ДОСЛІДЖЕННЯ ПОШИРЕНИХ АЛГОРИТМІВ ШИФРУВАННЯ ДАНИХ

3.1 АЛГОРИТМ RSA

Алгоритм RSA, назва якого походить від абревіатури прізвищ авторів - Rivest, Shamir та Adleman належить до групи алгоритмів криптографії з ключем відкритого типу (відкритим ключем). Даний алгоритм базується на тому факті, що рішенню завдань факторизації досить великих цілочисельних величин відповідає значна обчислювальна складність [21, 22]. Тобто, процес розшифрування даних зловмисником потребує:

- значних обчислювальних потужностей;
- суттєвих часових витрат.

Інакше кажучи, вважається, що до моменту розшифрування зловмисником закритих даних їхня цінність буде, у сутності, нульовою.

RSA, як криптосистема, є першим засобом шифрування, на базі якого однаково ефективно вирішується такі завдання, як:

- шифрования інформації;
- створення цифрового підпису.

3.1.1 Загальний принцип роботи алгоритму RSA

Алгоритми шифрування, що використовують ключ відкритого типу, базуються на так званих односторонніх функціях, шо характеризуються такими властивостями, як [6, 22, 23]:

1. За умови, що значення величини x є відомим, може бути обчислено $f(x)$ без будь-яких труднощів.
2. Якщо існує відоме $y = f(x)$, за цих умов знаходження x є нетривіальним завданням, для якого відсутні прості рішення.

У даному випадку термін ***односторінність*** означає не односпрямованість функції, яка є математично обґрунтованою.

Тут односторонність вказує на те, що в умовах застосування існуючих обчислювальних засобів та у рамках наявного технологічного базису розрахувати зворотнє значення функції за деякий локальний часовий інтервал практично неможливо.

Як вже згадувалося раніше, головна ідея RSA полягає у тому, що завдання факторизації великих чисел потребує значних обчислювальних потужностей та часових ресурсів, що зумовлюється його високою обчислювальною складністю. При цьому, в основі процесу побудови шифрограми знаходиться операція обчислення ступеню за модулем великого числа. У свою чергу, для відкриття шифрограми, тобто, виконання оберненої дії, має бути обчислено значення функції Ейлера від згаданого великого числа, що потребує відомостей щодо особливостей розкладання цього числа на прості множники. При цьому, процес такого розрахунку повинен виконуватися за відносно невеликий часовий проміжок.

При цьому, у рамках типової криптографічної системи, що використовує ключ відкритого типу, кожна зі сторін обміну даними має у своєму розпорядженні ключі двох типів, а саме:

- ключ відкритого типу p, або public key;
- ключ закритого типу s (private key).

Алгоритм RSA передбачає, що кожний з ключів формується парою цілочисельних величин.

Водночас, кожна зі сторін обміну даними самостійно генерує ключі відкритого та закритого типів.

При цьому, ключ закритого типу, або приватний ключ, другій стороні не повідомляється. На відміну від приватного ключа, публічний ключ може бути повідомлено будь-кому або взагалі опубліковано.

Разом два дані ключі кожної зі сторін обміну інформацією у рамках алгоритму RSA формують так звану ***узгоджену пару***. Іншими словами, ключі є взаємно оберненими.

Це означає, що для кожної пари (p,s) ключів публічного p та приватного s типів існують такі шифрувальні $E_p(x)$ та дешифрувальні $D_s(x)$ функції, що для буль-якого повідомлення m, що належить множині M допустимих повідомлень, забезпечується справедливість рівності:

$$m = D_s(E_p(m)) = E_p(D_s(m)). \qquad (3.1)$$

Будь-якому сеансу передавання шифрованих даних передує етап формування відкритого та закритого ключів

3.1.2 Формування публічного та приватного ключів

У загальному випадку, процес формування ключів RSA містить у собі ряд технологічних етапів, а саме [6, 22, 23]:

- вибір пари випадкових простих чисел α та β попередньо встановленого розміру Z; у багатьох існуючих сьогодні RSA-орієнтованих алгоритмах зазвичай приймається $Z = 2048, 4096$ і.т.д.;

- виконання розрахунку модулю θ, що являє собою добуток чисел α та β, тобто:

$$\theta = \alpha \times \beta; \tag{3.2}$$

- розрахунок значення функції Ейлера у залежності від величини θ модулю, а саме:

$$\psi = (\alpha - 1) \times (\beta - 1); \tag{3.3}$$

- вибір цілочисельного значення e (т.з. «***відкрита експонента***»), яке є взаємно простим з попередньо розрахованим значенням функції $\psi(\theta)$. Тобто, як e, так і $\psi(\theta)$, не повинні мати жодних інших спільних дільників, окрім 1. При цьому, величина e має відповідати наступній умові:

$$1 < e < \psi. \tag{3.4}$$

Також додатковою вимогою до величина e може бути незначна кількість одиниць у двійковому форматі її опису. З цієї точки зору доцільним є вибір значення e з множини т.з. чисел Ферма, тобто, величину e буде сформовано за наступним принципом:

$$e = 2^{2^n} + 1,\ n \geq 0. \tag{3.5}$$

Нерідко необхідне значення величини e, згенероване за принципом, поданим виразом (3.5) обирається серед величин 17 (при n=2), 257 (n=3), або 65537 (n=4). Це зумовлено тією обставиною, що за даного значення e може бути досягнуто балансу між часом, який витрачається для побудови

шифрограм на базі швидкого зведення до ступеня, та рівнем захищеності шифрограми. У даному випадку час побудови шифрограми буде мінімально можливим за умови високого ступеню її захищеності.

Разом з тим, у випадках, коли, наприклад, $e=1$ (n=0), або $e=3$ (n=1), складаються умови, у яких рівень захищеності шифрограми на базі RSA може суттєвим чином погіршуватися [16];

- знаходження деякого числа δ, що задовольняє наступне порівняння:

$$(\delta \times e) \equiv 1 \mod(\psi(\theta)), \tag{3.6}$$

тобто, число δ повинно бути мультиплікативним щодо величини e, а остача від ділення добутку $(\delta \times e)$ за модулем $\psi(\theta)$ повинна рівнятися 1. Тут число δ далі має назву ***секретної експоненти***, знаходження якої виконується з застосуванням розширеного Евклідового алгоритму;

- публікування пари значень $(e;\theta)$ як публічного, або відкритого ключа;

- використання пари значень $(\delta;\theta)$ як приватного ключа.

3.1.3 Загальний принцип шифрування та дешифрування повідомлень на базі RSA

Нехай у загальному випадку необхідно надіслати деяке повідомлення від Користувача 1 до Користувача 2.

На цей випадок на боці Користувача буде виконано такі дії, як:

- отримання публічного ключа $(e;\theta)$ Користувача 2;
- формування відкритого повідомлення m;
- побудова шифрограми c повідомлення на базі ключа $(e;\theta)$ як:

$$\mathrm{c} = \mathrm{E}(\mathrm{m}) = \mathrm{m}^{e} \mod \theta. \tag{3.7}$$

У свою чергу, дешифрування повідомлення m Користувачем 2 потребує виконання таких кроків:

- отримання шифрограми c;
- застосування приватного ключа $(\delta;\theta)$ для відкриття повідомлення за принципом:

$$m = D(c) = c^{\delta} \mod \theta . \qquad (3.8)$$

Даний процес може бути схематично проілюстровано рис.3.1.

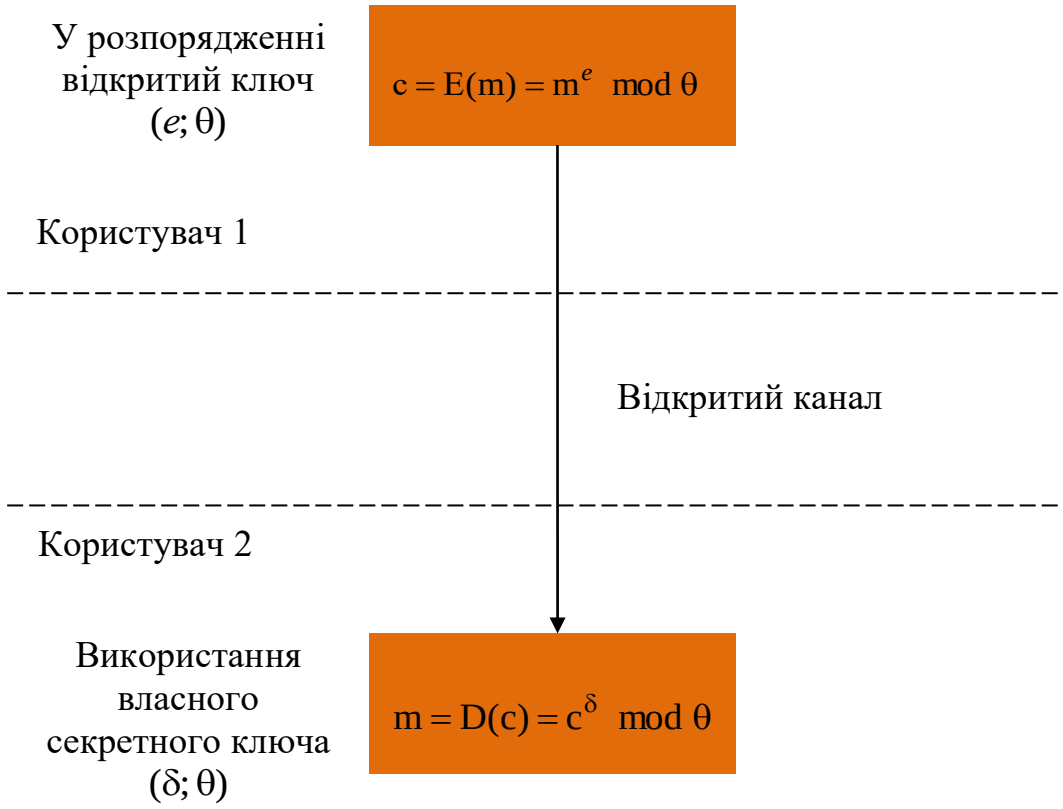

Рисунок 3.1 – Загальна схема формування шифрограми повідомлення та його дешифрування з використанням алгоритму RSA

Разом з тим, практична реалізація алгоритму RSA на сьогодні передбачає використання т.з. ***сеансового ключа***. Даний ключ існує виключно протягом єдиного сеансу, та використовується для шифрування повідомлень симетричними системами учасників сеансу обміну даними. По закінченню сеансу такий ключ знищується.

У загальному випадку для того, щоб побудувати шифрограму c повідомлення m, на боці джерела повідомлення (Користувача 1) виконуються наступні дії:

- формування випадкового сеансового ключа ζ;
- закриття ключа ζ на базі ключа $(e;\theta)$ відкритого типу, а саме:

$$\omega = \mathrm{E}(\zeta) = \zeta^{e} \mod \theta; \tag{3.9}$$

- безпосереднє шифрування повідомлення m на базі сеансового ключа відповідно до симетричного алгоритму:

$$\mathrm{c} = \mathrm{E}_{\zeta}(\mathrm{m}). \tag{3.10}$$

У свою чергу, для того, щоб у такій схемі обміну даними Користувач 2 мав можливість дешифрування прийнятого повідомлення m, виконується зазначений перелік операцій:

- прийом попередньо сформованої Користувачем 1 шифрограми ω сеансового ключа ζ;
- дешифрування шифрограми ω з використанням ключа $(\delta; \theta)$ закритого типу, що еквівалентно виразу:

$$\zeta = \mathrm{E}(\omega) = \zeta^{\delta} \mod \theta; \tag{3.11}$$

- дешифрування повідомлення c, користуючись дешифрованим сеансовим ключем на базі симетричного алгоритму, як показано наступним виразом:

$$\mathrm{m} = \mathrm{D}_{\zeta}(\mathrm{c}). \tag{3.12}$$

При цьому, якщо справедливим є наступне співвідношення:

$$\zeta > \theta, \tag{3.13}$$

тобто, розмір сеансового ключа перевищує величину модулю θ, попередньо виконується поділ сеансового ключа на множину $\{\chi\}$ сегментів необхідної довжини θ, яких буде $\frac{\zeta}{\theta}$. Далі, якщо виконується наступна умова:

$$\mathrm{mod}\left(\frac{\zeta}{\theta}\right) > 0, \tag{3.14}$$

це може означати, що для довжини ℓ останнього сегменту з множини $\{\chi\}$ справедливо:

$$\ell < \theta. \tag{3.15}$$

У зазначених умовах $(\theta - \ell)$ біт даного сегменту може бути доповнено нульовими бітами. Далі відносно цього сегменту застосовується шифрування аналогічно, як і у випадку усіх інших сегментів множини $\{\chi\}$.

Схему формування шифрограми повідомлення та його дешифрування з використанням алгоритму RSA для умов застосування сеансового ключа наведено рис.3.2.

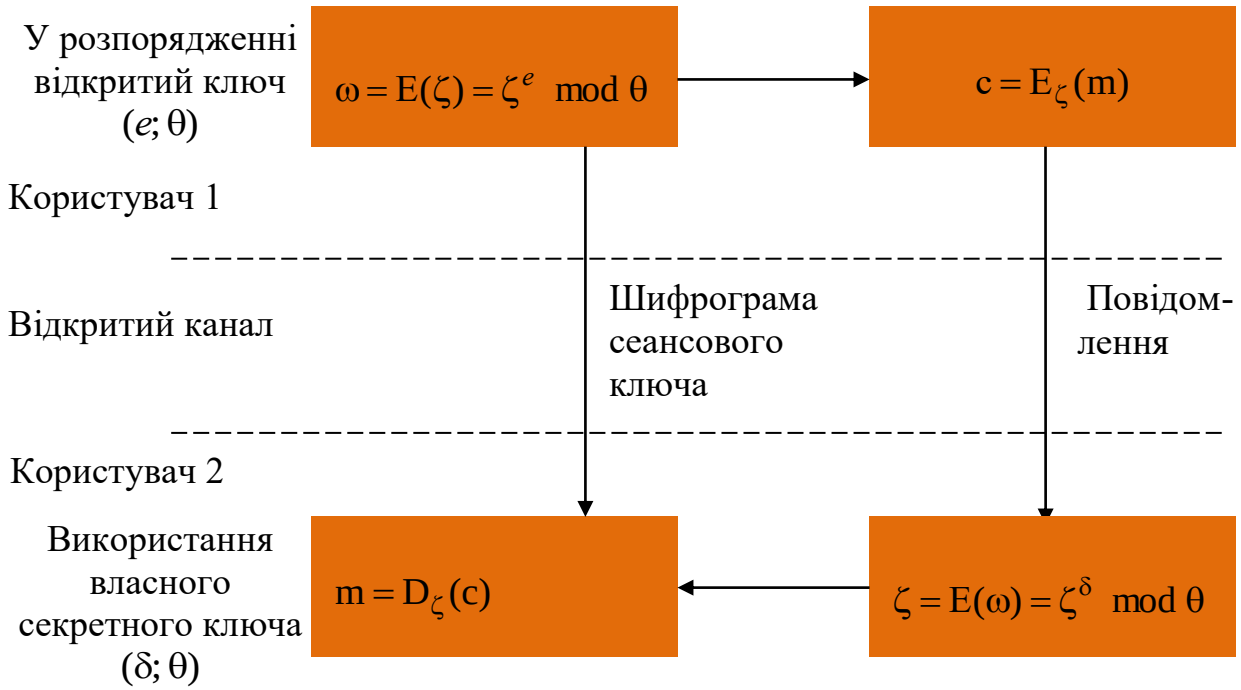

Рисунок 3.2 - Схема формування шифрограми повідомлення та його дешифрування з використанням алгоритму RSA для умов застосування сеансового ключа

3.1.4 Приклади формування шифрограм на базі алгоритму RSA

Приклад 1.

Генерування ключів.

На першому етапі роботи алгоритму створюються відкритий та закритий ключі.

Для цього у загальному випадку може обрано пара простих чисел – наприклад, $\alpha = 3$ і $\beta = 7$. Далі за виразом 3.2) розраховується значення модулю θ, а саме $\theta = 3 \times 7 = 21$. Після цього для обраних α та β знаходимо значення функції Ейлера за виразом (3.3), тобто $\psi = (3-1) \times (7-1) = 12$.

На наступному кроці виконання алгоритму виконується вибір величини e, що задовольняє вимозі, поданій виразом (3.4), і окрім цього обов'язково має бути:

- простою;
- взаємно простою щодо ψ.

Тобто, за умовою (3.4) величина e у нашому випадку має бути меншою, ніж 12 та більшою 1, що дає підставу попередньо розглядати такі прості числа, як 3, 5, 7 та 11. Далі з урахуванням вимоги відносно взаємної простоти з ψ діапазон можливих значень e звужується до 5, 7 та 11. Розглянемо варіант застосування числа 5 з даного діапазону у якості відкритої експоненти. Для цього випадку відкритий ключ $(e; \theta)$ сформовано числами 5 та 21. Даний ключ буде передано віддаленому користувачеві для того, щоб з його використанням шифрувати повідомлення. Після цього необхідно сформувати ключ закритого типу.

У свою чергу, приватний ключ потребує розрахунку числа δ, яке розраховується за виразом (3.6) та являє собою величину, зворотню до e за модулем ψ, іншими словами, щоб значення залишку від ділення за модулем ψ добутку $(\delta \times e)$ рівнялося одиниці.

Для спрощення виразу застосуємо позначення, прийняті у багатьох мовах програмування.

Таким чином, вираз $(\delta \times e) \equiv 1 \mod(\psi(\theta))$ буде подано у вигляді $(\delta \times e) \,\%\, \psi = 1$. Отже, тоді маємо $(\delta \times 5) \,\%\, 12 = 1$. У цьому випадку δ може бути рівною 5, так як $(5 \times 5) \,\%\, 12 = 25 \,\%\, 12 = 1$.

Разом з тим, значення δ тут може бути також 17, оскільки $(17\times 5)\%12 = 85\%12 = 1$. Таким чином, ключом закритого типу у нашому випадку може бути пара значень $(e;\theta)$, рівна $(17;21)$. Цей ключ зберігається локально.

Шифрування повідомлення.

Найпростішим випадком повідомлення може бути деяке число. Розглянемо для прикладу випадок, коли необхідно шифрувати число 19. Тобто, $m = 19$.

На першому кроці шифрування, як показано виразом (3.7), шифрограма c отримується шляхом зведення величини m до ступеню e за модулем θ, отже, необхідно розрахувати значення 19^5 і далі обчислити остачу від ділення на 21. Тобто, шифрограма для цього випадку дорівнюватиме $c = 19^5 \% 21 = 10$.

Дешифрування повідомлення.

Користувач, який має у своєму розпорядження шифроване повідомлення та відкритий ключ, не може виконати процедуру дешифрування.

Сам процес розшифрування повідомлення, згідно з виразом (3.8), у нашому випадку зводиться до розрахунку остачі від ділення шифрограми c у ступені δ за модулем θ, тобто, повідомлення m відновлюється як: $m = 10^{17} \% 21 = 19$.

Оскільки при цьому вважається, що будь-хто інший, окрім Користувача 2, який є приймачем повідомлення, не має у своєму розпорядженні закритого ключа, теоретично ніхто інший не здатен дешифрувати отримане повідомлення.

Це, як зазначалося вище, зумовлюється високою обчислювальною складністю генерування ключа закритого типу на базі відкритого ключа. Попри те, що формування ключів обох типів здійснюється на базі простих чисел α і β, тобто, існує взаємозв'язок між ключами, пошук даного зв'язку є досить складним багатокроковим завданням. Найбільшу складність являє собою декомпозиція модуля θ з виокремленням з нього простих складових, тобто - α і β. При цьому за умови, що модуль θ утворено на базі двох дуже великих простих чисел, процес його розкладання на множники з подальшою перевіркою вірності знайдених пар може потребувати надмірного обсягу часових та обчислювальних ресурсів.

Розглянемо приклад модулю $\theta = 360$. Покроково розглянемо процес його розкладання на множники:

1. Число є парним, тобто, перший множник – 2.

2. Другий множник, 180 – також ділиться на 2.

3. Число 90 є також парним, що дає ще одне число 2.

4. Число 45 є непарним, але ділиться на 3.

5. Величина 15, отримана на попередньому кроці, також ділиться на 3.

6. Число 5 є простим.

Декомпозиція даного числа є досить простою - $\theta = 360 = 2 \times 2 \times 2 \times 3 \times 3 \times 5$.

Разом з тим, на випадок числа 361 завдання суттєво ускладнюється, так як воно не є парним, і перший множник, який може бути знайдено шляхом перебору – 19. На випадок же ще більших простих чисел вважається. що зловмиснику не вистачить часу та обчислювальної потужності системи для того, щоб розшифрувати повідомлення за деякий лімітований час.

Приклад 2.

Для оцінки процесу шифрування більш складної послідовності розглянемо випадок, коли повідомлення являє собою послідовність символів алфавіту. Наприклад – О, К, Т, Р.

Для цього випадку тепер використовуються ключі, побудовані на базі простих чисел α і β, що дорівнюють 17 та 19 відповідно. При цьому, керуючись виразами (3.2) та (3.3), формуються такі ключі, як:

- $(e; \theta) = (5; 323)$ - публічний;
- $(\delta; \theta) = (173; 323)$ - приватний.

Для побудови шифрограми далі необхідно застосувати операцію переведення зазначених літер алфавіту до цифрового вигляду. У найпростішому випадку тут може бути використано нумерацію цих літер, тобто,отримується послідовність 15, 11, 19, 17.

У результаті застосування відкритого ключа $(e; \theta) = (5; 323)$ на боці Користувача 1 утворюється шифрограма 272, 304, 197, 2. Її може бути надіслано Користувачеві 2 відкритим каналом, після чого з використанням приватного ключа $(\delta; \theta) = (173; 323)$ повідомлення буде дешифровано.

Недоліком розглянутого способу побудови шифрограми тексту є пряме співставлення окремій літері її шифрованого значення. Таким чином у випадку, коли зловмисник виконає перехоплення деякої чистини

повідомлення, йог зміст може бути розкрито без пошуку секретного ключа. Це зумовлюється тим, що:

- у загальному випадку текст обов'язково містить символи «пробіл», що вказують на межі слів;

- у тексті містяться сполучники а також відносно короткі слова, що часто згадуються у тексті.

Таким чином, керуючись синтаксисом та семантикою тексту, та маючи у своєму розпорядження короткі слова, зловмисник шляхом недовгого перебору зможе відновити текст цілком.

Для усунення випадків розшифрування тексту за зазначеними ознаками, необхідно застосувати додаткові механізми, що встановлюють залежність між окремими складниками повідомлення за деяким законом, що у загальному випадку пояснюється виразом:

$$b := f(b; a; k), \tag{3.16}$$

де a – передуючий елемент повідомлення;

b – наступний елемент повідомлення;

k – елемент ключа шифрування.

Наприклад, елементи повідомлення може бути змінено на базі виразу $b := (b + a)\,\%\,\theta$.

При цьому на етапі обробки повідомлення, що передує безпосередньо шифруванню, на базі виразу (3.16) послідовність 15, 11, 19, 17 зазнає змін. Тут складник 15 не буде змінено, складник 11 зміниться на 26, так як $(11+15)\,\%\,323 = 26$, а 19 – на 45 $((26+19)\,\%\,323 = 45)$. У свою чергу, замість 15 буде утворено складник $((45+17)\,\%\,323 = 62)$.

Отже, утворена таким чином послідовність на базі вихідних складових тексту не має характерних ознак належності до буквенного алфавіту. При цьому, на кожну наступну величину впливають усі попередні.

У свою чергу, на прийомному боці необхідно виконувати обернену процедуру, що у нашому випадку задається виразом $b := (b - a)\,\%\,\theta$. Після цього може бути одержано вихідну послідовність символів.

Окрім зазначеного, нерідко практикується:

- додавання «нульових» символів у алфавіт, що веде до зміщення нумерації існуючих літер;

- додавання випадкових символів у початок змістовного повідомлення, яке має бути зашифроване.

У свою чергу, практичне застосування алгоритму RSA дещо відрізняється від розглянутих прикладів, так як обсяги даних, які необхідно шифрувати, можуть мати значний розмір.

За таких умов послідовно виконуються такі операції, як:

- вирівнювання даних;
- сегментація блоків даних;
- змішування блоків за принципом, поданим виразом (3.16);
- безпосереднє RSA-шифрування

3.2 Недоліки алгоритму RSA

3.2.1 Недоліки процедури генерування простих чисел для обчислення модулю RSA

Як зазначалося, ключовим етапом роботи алгоритму є формування модулю θ, що утворюється за участю простих чисел α і β.

Відтак, обрані числа α і β у цілому визначають ступінь захищеності шифрограми, що накладає ряд вимог відносно (рис.2.3):

- процесу підбору зазначених величин, що може бути зазначено виразом:

$$t_{gen} \to \min, \tag{3.17}$$

де t_{gen} - час, протягом якого генеруються величини α і β;

- самих значень α і β.

При цьому, виконання умови (3.17) гарантує, що даний алгоритм RSA може бути ефективно використано для шифрування потокового трафіку – голосу, відео та даних будь-якого типу, для яких необхідно забезпечити обробку у реальному часі.

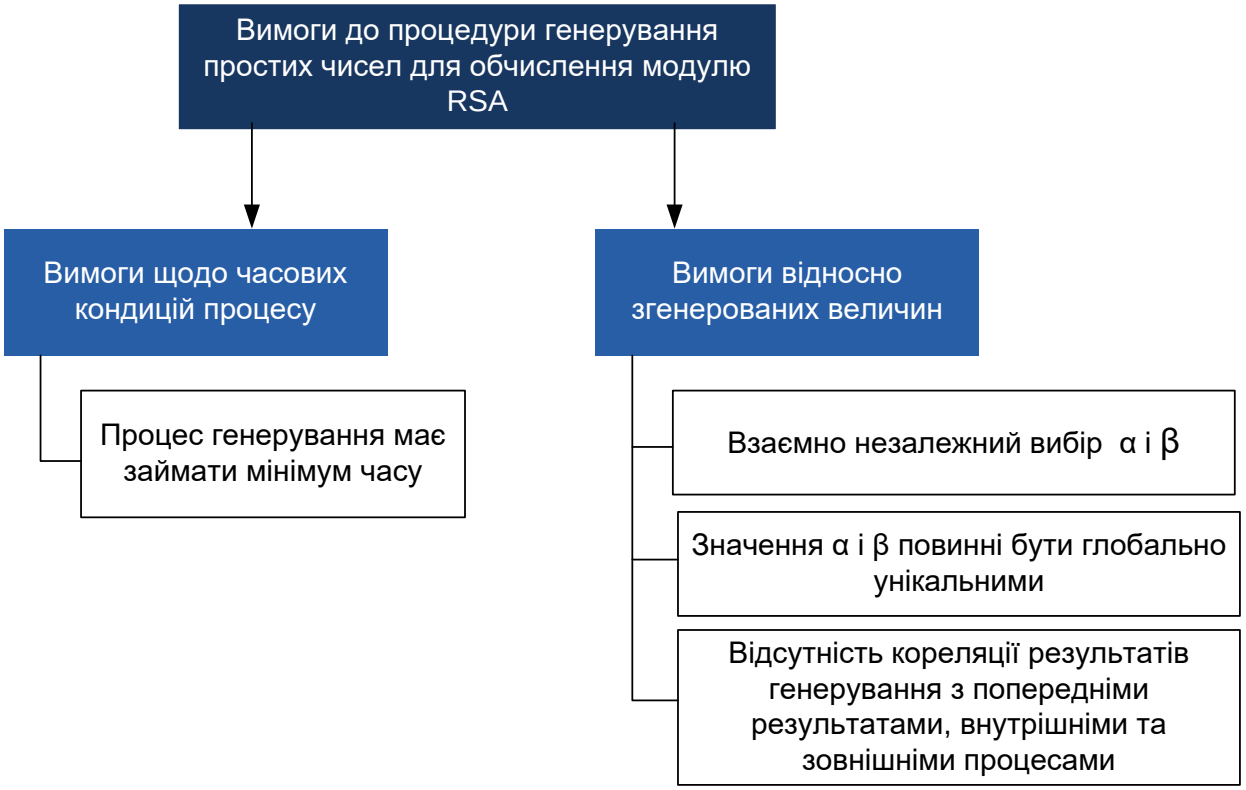

Рисунок 3.3 – Ключові вимоги до процесу вибору простих чисел для розрахунку модулю RSA

Насправді ж, процес пошуку величин α і β, що потенційно здатні забезпечити високий ступінь захищеності, на практиці займає значно більше часу, ніж для випадків інших криптографічних протоколів, для яких достатньо задіяти лише декілька випадкових згенерованих біт.

У свою чергу, щоб створені α і β дозволили побудувати стійку до зламу шифрограму, необхідно забезпечити виконання додаткових вимог, зокрема [6]:

- для вибору чисел α та β необхідно використовувати генератори простих випадкових величин, результат роботи яких не має жодної чіткої кореляції з попередніми результатами та не прив'язаний до жодних внутрішніх чи зовнішніх обчислювальних процесів;
- пара значень α і β має бути глобально унікальною;
- вибір α повинен здійснюватися незалежно від β і також навпаки.

Проте, нерідко замість генерування випадкової простої величини можуть застосовуватися числа, створені за певними алгоритмами. Такий

підхід є потенційно небезпечним, оскільки зловмиснику у тій же мірі можуть бути відомі більшість таких алгоритмів, що веде до швидкого знаходження α і β на у підсумку – розкриття шифрограми.

При цьому, вимога щодо унікальності пари простих величин пояснюється тим, що за її виконання створюються умови для унеможливлення підбору α і β. Якщо ж, навпаки, комбінація простих чисел була попередньо використана раніше у складі інших модулів RSA, дану пару складників може бути обчислено на базі алгоритму GCD (greatest common divisor). За існуючими сьогодні даними, у наслідок використання неефективних генераторів випадкових величин, ймовірність розкриття сценарію шифрування/розшифрування є досить суттєвою. Так, з початку 2012 року і понині близько 1% TLS-трафіка зазнало атак з боку зловмисників, які завершилися успіхом [4, 5].

Водночас, існування третьої вимоги щодо незалежного вибору величин α та β пояснюється наступним. Нехай величини α і β спільно використовують приблизно 50% старших біт, необхідних для опису кожної з них. За цих умов значення θ модулю RSA може бути обчислено з застосуванням методу Ферма.

На сьогодні одним з найбільш відомих наслідків застосування неефективних α і β, та, зокрема, відкритої експоненти *e* надмірно малої величини є існування системної уразливості ROCA, що суттєво збільшує ймовірність викриття шифрограми в умовах, коли зловмиснику відомий ключ відкритого типу [24].

3.2.2 Недоліки вибору розмірності секретної експоненти

У загальному випадку, принцип вибору розмірності секретної експоненти δ може бути проілюстровано наступною залежністю:

$$\begin{cases} \delta\uparrow \rightarrow S\uparrow \,\&\uparrow t_d; \\ \delta\downarrow \rightarrow S\downarrow \,\&\downarrow t_d, \end{cases} \tag{3.18}$$

де S - рівень захищеності шифрограми;

t_d - час дешифрування.

Тобто, одночасно зі збільшенням стійкості шифрованих даних до зловмисного впливу, збільшується також час, необхідний для його розшифрування.

З цієї точки зору для розробника є доцільним застосовувати δ відносно невеликої розмірності, що першочергово актуально для енергоефективних пристроїв, наприклад, таких, як смарт-карти [21].

Водночас, існує ймовірність того, що при виконанні наступних умов зловмисник може відновити приватний ключ використовуючи атаку Вінера [25]:

$$\delta < \sqrt[4]{\theta}. \tag{3.19}$$

Окрім цього, загальний принцип алгоритму RSA передбачає таку черговість формування величин, необхідних для його реалізації:

- модуль θ, використовуючи значення α і β;
- публічна експонента e;
- секретна експонента δ.

Слідування даному принципу знижує ймовірність успішної атаки навіть на випадок використання досить невеликого значення δ, якщо при цьому застосовується публічна експонента, яку було сформовано у наслідок коректного вибору α та β, як було показано у п.3.2.1.

Насправді ж розробники нерідко можуть першочергово задати величину δ, після чого далі розраховувати значення e.

3.2.3 Недоліки, зумовлені величиною публічної експоненти

Аналогічно ситуації щодо секретної експоненти δ, для реалізації алгоритму RSA також є властивим принцип взаємозв'язку захищеності та швидкості шифрування/дешифрування та перевірки підписів, як зазначено виразом (3.18). При цьому, значна кількість розробників орієнтуються на застосування публічних експонент незначного розміру. Зазвичай при цьому використовуються прості числа Ферма, серед яких, як вже згадувалося, першими є 3, 17 257 та 65537.

Зрозуміло, що з точки зору збільшення ефективності алгоритму доцільним для застосування є число 65537. Водночас, суттєвий відсоток розробників у ході реалізації RSA обирає $e = 3$, що, у підсумку, спричинює

появу вразливостей у криптографічній системі, побудованій на базі зазначеного алгоритму [21].

3.3 Алгоритм AES

AES, або Advanced Encrypting Standard, на сьогодні являє собою один з найбільш широко застосовуваних алгоритмів шифрування симетричного типу, що використовується зараз у складі значного відсотку криптографічних продуктів.

Даний алгоритм належить до класу блочних, та орієнтований на шифрування 128-бітних блоків даних, кожен з яких утворений масивами фіксованої розмірності, а саме – 4х4 чарунки [23, 26].

Отже, кожна чарунка містить у собі 1 байт даних.

При цьому, алгоритм може використовувати ключі розміром 128, 192 або 256 біт.

Однією з особливостей алгоритму є те. що у ході побудови шифрограми обробляються як окремі байти блоку, так і його стовпці та рядки.

Ще одна особливість полягає у розбитті усього процесу формування шифрограми на т.з. ***раунди шифрування***, кількість N_{rnd} яких визначається розмірністю використаного ключа та ілюструється табл.3.1

Таблиця 3.1 – Кількість раундів виконаня алгоритму AES залежно від розміру використаного ключа

Довжина ключа, біт	N_{rnd}
128	10
192	12
256	14

При цьому, у ході кожного з раундів відносно вихідних даних застосовується каскад перетворень, як показано рис.3.4.

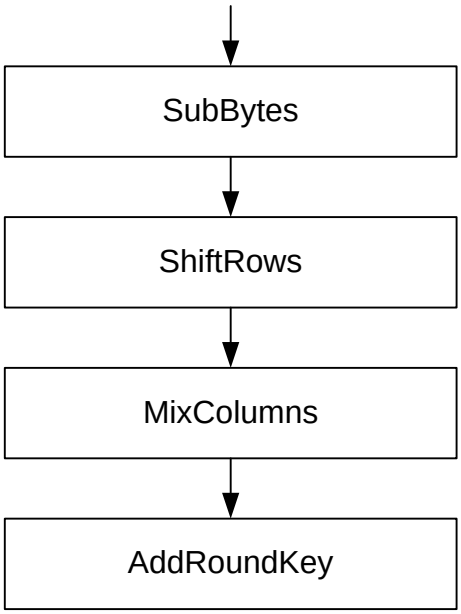

Рисунок 3.4 – Загальна схема виконання технологічних етапів у складі алгоритму AES

Це такі технологічні кроки перетворень, як:

1. SubBytes, у ході якого здійснюється поелементна таблична заміна усіх 16 байт вихідного блоку у спосіб, зазначений рис.3.5.

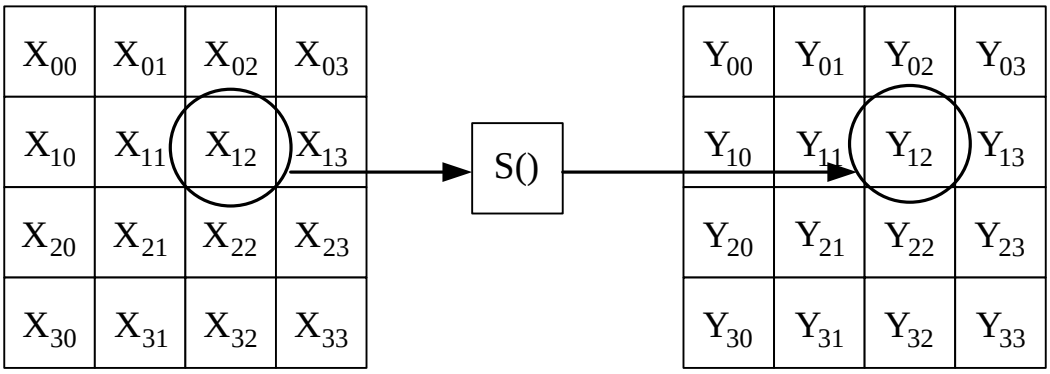

Рисунок 3.5 – Загальний принцип табличної заміни масиву байт вихідного блоку

На рис. 3.5 операнд S() являє собою т.з. таблицю замін S-box. Дана операція є нелінійною, при цьому для побудови таблиці S-box виконується 2 кроки, а саме:

- обчислення зворотнього числа у полі Галуа;
- перетворення кожного байту, що формує S-box, відповідно до виразу:

$$b'_i = b_i \oplus b_{i+4 \bmod 8} \oplus b_{i+5 \bmod 8} \oplus b_{i+6 \bmod 8} \oplus b_{i+7 \bmod 8} \oplus c_i, \quad 0 < i < 8, \tag{3.20}$$

де b_i - і-й біт байту b;

c_i - і-й біт, що належить константі $c = 63_{16} = 99_{10} = 01100011_2$.

Такий підхід до формування S-box дозволяє захистити шифрограми від атак, що базуються на виявленні простих алгебраїчних властивостей.

2. Етап ShiftRows. У ході даного технологічного етапу виконується обробка рядків State.

Дані рядки являють собою проміжний підсумок формування шифрограми, та описується як двовимірний масив байт, що містить у собі 4 рядки та и N_b стовпців.

У ході даного перетворення виконується циклічне зміщення рядків на r байт горизонтально, беручи до уваги початковий номер рядку.

Так, нульовий рядок зміщується на 0 байт, тоді як для першого $r = 1$ байт, другого - $r = 2$ і далі аналогічно для інших рядків (рис. 3.6).

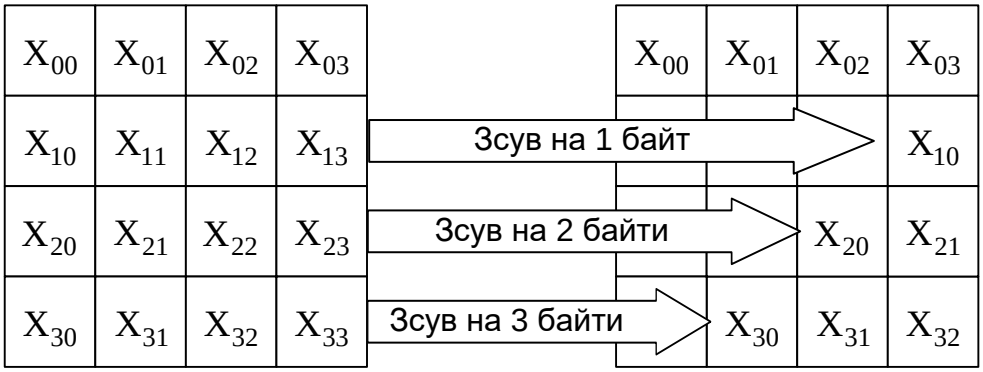

Рисунок 3.6 – Схематичне зображення виконання процесу ShiftRows

3. Етап MixColumns. Протягом цього етапу здійснюється множення за модулем $(x^4 + 1)$ усіх стовпців масиву на поліном $a(x)$, як зазначає наступний вираз:

$$a(x) = 3x^3 + x^2 + x + 2. \tag{3.21}$$

Схематично даний процес може бути зображено рис. 3.7.

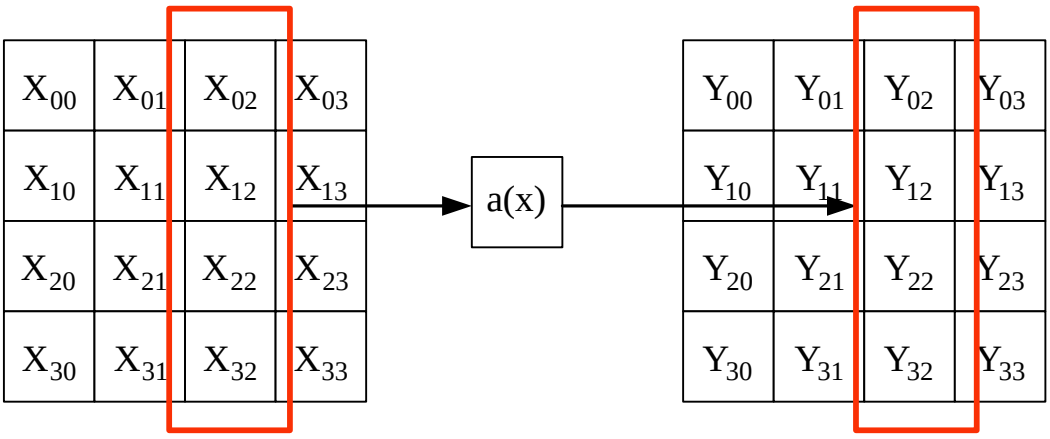

Рисунок 3.7 – Принцип виконання технологічного процесу MixColumns

Так, під час виконання етапу MixColumns здійснюється змішування 4 байт кожного зі стовпців State на базі зворотного лінійного перетворення. При цьому, кожен зі стовпців розглядається у вигляді поліному 4-го ступеню, відносно яких виконується множення у полі Галуа $GF(2^8)$ за модулем (x^4+1) відповідно до виразу (3.21).

4. Даний етап перетворень, а саме, AddRoundKey, передбачає накладення секретного ключа на масив байт.

При цьому, для кожного i-го стовпцю $(i=\overline{0;3})$ побітово застосовується операція XOR з накладенням певного кодового слова ключа W_{4r+i}. У цьому випадку r являє собою індекс поточного раунду виконання шифрування.

Так, для кожного r-го раунду шифрування генерується т.з. RoundKey, для чого відносно CipherKey (тобто, секретного ключа) виконується операція KeyExpansion. Утворений у наслідок цього RoundKey має розмірність, що дорівнює розміру State.

3.3.1 Криптографічна стійкість алгоритму AES

Розглянутому алгоритму AES не властиві недоліки, що характерні для RSA.

При цьому, відносно його високої криптографічної стійкості свідчить те, що з 2003 року АНБ США дозволило застосування AES для захисту даних, що являють собою державну таємницю [23].

При цьому, існують вимоги щодо використання секретних ключів. А саме:

- для шифрування даних рівня SECRET може бути використано ключ довжиною 128 біт;
- дані рівня TOP SECRET необхідно шифрувати ключами довжиною 192 та 256 біт.

У той же час, існує ряд підходів щодо зламу AES-шифрів. Наприклад, неодноразово зазначалося [27] щодо потенційної вразливості шифрограм на базі AES до XSL-атак.

Окрім того, AES-шифри може бути зламано з використанням атак за сторонніми параметрами, а саме – з використанням атаки за часом [28]

3.4 Порівняльний аналіз алгоритмів RSA та AES з точки зору доцільності застосування для побудови шифрограм відеоданих

Аналіз виконується за такими показниками, як:

- ефективність для обробки саме відеоконтенту;
- рівень захищеності шифрограми.

Як свідчить аналіз архітектури алгоритму AES, його застосування передбачає обробку матриць елементів 4x4, для опису кожного з яких застосовано 1 байт.

Таким чином, AES є криптосистемою, що є достатньо адаптованою для обробки відеокадрів, стандартизована сітка розмірів яких є прив’язаною до 8, а опис як яскравісної Y, так і хроматичних Cr, Cb компонент здійснюється 8 бітами.

Отже, у цьому випадку шифрограма 1 блоку відеокадру складатиметься з 4 окремих шифрограм його фрагментів розміром 4x4 кожен.

Водночас, алгоритм RSA потребує додаткової адаптації для ефективного застосування у якості інструментарію шифрування для випадку фронтальної обробки відеоінформації.

Наслідком цього є збільшення часу на виконання самого алгоритму а відтак – збільшення ймовірності того, що загальний час обробки на рівні джерела перевищуватиме значення, встановлене системою QoS.

Врешті решт, ураховуючи також виявлені уразливості алгоритму RSA можна зазначити, що більш доцільним для шифрування відеоконтенту є алгоритм AES.

4. ЗАСТОСУВАННЯ ПОЛІАДИЧНИЙ КОДІВ НЕФІКСОВАНОЇ ВАГИ У БАЗИСІ JPEG ДЛЯ ЗМЕНШЕННЯ ОБЧИСЛЮВАЛЬНОГО НАВАНТАЖЕННЯ У ХОДІ ФОРМУВАННЯ ШИФРОГРАМ

4.1 Загальні шляхи зменшення обчислювального навантаження під час утворення шифрограм ключових кадрів відеопотоку

Як попередньо було зазначено у розділі 2, хоча підхід, у рамках якого шифруванню підлягають виключно кадри I-типу, дозволяє суттєвим чином зменшити обсяг обчислень. Водночас, як показує оцінка потенційно можливої кількості інформації, яку належить шифрувати, обчислювальне навантаження для випадку потоку 4K може бути критичним.

Таким чином необхідно розглянути шляхи зменшення об’єму операцій, які мають виконуватися при побудові шифрограм.

При цьому, у загальному випадку завдання зменшення обсягу обчислювальних операцій може вирішуватися за двома напрямками, а саме:

- вибір алгоритму шифрування, який для побудови шифрограми та далі для її відкриття використовує мінімальну кількість обчислювальних операцій;
- зменшення вихідної кількості γ десяткових чисел, які утворюються у наслідок JPEG-кодування на рівні одного блоку $\lambda_{x,y}^{(Q)}$ відеокадру.

Разом з тим, вибір алгоритму шифрування за ознакою простоти реалізації не гарантує, що такому алгоритму відповідатиме необхідний рівень криптографічної стійкості. Тому більш доцільним є зменшення кількості даних, які підлягають шифруванню.

У свою чергу, за умови кодування опорних кадрів відповідно до принципів JPEG, до зменшення величина γ можуть вести такі різнорідні фактори, як:

- особливості змісту кадру;
- обрані опції кодування на етапах, що передують стисненню без втрат.

Зрозуміло, що перший з наведених факторів цілком залежить виключно від оброблюваного відеоряду.

При цьому, умова $\gamma \to \min$ виконується тоді, коли:

- відеоряд являє собою продукт комп'ютерного моделювання та не має у своєму складі (або має у обмеженому об'ємі) об'єктів природного походження;
- відео сцени містять суцільні ділянки одного тону, або з незначними їх градаціями.

Ураховуючи те, що у реальних умовах відеоряд характеризується високим рівнем гетерогенності, у загальному випадку даний фактор не може вважатися вирішальним глобально.

Разом з тим, суттєво зменшити величину γ можливо шляхом збільшення рівня квантування та зміни порогу округлення квантованих компонент $\nu_{i,j}^{(\lambda)}$.

При цьому, збільшиться кількість нульових компонент у високочастотній та середньочастотній зонах блоку $\lambda_{x,y}^{(Q)}$ у його спектральному описі.

У той же час, можливість зменшення величини γ обмежується тим, що її зростання зумовлює порушення рівня R якості кодованого кадру, тобто, справедливим є наступне співвідношення:

$$\phi \uparrow \to \gamma \downarrow | R \downarrow, \tag{4.1}$$

.

де ϕ - крок квантування.

Вираз (4.1) також є справедливим для випадку зміни порогу округлення величин квантованих компонент $\nu_{i,j}^{(\lambda)}$.

Отже, можливість забезпечити зменшення величини γ є, по-перше, досить обмеженою, що пов'язано з імовірним падінням якості кадру.

По-друге, потенційна можливість зменшення γ також залежить від особливостей змісту кодованого кадру та, у окремих випадках, практично є неможливою без суттєвого падіння рівня R якості кадру.

При цьому, можливим шляхом до зменшення обсягу обчислень у ході шифрування є виокремлення та подальше шифрування найбільш критичних складових з множини γ.

Тоді залишиться необхідність формування шифрограми відносно наступної множини:

$$\gamma' = \gamma - \Delta\gamma, \tag{4.2}$$

де γ' - зменшена множина десяткових величин, які підлягають шифруванню;

$\Delta\gamma$ - сукупність некритичних складових у межах початкової множини γ.

Єдиним, але критичним обмеженням щодо реалізації даного підходу у рамках JPEG є те, що чіткого механізму або навіть ознаки виокремлення першочергово важливих складових з масиву квантованих компонент $\nu_{i,j}^{(\lambda)}$ на сьогодні не існує.

За таких умов доцільно розглянути підхід, у рамках якого передбачається модифікація технологічного етапу кодування без втрат, який у класичному JPEG реалізовано на базі методу Хафмана, або з використання арифметичного коду.

Такою модифікацією може бути заміна означених методів кодування. Альтернативною у цьому випадку може бути позиційний код з нефіксованими вагами.

4.2 Обгрунтування доцільності використання коду з нефіксованими вагами на етапі кодування без втрат у схемі JPEG

На величину загальної затримки t, як свідчить вираз (1.1), впливає, окрім інших складових, також час t_{enc} кодування відеоданих, сформованих на рівні джерела.

При цьому наближено можна вважати, що сучасні методи кодування здатні виконувати обробку у реальному часі.

Тому, навіть для випадку інтерактивного відео, першочергово критичним з точки зору формування загальної затримки t вважається не час t_{enc} кодування, а час t_{sh}, який витрачається на побудову шифрограми, як показано виразами (1.4) та (1.5).

Це зумовлено тим, що величина t_{sh} може бути суттєво більшою, ніж t_{enc}.

Разом з тим, в існуючому технологічному базисі спостерігається залежність між роздільною здатністю HW кадру та величиною t_{enc} затримки, зумовленою процесом кодування, як показано рис.4.1.

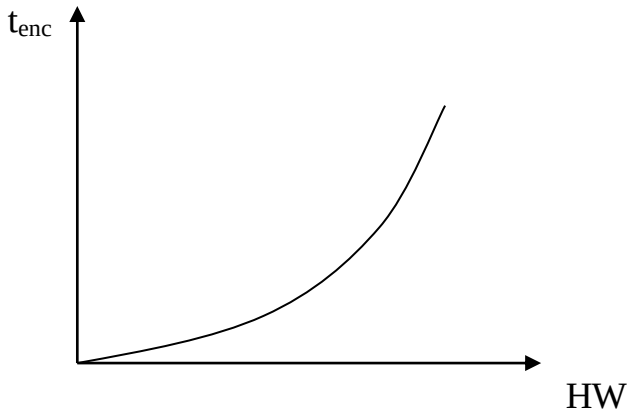

Рисунок 4.1 – Характер залежності між роздільною здатністю відеопотоку та часом затримки на його кодування

Залежність, проілюстрована рис.4.1, найбільш характерна для відео форматів UHD.

Таким чином, зменшення складової t_{enc} є також важливим завданням поряд з забезпеченням мінімізації часу t_{sh}.

У зв'язку з зазначеним, необхідно урахувати таку властивість поліадичного коду з нефіксованими вагами, як відсутність необхідності попередньо створювати контекстну модель кодованих даних, як це зараз є необхідним для методу Хафмана та арифметичного кодування.

Тобто, класичний JPEG у ході кодування без втрат вимагає попереднього виконання ряду додаткових етапів, зокрема:

- лінеаризації компонент;
- кодування довжин серій (RLE).

На відміну від цього, поліадичний код не потребує побудови контекстної моделі, відтак етапи лінеаризації та RLE у випадку його застосування будуть відсутніми. Тим самим створюються умови для зменшення часу t_{enc}.

Даний факт може розцінюватися як один з чинників, що обґрунтовують доцільність використання поліадичного коду з нефіксованими вагами у якості альтернативи існуючим ймовірнісно-статистичним методам.

Разом з тим, архітектура кодів даного типу передбачає, що кодограма на їх базі у будь-якому випадку являє собою згортку множини γ, яка від початку дорівнює 64.

Для того, щоб обґрунтувати можливість поліадичних кодів з нефіксованими вагами до скорочення множини γ, далі розглянемо принцип їх побудови.

4.3 Принцип побудови поліадичного коду з нефіксованими вагами

Вихідними даними для побудови поліадичного коду з нефіксованими вагами на етапі кодування без втрат є матриця квантованих компонент $\nu_{i,j}^{(\lambda)}$, що утворюють блок $\lambda_{x,y}^{(Q)}$ кадру у спектральному представленні.

При цьому, у складі блоку $\lambda_{x,y}^{(Q)}$ формується n кодових конструкцій окремих поліадичних чисел Θ_i.

Такі поліадичні числа можуть утворюватися як на рівні рядків блоку $\lambda_{x,y}^{(Q)}$, так і на рівні стовпців.

Тобто, кодове представлення $E(\lambda_{x,y}^{(Q)})$ блоку відеокадру у спектральному просторі розглядається як поєднання кодових конструкцій поліадичних чисел, а саме:

$$E(\lambda_{x,y}^{(Q)}) = \Theta_1 \,\&\, \Theta_2 \,\&\, \ldots \,\&\, \Theta_i \ldots \,\&\, \Theta_n . \qquad (4.3)$$

Далі розглянемо випадок формування поліадичних чисел за стовпцями блоку $\lambda_{x,y}^{(Q)}$ у його спектральному представленні.

У свою чергу, поліадичні числа, з 1-го по n-те, формуються незалежно одне від одного на базі величини $\nu_{i,j}^{(\lambda)}$ компонент відповідних стовпців блоку $\lambda_{x,y}^{(Q)}$ (рис.4.2).

Принцип поліадичного кодування з нефіксованими вагами дозволяє однаково ефективно утворювати кодограми на базі блоків $\lambda_{x,y}^{(Q)}$ довільного розміру, що дозволяє його застосовувати для усіх без обмеження технологій сімейства H.26* - H.264/AVC, H.265/HEVC, H.266 та будь-яких інших, де

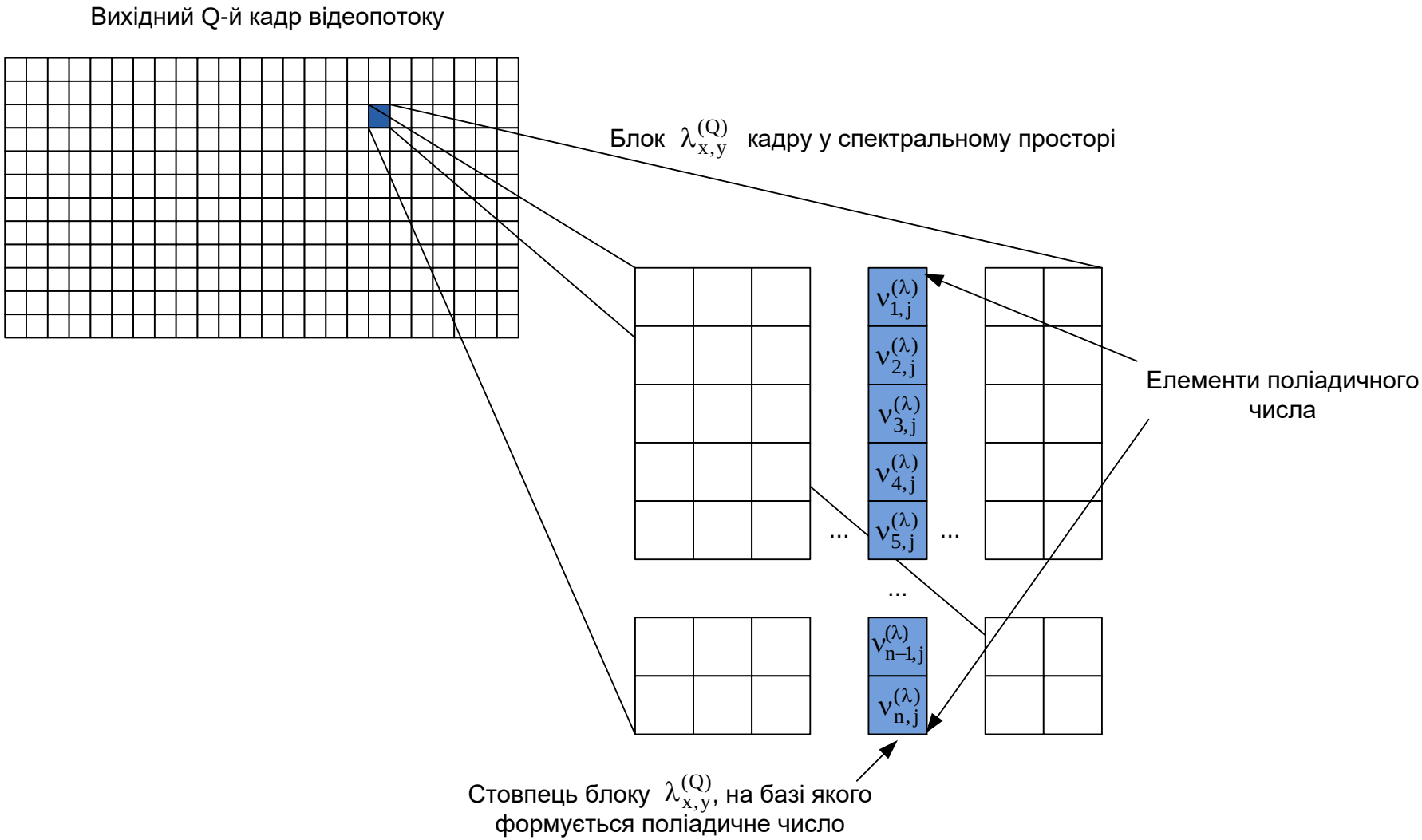

Рисунок 4.2 – Локалізація елементів поліадичних чисел за стовпцями блоку для побудови його кодового опису

передбачається обробка блоків з пропорцією розмірності nxn, її зміною або фіксованою величиною у межах кадру.

4.3.1 Утворення поліадичних чисел

У рамках поліадичного простору компоненти $\nu_{i,j}^{(\lambda)}$ мають назву елементів поліадичних чисел.

Перший етап формування поліадичного числа Θ_i кодограми $E(\lambda_{x,y}^{(Q)})$ блоку кадру передбачає обчислення основ ϑ_i. При цьому, основи ϑ_i на випадок формування поліадичних чисел за стовпцями блоку $\lambda_{x,y}^{(Q)}$ обчислюються на рівні рядків, та визначаються за найбільшою компонентою у рядку згідно з виразом:

$$\vartheta_i = \max(\nu_{i,j}^{(\lambda)}) + 1, \tag{4.4}$$

де доданок 1 необхідний для того, щоб уникнути ситуації невизначеності на випадок формування поточного рядку виключно з нульових елементів.

Далі, після того, як для усіх n рядків основи ϑ_i розраховано, на їх базі будується множина $\{\Psi_j\}$ вагових коефіцієнтів елементів поліадичних чисел, що визначаються як:

$$\Psi_j = \prod_{k=j+1}^{n} \vartheta_{i,k} \tag{4.5}$$

Як показує аналіз виразу (4.5), обчислення коефіцієнту Ψ_i, що відноситься до елементу $\nu_{i,j}^{(\lambda)}$, який має j-й індекс у межах стовпця, зводиться до знаходження добутку основ рядків з (j+1)-го по n-й включно.

У підсумку, поліадичне число Θ_i після розрахунку усієї множини $\{\Psi_i\}$ вагових коефіцієнтів обчислюється як сума добутків елементів позиційних чисел на відповідні величини вагових коефіцієнтів згідно з виразом [22]:

$$\Theta_i = \sum_{j=1}^{n} [\nu_{i,j}^{(\lambda)} \times \Psi_j], \qquad (4.6)$$

У результаті виконання операцій, зазначених виразами (4.4)-(4.6) на базі одного стовпця утворюється поліадичне число Θ_i, що являє собою десяткову величину.

4.3.2 Формування кодограми блоку кадру

Як показує вираз (4.3), кодограма $E(\lambda_{x,y}^{(Q)})$ блоку кадру утворюється поєднанням поліадичних чисел.

Тобто, являє собою масив з n десяткових чисел за кількістю стовпців блоку $\lambda_{x,y}^{(Q)}$. Звідси виходить, що у будь-якому випадку буде виконуватися наступна рівність:

$$n = \gamma, \qquad (4.7)$$

тобто, застосування поліадичного кодування з нефіксованими вагами для стиснення без втрат блоку, відносно якого далі буде виконано процедуру шифрування, дозволяє суттєво скоротити величину γ, а саме – у n разів, так як кількість поліадичних чисел Θ_i рівняється кількості стовпців (або рядків).

Розглянемо далі, яку саме структуру мають кодовані у зазначений спосіб дані для того, щоб визначити, яким саме чином буде побудовано процес шифрування.

По-перше, оскільки за виразом (4.3) опис кодованого блоку $\lambda_{x,y}^{(Q)}$ є масивом з n чисел, то на прийомному боці його неможливо буд відновити.

У свою чергу, відновлення компонент $\nu_{i,j}^{(\lambda)}$ блоку $\lambda_{x,y}^{(Q)}$ здійснюється відповідно до принципу, зазначеному наступним виразом:

$$c_{i,j} = \left[\frac{\Theta_i}{\Psi_j}\right] - \left[\frac{\Theta_i}{\vartheta_i \times \Psi_j}\right] \times \vartheta_i, \qquad (4.8)$$

де квадратні дужки вказують на те, що береться до уваги цілочисельна частина відношення.

Тобто, для реконструкції блоку разом з безпосередньо масивом поліадичних чисел необхідно надсилати приймачеві сукупність основ ϑ_i рядків, як основні службові дані кодового опису, використовуючи який можливо відновити кодовані дані.

Таким чином, підсумкова структура кодограми блоку $\lambda_{x,y}^{(Q)}$ кадру буде такою, як показано рис. 4.3.

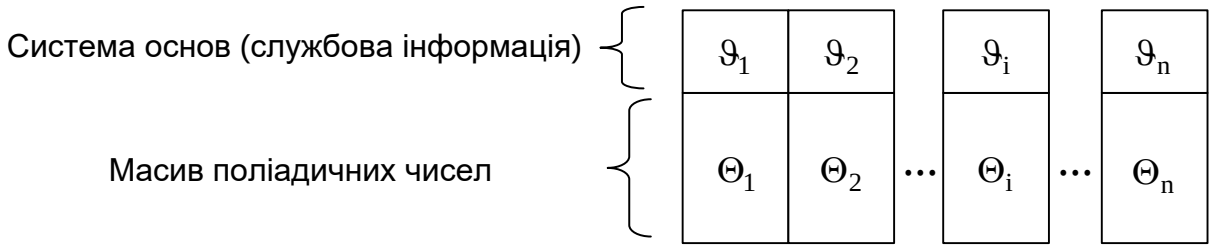

Рисунок 4.3 – Структура кодограми блоку кадру

Отже, на базі аналізу виразів (4.3), (4.8) та рис. 4.3 можна зробити висновок про те, що для повного закриття ключового кадру достатньо шифрувати n основ рядків кодограми блоків як базової та єдиної інформації, на базі якої можливим є відновлення початкової структури блоку.

Разом з тим, для величин основ ϑ_i справедливе наступне співвідношення:

$$\vartheta_i < \Theta_i, \tag{4.9}$$

при цьому значення основ менше від величин поліадичних чисел у середньому на порядок.

Таким чином, це зумовлює доцільність шифрування саме системи основ, а не безпосередньо множини поліадичних чисел.

4.4 Оцінка об’єму даних, які підлягають шифруванню на випадок застосування поліадичних кодів з нефіксованими вагами у базисі JPEG

Раніше, для випадку JPEG-кодування, для найгірших умов приймалося, що $\gamma = 63$.

При цьому, так було зазначено у п.4.3, поліадичний код являє собою згортку вихідної кількості γ десяткових величин. Це спричинює суттєве скорочення величини γ, як зазначає вираз (4.7), а саме – до значення n, хоча при цьому величина поліадичного числа буде значно більшою, ніж значення коефіцієнтів кодограми, утвореної за результатами ймовірнісно-статистичного кодування.

Отже, якщо розглядати випадок, коли $n = 8$ і при цьому $\gamma = n$, тоді для окремого кадру FullHD згідно з виразом (2.3) обробці з використанням алгоритму шифрування підлягатиме $\sigma = 32400 \times 8 = 259200$ біт, (0,0324 Мб).

Водночас, для потоку FullHD при χ=30 і за умови розмірності групи $G = 8$ маємо $\sigma(t) = \sigma \times N_{key} = 259200 \times 3 = 777600$ біт (0,0972 Мб).

Для випадку ж, коли $G = 32$ отримуємо $\sigma(t) = \sigma \times N_{key} = 259200 \times 1 = 259200$ біт, (0,0324 Мб).

У свою чергу, якщо розглядати окремий І-кадр формату 4К, тоді $\sigma = 196608 \times 8 = 1572864$ біт (0,196608 Мб).

Далі оцінимо кількість даних, що будуть шифруватися для випадку 4К потоку з різною розмірністю групи.

Так, при при χ=30 та за умови, що $G = 8$, отримуємо $\sigma(t) = 1572864 \times 3 = 4718592$ біт (0,589824 Мб).

Водночас, для $G = 32$ за аналогічних умов маємо $\sigma(t) = 1572864 \times 1 = 1572864$ біт (0,196608 Мб).

Зведені відомості щодо кількості даних, які необхідно шифрувати для відеопотоків форматів FullHD та UHD за різних умов та при використанні ймовірнісно-статистичного кодування та поліадичних кодів з нефіксованими вагами ілюструє табл. 4.1

Таблиця 4.1 - Зведені відомості щодо кількості даних, які необхідно шифрувати для відео потоків при використанні ймовірнісно-статистичного кодування та поліадичних кодів з нефіксованими вагами

Роздільна здатність	Ймовірнісно-статистичне кодування			Поліадичний код з нефіксованими вагами		
	Окремий ключовий кадр	G = 8	G = 32	Окремий ключовий кадр	G = 8	G = 32
2K	0,25515 Мб	0,76545 Мб	0,25515 Мб	0,0324 Мб	0,0972 Мб	0,0324 Мб
4K	1,548288 Мб	4,644864 Мб	1,548288 Мб	0,196608 Мб	0,589824 Мб	0,196608 Мб

З аналізу табл. 4.1 можна зробити висновок, що загальний об’єм даних, які підлягають шифруванню за умови, що попередньо їх було кодовано на базі модифікованого JPEG, майже у 7,9 разів менший порівняно з випадком кодування у класичному JPEG.

Таким чином, створюються умови для зменшення обсягу обчислень при формуванні шифрограм. Тобто при цьому зменшується обчислювальне навантаження на систему та скорочується час формування шифрограм, що дозволяє мінімізувати затримку, внесену у ході обробки відеопотоку.

ВИСНОВКИ

Згідно з вимогами технічного завдання, під час виконання кваліфікаційної роботи виконано дослідження підходів щодо забезпечення ефективної побудови шифрограм відеоінформаційних потоків реального часу.

Так, було обґрунтовано, що забезпечення ефективного шифрування інтерактивного відеоконтенту вимагає:

- мінімізації часу, що витрачається на шифрування;
- зменшення часу, який займає процедура кодоутворення.

Це пояснюється тим, що, по-перше, на відміну від відеосервісів потокового типу, інтерактивне відео вимагає за досить лімітований час виконання процедур утворення первинних відеоданих, формування кодових конструкцій відео та наступного його шифрування.

При цьому, алгоритм шифрування являє собою додатковий блок у конвеєрі перетворень відео від його генерації і то надсилання пакетованих даних у мережу. Це і зумовлює існування вимог відносно мінімізації часу шифрування.

Показано, що в існуючих умовах більш надійним є надсилання одержувачам кодовано них відеоданих, замість їх трансляції захищеним каналом у відкритому вигляді.

Разом з тим, ураховуючи значні інформаційні інтенсивності відео, та його домінуючий характер серед інших типів трафіку, кодування усього потоку являє собою складну задачу, що вимагає підвищених обчислювальних потужностей як передавача, так і приймача, та на рівні більшості Існуючих сьогодні клієнтських терміналів не може бути гарантовано реалізованим.

Відтак, шляхом дослідження принципів побудови відеопотоку на базі платформи MPEG визначено, що для повного його закриття від несанкціонованого доступу достатньо шифрувати виключно ключові кадри. При цьому, оскільки загальний обсяг ключових кадрів потоку є суттєво нижчим, порівняно з масою передбачених кадрів, створюються умови для скорочення часу шифрування.

У рамках дослідження підходів до подальшого скорочення часу шифрування виконано оцінку обсягу даних, які підлягають шифруванню, в умовах, коли кодування ключових кадрів виконується за загальними принципами JPEG, характерними для MPEG-платформи. При цьому

виявлено, що найбільший обсяг даних, що мають шифруватися щосекунди, відповідає випадку, коли розмірність групи у рамках відеопотоку є мінімальною та дорівнює 8. У свою чергу, обсяг даних, які необхідно шифрувати за секудну за умов групи максимального розміру (32 кадри) є найменшим, та відповідає об'єму одного ключового кадру.

У підсумку виявлено, що для випадку потоку відео 4К та групи мінімального розміру обсяг шифрованих даних є критичним. Отже, за цих умов необхідно:

- використовувати більш швидкий алгоритм шифрування:

- розглянути підхід, у рамках з відеокадру може бути виокремлено найбільш критичні складові, шифрування яких, по-перше дозволить повністю закрити кадр, по-друге – займе значно менше часу на обробку.

При цьому, оскільки для вибору алгоритму шифрування орієнтуватися виключно на його швидкодію є недоцільним, було розглянуто другий шлях рішення завдання підвищення швидкості шифрування, а саме – зменшення кількості вихідних даних, які необхідно шифрувати на базі того чи іншого криптографічного алгоритму для повного закриття кадру

Виходячи з цього, досліджено підхід, у рамках якого здійснюється заміна класичного алгоритму ймовірнісно-статистичного кодування зі складу JPEG поліадичним кодом з нефіксованими вагами.

Такий підхід передбачає формування на базі вихідної кількості компонент блоку кадру після його квантизації у спектральному просторі ряду поліадичних чисел, кількість яких завжди є фіксованою та дорівнює кількості стовпців (рядків) блоку.

Тобто,перевагами такого підходу є:

- детермінована кількість математичних операцій у ході кодоутворення;
- апріорі відома і постійна кількість підсумкових поліадичних чисел, що утворюються за результатами кодування;
- той факт, що кількість сформованих поліадичних чисел завжди є меншою у n разів по відношенню з кількістю nxn компонент після процедури квантування.

Отже, відповідно до даного підходу шифруванню підлягає службова інформація – система основ, яка являє собою n десяткових чисел, які використовуються для відновлення кодограми на прийомному боці.

Показано, що за даних умов для найгіршого випадку кількість біт, які необхідно шифрувати у складі ключового кадру, майже у 7,9 разів є меншою, ніж для випадку, коли використовується ймовірнісно-статистичний підхід на етапі кодування без втрат у JPEG. Відповідно, це суттєво заощаджує час для формування шифрограм та створює умови, у яких може гарантуватися затримка на обробку та надсилання шифрованих даних відповідно до вимог QoS.

У ході дослідження алгоритмів шифрування даних виявлено, що криптографічна система AES є доцільною для застосування як базовий інструмент шифрування відео. Це зумовлюється наступним:

- архітектура AES є максимально адаптованою до обробки кодованого відеоконтенту, що дозволяє заощаджувати час на його виконання;
- алгоритм не потребує надмірних обчислювальних потужностей та може бути реалізованим практично на базі будь-якого клієнтського терміналу.

Таким чином, усі пункти технічного завдання опрацьовано та виконано у повному обсязі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Утечка данных. Статистика за 2021 год. [Електронний ресурс] – Режим доступу: https://www.tadviser.ru/index.php/%D0%A1%D1%82%D0%B0%D1%82%D1%8C%D1%8F:%D0%A3%D1%82%D0%B5%D1%87%D0%BA%D0%B8_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85
2. VNI Forecast Highlights Tool [Електронний ресурс] – Режим доступу: https://www.cisco.com/c/m/en_us/solutions/service-provider/vni-forecast-highlights.html .
3. Digital in 2020: World's internet users pass the 4 billion mark [Електронний ресурс] – Режим доступу: https://wearesocial.com/blog/2020/01/global-digital-report-2020
4. Обеспечение безопасности корпоративных систем: какими уязвимостями пользуются злоумышленники [Електронний ресурс] – Режим доступу: https://www.ptsecurity.com/ru-ru/research/analytics/corporate-vulnerabilities-2019/
5. Критические угрозы безопасности. Компания Cisco представила результаты отчетов в сфере защиты данных, опубликованных в 2019 году - Connect-WIT [Електронний ресурс] – Режим доступу: https://www.connect-wit.ru/kriticheskie-ugrozy-bezopasnosti-kompaniya-cisco-predstavila-rezultaty-otchetov-v-sfere-zashhity-dannyh-opublikovannyh-v-2019-godu.html
6. Шнайер Б.. Прикладная криптография. Протоколы, алгоритмы и исходный код на С. – М.: Вильямс, 2016. – 1024 с.
7. Чемпен Н., Чемпен Д. Цифровые технологии мультимедиа. – М.: Вильямс, 2006. – 624 с.
8. Качество обслуживания в операторских сетях [Електронний ресурс] – Режим доступу: https://www.opennet.ru/docs/RUS/qos_oper/
9. Культура мультимедиа : учебное пособие / О.В. Шлыкова. - М. : ФАИР-пресс, 2004.
10. Диогенес Ю., Озкайя Э. Кибербезопасность. Стратегии атак и обороны. – М.: ДМК Пресс, 2016. – 326 с.

11. Как устроено сквозное шифрование в Zoom | Блог Касперского [Електронний ресурс] – Режим доступу: https://www.kaspersky.ru/blog/rsa2021-zoom-end-to-end-encryption/31021/
12. Zoom isn’t actuall and-to-end encrypted [Електронний ресурс] – Режим доступу: https://www.theverge.com/2020/3/31/21201234/zoom-end-to-end-encryption-video-chats-meetings
13. MITM-атака (атака «человек посередине») | Блог "Касперского" [Електронний ресурс] – Режим доступу: https://encyclopedia.kaspersky.ru/glossary/man-in-the-middle-attack/
14. Гонсалес Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. – М. : Техносфера, 2005. – 1073 с.
15. Ватолин Д. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео / Д. Ватолин, А. Ратушняк, М. Смирнов, В. Юкин. – М. : ДИАЛОГ – МИФИ, 2003. – 384 с.
16. Ричардсон Ян. H.264 and MPEG-4 Video Compression: Video Coding for Next-Generation Multimedia / Ян Ричардсон. – Город. : Издательство, 2005. – 368 с.
17. Shi, Yun Q. Image and video compression for multimedia engineering: fundamentals, algorithms, and standards / Yun Q Shi, Huifang Sun.
18. Айфичер Эммануил С. Цифровая обработка сигналов: практический поход / Эммануил С. Айфичер, Барри У. Джервис. – 2-е изд. – М. : Вильямс, 2008. – 992 с.
19. Сэломон Д. Сжатие данных, изображений и звука / Д. Сэломон. – М.: Техносфера, 2004. – 368 с.
20. Красильников Н.Н. Цифровая обработка изображений. – М.: Вузовская книга, 2011. – 320 с.
21. Cryptology ePrint Archive [Електронний ресурс] – Режим доступу: https://eprint.iacr.org/
22. Авдошин С., Набебин А. Дискретная математика. Модулярная алгебра, криптография, кодирование. – М.: ДМК Пресс, 2016. – 352 с.
23. Крэндалл Р., Померанс К. Простые числа. Криптографические и вычислительные аспекты. – М.: Либроком, 2011. – 664 с.
24. ROCA: Vulnerable RSA generation (CVE-2017-15361) | Crocs Wiki [Електронний ресурс] – Режим доступу: https://crocs.fi.muni.cz/public/papers/rsa_ccs17

25. Maitra S. Revisiting Wiener's Attack - New Weak Keys in RSA. — Indian Statistical Institute. — 2008.
26. Ferguson N., Schroeppel R. and Whiting D. A simple algebraic representation of Rijndael. Selected Areas in Cryptography, Proc. SAC 2001, Lecture Notes in Computer Science #2259. — Springer Verlag, 2001. — P. 103—111.
27. Rijndael block cipher [Електронний ресурс] – Режим доступу: https://csrc.nist.gov/csrc/media/projects/cryptographic-standards-and-guidelines/documents/aes-development/rijndael-ammended.pdf
28. Schindler W., Koeune F., Quisquater J-J. Improving Divide and Conquer Attacks against Cryptosystems by Better Error Detection/Correction Strategies. Proc. of 8th IMA International Conference on Cryptography and Coding :— 2001. — P. 245—267. — doi:10.1.1.13.5175