**МІНІСТЕРСТВО ОСВІТИ І НАУКИ,
МОЛОДІ ТА СПОРТУ УКРАЇНИ**

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ

Кузнецов О. О.
Євсеєв С. П.
Король О. Г.

СТЕГАНОГРАФІЯ

Навчальний посібник

Харків. Вид. ХНЕУ, 2011

УДК 004.67(075.8)
ББК 32.973я73
К89

Рецензенти: докт. техн. наук, професор, зав. кафедри систем захисту інформації, директор Навчально-наукового інституту захисту інформації *Хорошко В. О.*; докт. техн. наук, провідний науковий співробітник наукового центру Харківського університету Повітряних Сил ім. Івана Кожедуба *Бараннік В. В.*; докт. техн. наук, професор кафедри телекомунікаційних систем Харківського національного університету радіоелектроніки *Лемешко О. В.*

Рекомендовано до видання рішенням вченої ради Харківського національного економічного університету.
Протокол № 12 від 30.06.2009 р.

Авторський колектив: Кузнецов О. О., докт. техн. наук, професор – вступ, розділи 1, 3, 4, додатки; Євсеєв С. П., канд. техн. наук, ст. наук. співробітник – розділ 5; Король О. Г., викладач – розділ 2.

Кузнецов О. О.
К89 Стеганографія : навчальний посібник / О. О. Кузнецов, С. П. Євсеєв, О. Г. Король. – Х. : Вид. ХНЕУ, 2011. – 232 с. (Укр. мов.)

Розглянуто основні напрями стеганографії, математичну модель та структурну схему стеганографічної системи, класифікацію систем цифрової стеганографії та їх використання, атаки на стеганосистеми і протидію їм.

Рекомендовано для аспірантів і студентів спеціалізацій "Інформаційні управляючі системи та технології" та "Комп'ютерний еколого-економічний моніторинг" усіх форм навчання.

ISBN

УДК 004.67(075.8)
ББК 32.973я73

Вступ

Завдання захисту інформації від несанкціонованого доступу вирішувалося в усі часи історії людства. Уже в стародавньому світі виділилося два основні напрями рішення цього завдання, що існують і до сьогоднішнього дня: криптографія й стеганографія. Метою криптографії є приховання вмісту повідомлень за рахунок їх шифрування. На відміну від цього, при стеганографії приховується сам факт існування таємного повідомлення.

Розвиток засобів обчислювальної техніки в останнє десятиліття в новий поштовх для розвитку комп'ютерної стеганографії. З'явилося багато нових галузей застосування. Існують два основні напрями в комп'ютерній стеганографії: пов'язаний із цифровою обробкою сигналів і непов'язаний. Останній напрям має обмежене застосування у зв'язку з відносною легкістю розкриття й/або знищення прихованої інформації. Більшість поточних досліджень у множині стеганографії так чи інакше пов'язані із цифровою обробкою сигналів. Це дозволяє говорити про цифрову стеганографію. Саме цій науці і присвячений даний навчальний посібник.

Серед завдань навчальної дисципліни "Технології захисту інфомації" важливим також є вивчення основ стеганографічного захисту інформації, методів та обчислювальних алгоритмів стеганографічного перетворення, дослідження відповідних атак на стеганосистеми та вивчення методів протидії. Саме для вивчення відповідної теми навчальної дисципліни і призначений даний навчальний посібник, який висвітлює методи та принципи побудови та застосування стеганографічних систем і протоколів, методи, алгоритми та засоби аналізу стеганографічної стійкості стеганографічних систем та безпечності стеганографічних протоколів. Вивчення відповідної теми має основоположне значення, оскільки стеганографічні системи та протоколи є, після криптографії, основою захисту інформації та ресурсів з визначеним рівнем безпеки.

У даному посібнику наведені основні напрями стеганографії, математична модель та структурна схема стеганографічної системи, класифікація систем цифрової стеганографії та їх використання, методи

стеганографічного захисту інформації, розглянуті атаки на стеганосистеми та протидія їм. За результатами вивчення відповідної теми навчальної дисципліни "Технології захисту інформації студенти повинні:

знати: основні методи, системи та засоби забезпечення стеганографічного захисту інформації, аналізу стійкості стеганосистем та безпечності стеганопротоколів; порядок та умови застосування ключових стеганосистем, а також методи та засоби управління ключовими даними; основні тенденції та напрями розвитку теорії та практики стеганосистем та стеганопротоколів, прогнозування їх можливостей та можливостей стеганоаналітиків (порушників); функціональні можливості та порядок застосування сучасних пакетів програмної реалізації стеганографічних перетворень та стеганографічних бібліотек; порядок оцінки якісних показників стеганосистем та стеганопротоколів;

вміти: розробляти вимоги та обирати для застосування стеганосистеми та стеганопротоколи, що мінімізують впливи порушників; вибирати та застосовувати критерії та показники оцінки стійкості стеганосистем та безпечності стеганопротоколів; обґрунтовувати вимоги до ключових систем та управління ключовими даними стеганосистем, здійснювати аналіз їх властивостей; проводити аналіз та синтез стеганопротоколів за критерієм безпечності, порівнювати їх з використанням умовних та безумовних критеріїв; застосовувати стандартні пакети при розв’язанні прикладних задач моделювання стеганосистем, ключових систем і стеганографічних протоколів.

Таким чином, у результаті вивчення відповідної теми навчальної дисципліни, для викладення якої і призначено даний навчальний посібник, студенти повинні засвоїти методи та принципи побудови, реалізації та застосування стеганографічних систем та протоколів, вміти застосовувати методи, алгоритми та засоби оцінки стеганостійкості та інших якісних показників стеганосистем та стеганографічних протоколів. При вивченні стеганографічних протоколів студенти повинні вміти обґрунтовувати вимоги, розв’язувати завдання аналізу та синтезу стеганографічних протоколів, складати програмні моделі та здійснювати моделювання стеганосистем, практично реалізовувати обчислювальні алгоритми стеганографічного захисту інформації.

Розділ 1. Вступ до стеганографії

1.1. Предмет стеганографії, основні терміни та визначення. Історичні приклади стеганосистем

Інформація є одним з найцінніших предметів сучасного життя. Одержання доступу до неї з появою глобальних комп'ютерних мереж стало неймовірно простим. У той же час легкість і швидкість такого доступу значно підвищили і загрозу порушення безпеки даних при відсутності засобів щодо їх захисту, а саме загрозу неавторизованого доступу до інформації.

Завдання надійного захисту авторських прав, прав інтелектуальної власності або конфіденційних даних (які в більшості випадків мають цифрової формат) від несанкціонованого доступу є однією з найдавніших і невирішених на сьогодні проблем. У зв'язку з інтенсивним розвитком і поширенням технологій, які дозволяють за допомогою комп'ютера інтегрувати, обробляти та синхронно відтворювати різні типи сигналів (так звані мультимедійні технології), питання захисту інформації, представленої в цифровому вигляді, є надзвичайно актуальним.

Переваги подання та передачі даних у цифровому вигляді (легкість відновлення, висока потенційна завадостійкість, перспективи використання універсальних апаратних і програмних рішень) можуть бути перекреслені з легкістю, з якою можливі їх викрадення та модифікація. Тому в усьому світі назріло питання розробки методів (засобів) захисту інформації організаційного, методологічного й технічного характеру, серед них – методи криптографії та стеганографії.

Криптографічний захист інформації – система зміни останньої з метою зробити її незрозумілою для непосвячених, приховання змісту повідомлень за рахунок їх шифрування). Цей захист не вирішує згадану вище проблему повністю, оскільки наявність шифрованого повідомлення привертає увагу, і зловмисник, заволодівши криптографічно захищеним файлом, відразу розуміє про розміщення в ньому секретної інформації й переводить всю сумарну міць своєї комп'ютерної мережі на дешифрування даних.

Приховання ж самого факту існування секретних даних при їх передачі, зберіганні або обробці є завданням стеганографії – науки, що вивчає способи та методи приховання конфіденційних відомостей.

Завдання виявлення інформації при цьому відступає на другий план і вирішується в більшості випадків стандартними криптографічними методами.

Інакше кажучи, під *приховуванням існування інформації* мається на увазі не тільки неможливість виявлення в перехопленому повідомленні наявності іншого (прихованого) повідомлення, але й взагалі унеможливлення виникнення будь-яких підозр на цей рахунок, оскільки в останньому випадку проблема інформаційної безпеки вертається до стійкості криптографічного коду. Таким чином, займаючи свою нішу в забезпеченні безпеки, стеганографія не заміняє, а доповнює криптографію.

Стеганографування здійснюється різними способами. Загальною ж рисою таких способів є те, що приховуване повідомлення вбудовується в об'єкт, що не привертає увагу і потім відкрито транспортується (пересилається) адресатові. Історично напрям стеганографічного приховання інформації був першим, але згодом багато в чому був витиснутий криптографією. Інтерес до стеганографії відродився в останнє десятиліття і був викликаний широким поширенням технологій мультимедіа, що цілком закономірно, беручи до уваги зазначені вище проблеми, пов'язані із захистом інформації. Не менш важливим стала поява нових типів каналів передачі інформації, що в сукупності з першим фактором дало новий імпульс розвитку та удосконаленню стеганографії, сприяло виникненню нових стеганографічних методів, в основу яких були покладені особливості подання інформації в комп'ютерних файлах, обчислювальних мережах і т. д. Це, у свою чергу, дає можливість говорити про становлення нового напряму у сфері захисту інформації – комп'ютерної стеганографії (КС) [2; 5;11; 35].

З 1996 р. проводяться міжнародні симпозіуми із проблем прихування даних (Information Workshop on Information Hiding). Перша конференція, присвячена стеганографії, відбулася в липні 2002 р. На сьогоднішній день стеганографія є наукою, що швидко і динамічно розвивається, використовуючи при цьому методи й досягнення криптографії, цифрової обробки сигналів, теорії зв'язку та інформації.

Методи стеганографії дозволяють не тільки приховано передавати дані (так звана класична стеганографія), але й успішно вирішувати завдання завадостійкої автентифікації, захисту інформації від несанкціонованого копіювання, відстеження поширення інформації мережами зв'язку, пошуку інформації в мультимедійних базах даних і т. д. Ці обставини

дозволяють у рамках традиційно існуючих інформаційних потоків або інформаційного середовища вирішувати деякі важливі питання захисту інформації ряду прикладних галузей.

Існує два ключових напрями використання КС: пов'язаний із цифровою обробкою сигналів (ЦОС) і непов'язаний. У першому випадку секретні повідомлення вбудовуються в цифрові дані, які, як правило, мають аналогову природу (мова, зображення, аудіо- і відеозаписи). У другому – конфіденційна інформація розміщається в заголовках файлів або пакетів даних (цей напрям не знайшов широкого застосування через відносну легкість витягання та/або знищення прихованої інформації). Переважна більшість поточних досліджень у сфері стеганографії так чи інакше пов'язана саме із ЦОС, що дозволяє говорити про цифрову стеганографію (ЦС) [5; 35].

Можна виділити принаймні дві причини популярності в наш час досліджень у сфері стеганографії: обмеження на використання криптографічних засобів у ряді країн світу та виникнення проблеми захисту прав власності на інформацію, представлену в цифровому вигляді.

Перша причина викликала велику кількість досліджень у стилі класичної стеганографії, тобто приховання властне факту передачі, а друга – не менш численні роботи у сфері так званих цифрових водяних знаків (ЦВДЗн) – спеціальних міток, приховано убудованих у зображення (або інші цифрові дані) для того, щоб мати можливість контролювати це використання.

Приховання інформації тільки на основі факту невідомості зловмисникові методу або методів, закладених в основу приховання, на сьогоднішній день є малоефективним. Ще в 1883 р. фламандський криптограф А. Керхгофс (A. Kerckhoffs) указував на той факт, що система захисту інформації повинна виконувати покладені на неї функції навіть при повній інформованості противника про її структуру та алгоритм функціонування [36].

Вся таємність системи захисту переданих повідомлень повинна втримуватися в ключі – фрагменті інформації, попередньо, як правило, розділеному між адресатами. Незважаючи на те, що цей принцип відомий уже більше 100 років, дотепер існують розробки, які ними зневажають. Очевидно, що вони не можуть використовуватися із серйозною метою. В основі багатьох підходів до рішення завдань стеганографії лежить

загальна із криптографією методична база, яку заклав ще в середині минулого століття К. Шеннон (C. E. Shannon) [19; 58]. Однак і дотепер теоретичні основи стеганографії залишаються практично непрорубленими.

Беручи до уваги вищесказане, можна зробити висновок про те, що на сьогоднішній день існує актуальна науково-технічна проблема вдосконалення алгоритмів і методів проведення стеганографічного приховання конфіденційних даних або захисту авторських прав на певну інформацію.

Сьогодні немає недоліку в стеганографічних програмах як початкового, так і професійного рівня (S-Tools, Steganos Security Suite, bmpPacker та ін.). Однак захищеність їх коду (особливо це стосується програм професійного рівня) не дозволяє простежити методи, покладені в основу алгоритмів їх дії. Розміщені ж в Internet-ресурсах численні тексти програм через свою низьку інформативність мало чим допомагають, тому що компіляція запропонованих текстів виконується програмою так, що її алгоритм вкрай важко простежити, оскільки остання видає вже готовий результат – заповнений стеганоконтейнер, і практично не існує можливості заздалегідь установити достатність рівня приховання конфіденційної інформації в цьому контейнері.

Таким чином, зовсім очевидна недостача саме програм початкового рівня, які б наочно, крок за кроком демонстрували весь процес стеганографічного перетворення, що можна було б використовувати в навчальному процесі при підготовці фахівців у сфері захисту інформації.

1.2. Галузі застосування стеганографії. Практичні аспекти побудови стеганосистем

Цифрова стеганографія як наука народилася буквально в останні роки. На думку авторів, вона містить у собі такі напрями:

вбудовування інформації з метою її прихованої передачі;

вбудовування цифрових водяних знаків (ЦВДЗн) (watermarking);

вбудовування ідентифікаційних номерів (fingerprinting);

вбудовування заголовків (captioning).

ЦВДЗн можуть застосовуватися, в основному, для захисту від копіювання та несанкціонованого використання. У зв’язку з бурхливим розвитком технологій мультимедіа гостро встало питання захисту авторських прав і інтелектуальної власності, представленої в цифровому

вигляді. Прикладами можуть бути фотографії, аудіо- та відеозаписи та ін. Переваги, які дають подання та передача повідомлень у цифровому вигляді, можуть виявитися перекресленими з легкістю, з якою можливі їх викрадення або модифікація. Тому розробляються різні засоби захисту інформації, організаційного та технічного характеру. Один з найбільш ефективних технічних засобів захисту мультимедійної інформації полягає у вбудовуванні в захисті об'єкта невидимих міток – ЦВДЗн. Розробки в цій сфері ведуть найбільші фірми в усьому світі. Оскільки методи ЦВДЗн почали розроблятися зовсім недавно (першою статтею на цю тему була робота [51]), то тут є багато неясних проблем, що вимагають свого вирішення.

Назву цей метод одержав від усім відомого способу захисту цінних паперів, у тому числі грошей, від підробки. Термін "digital watermarking" був уперше застосований у роботі [51]. На відміну від звичайних водяних знаків ЦВДЗн можуть бути не тільки видимими, але й, як правило, невидимими. Невидимі ЦВДЗн аналізуються спеціальним декодером, що виносить рішення про їх коректність. ЦВДЗн можуть містити деякий автентичний код, інформацію про власника або яку-небудь керуючу інформацію. Найбільш придатними об'єктами захисту за допомогою ЦВДЗн є нерухливі зображення, файли аудіо- й відеоданих.

Технологія вбудовування ідентифікаційних номерів виробників має багато загального з технологією ЦВДЗн. Відмінність полягає в тому, що в першому випадку кожна захищена копія має свій унікальний вбудований номер (звідси й назва – дослівно "відбитки пальців"). Цей ідентифікаційний номер дозволяє виробникові відслідковувати подальшу долю свого дітища: чи не зайнявся хто-небудь із покупців незаконним тиражуванням. Якщо так, то "відбитки пальців" швидко вкажуть на винного.

Вбудовування заголовків (невидиме) може застосовуватися, наприклад, для підпису медичних знімків, нанесення легенди на карту й т. д. Метою є зберігання різнорідно представленої інформації в єдиному цілому. Це, мабуть, єдиний додаток стеганографії, де в явному вигляді відсутній потенційний зловмисник.

Оскільки цифрова стеганографія є молодою наукою, то її термінологія не до кінця сформувалася. Основні поняття стеганографії були погоджені на Першій міжнародній конференції з приховання даних [21]. Проте навіть саме поняття "стеганографія" трактується по-різному. Так, деякі дослідники розуміють під стеганографією тільки приховану

передачу інформації. Інші відносять до стеганографії такі додатки, як наприклад, метеорний радіозв'язок, радіозв'язок із псевдовипадковою перебудовою радіочастоти, широкосмуговий радіозв'язок. Неформальне визначення того, що таке цифрова стеганографія, могло б виглядати в такий спосіб: "наука про непомітне і надійне приховання одних бітових послідовностей в інших, що мають аналогову природу". Під це визначення саме підпадають всі чотири вищенаведені напрями приховання даних, а додатка радіозв'язку – немає. Крім того, у визначенні міститься дві головні вимоги до стеганографічному перетворення: непомітність і надійність, або стійкість до різного роду перекручування. Згадування про аналогову природу цифрових даних підкреслює той факт, що вбудовування інформації виконується в оцифровані безперервні сигнали. Таким чином, у рамках цифрової стеганографії не розглядаються питання впровадження даних у заголовки IP-пакетів і файлів різних форматів, у текстові повідомлення.

Яким б різними не були напрями стеганографії, пропоновані ними вимоги багато в чому збігаються, як це буде показано далі. Найбільш істотна відмінність постановки завдання прихованої передачі даних від постановки завдання вбудовування ЦВДЗн полягає в тому, що в першому випадку зловмисник повинен виявити приховане повідомлення, тоді як у другому випадку про його існування всі знають. Більше того, у зловмисника на законних підставах може бути пристрій виявлення ЦВДЗн (наприклад, у складі DVD-програвача).

Під словом "непомітний" у нашому визначенні цифрової стеганографії мається на увазі обов'язкове включення людини в систему стеганографічної передачі даних. Людина тут може розглядатися як додатковий приймач даних, що висуває до системи передачі досить важко формалізовані вимоги.

Завдання вбудовування і виділення повідомлень із іншої інформації виконує стеганосистема. Стеганосистема складається з таких основних елементів, наведених на рис. 1.1:

прекодер – пристрій, призначений для перетворення приховуваного повідомлення до виду, зручного для вбудовування в сигнал-контейнер;

контейнером називається інформаційна послідовність, у якій ховається повідомлення;

стеганокодер – пристрій, призначений для здійснення вкладення прихованого повідомлення в інші дані з урахуванням їх моделі;

пристрій виділення убудованого повідомлення;

стеганодетектор – пристрій, призначений для визначення наявності стегаповідомлення;

декодер – пристрій, що відновлює приховане повідомлення. Цей вузол може бути відсутнім, це буде пояснено далі.

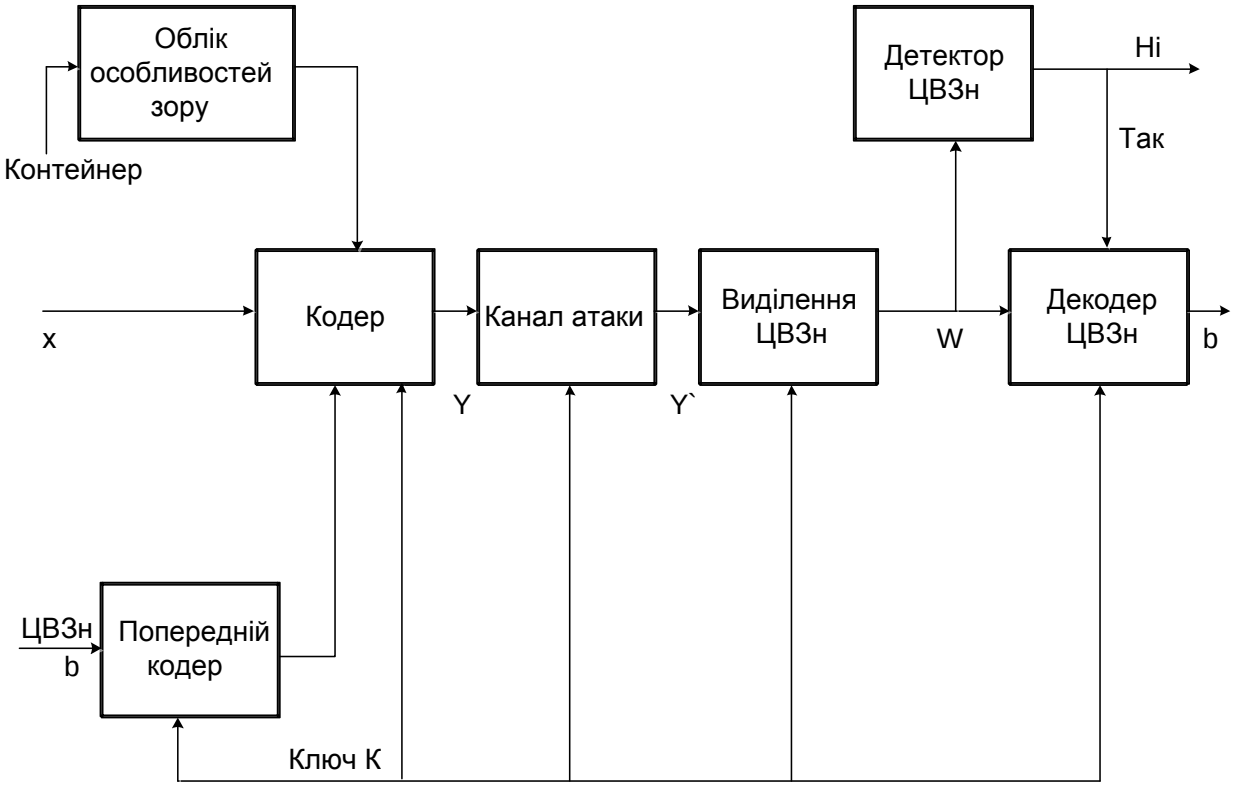


Рис. 1.1. **Структурна схема типічної стеганосистеми ЦВЗн**

На рис. 1.2 наведена класифікація систем цифрової стеганографії. Стеганосистема утворює стеганоканал, по якому передається заповнений контейнер. Цей канал уважається підданим впливам з боку порушників. Згідно з Г. Сіммонсом [63], у стеганографії звичайно розглядається така постановка завдання ("проблема ув’язнених").

Двоє ув’язнених, Аліса і Боб бажають конфіденційно обмінюватися повідомленнями, незважаючи на те, що канал зв’язку між ними контролює охоронець Віллі. Для того щоб таємний обмін повідомленнями був можливий, передбачається, що Аліса і Боб мають деякий відомий обом секретний ключ. Дії Віллі можуть полягати не тільки в спробі виявлення прихованого каналу зв’язку, але й у руйнуванні переданих повідомлень, а також їх модифікації та створенні нових, помилкових повідомлень. Відповідно можна виділити три типи порушників, яким повинна протистояти

стеганосистема: пасивний, активний і злочинний порушники. Помітимо, що пасивний зловмисник може бути лише в стеганосистемах прихованої передачі даних. Для систем ЦВДЗ характерні активні та злочинні порушники.

Рис. 1.2. **Класифікація систем цифрової стеганографії**

Для того щоб стеганосистема була надійною, необхідне виконання при її проектуванні ряду вимог.

Безпека системи повинна повністю визначатися таємністю ключа. Це означає, що зловмисник може повністю знати всі алгоритми роботи стеганосистеми та статистичні характеристики множин повідомлень і контейнерів, і це не дасть йому ніякої додаткової інформації про наявність або відсутність повідомлення в даному контейнері.

Знання зловмисником факту наявності повідомлення в якому-небудь контейнері не повинне допомогти йому при виявленні повідомлень в інших контейнерах.

Заповнений контейнер повинен візуально не відрізнятися від незаповненого. Для задоволення цієї вимоги треба, здавалося б, впроваджувати приховане повідомлення у візуально незначущі множини

сигналу. Однак ці ж множини використовують і алгоритми стиску. Тому, якщо зображення буде надалі піддаватися стиску, то приховане повідомлення може зруйнуватися. Отже, біти повинні вбудовуватися у візуально значущі множини, а відносна непомітність може бути досягнута за рахунок використання спеціальних методів, наприклад, модуляції з розширенням спектра.

Стеганосистема ЦВДЗ повинна мати низьку ймовірність помилкового виявлення прихованого повідомлення в сигналі, що його не утримує. У деяких додатках таке виявлення може привести до серйозних наслідків. Наприклад, помилкове виявлення ЦВДЗ на DVD-диску може викликати відмову від його відтворення плеєром.

Повинна забезпечуватися необхідна пропускна здатність (ця вимога актуальна, в основному, для стеганосистем прихованої передачі інформації). У третьому розділі введемо поняття прихованої пропускної здатності й розглянемо шляхи її досягнення.

Стеганосистема повинна мати прийнятну обчислювальну складність реалізації. При цьому можлива асиметрична за складністю реалізації система ЦВДЗ, тобто складний стеганокодер і простий стеганодекодер.

До ЦВДЗ висуваються такі вимоги [5; 11; 55]:

ЦВДЗ повинен легко (обчислювально) витягатися законним користувачем;

ЦВДЗ повинен бути стійким або нестійким до навмисних і випадкових впливів. Якщо ЦВДЗ використовується для підтвердження дійсності, то неприпустима зміна контейнера повинна призводити до руйнування ЦВДЗ (тендітний ЦВДЗ). Якщо ж ЦВДЗ містить ідентифікаційний код, логотип фірми тощо, то він повинен зберегатися при максимальних перекручуваннях контейнера, що звичайно, не приводять до істотних перекручувань вихідного сигналу. Наприклад, у зображенні можуть бути відредаговані колірна гама або яскравість, в аудіозаписі – посилене звучання низьких тонів і т. п. Крім того, ЦВДЗ повинен бути роботоздатним стосовно афінних перетворень зображення, тобто його поворотів, масштабування. При цьому треба розрізняти стійкість самого ЦВДЗ і здатність декодера правильно його виявити. Скажемо, при повороті зображення ЦВДЗ не зруйнується, а декодер може виявитися нездатним виділити його. Існують додатки, коли ЦВДЗ повинен бути стійким стосовно одних перетворень і нестійким стосовно інших. Наприклад, може бути дозволене копіювання зображення (ксерокс, сканер), але накладена заборона на внесення в нього яких-небудь змін.

Повинна бути можливість додавання до стега додаткового ЦВДЗ. Наприклад, на DVD-диску є мітка про допустимість однократного копіювання. Після здійснення такого копіювання необхідно додати мітку про заборону подальшого копіювання. Можна було б, звичайно, видалити перший ЦВДЗ і записати на його місце другий, однак це суперечить припущенню про важковіддаленністю ЦВДЗ. Кращим виходом є додавання ще одного ЦВДЗ, після якого перший не буде братися до уваги. Однак наявність декількох ЦВДЗ на одному повідомленні може полегшити атаку з боку зловмисника, якщо не почати спеціальних заходів.

У цей час технологія ЦВДЗ перебуває в початковій стадії свого розвитку. Як показує практика, повинно пройти років 10 – 20 для того, щоб новий криптографічний метод почав широко використовуватися в суспільстві. Напевно, аналогічна ситуація буде спостерігатися й зі стеганографією. Однією із проблем, пов’язаних зі ЦВДЗ, є різноманіття вимог до них, залежно від додатка. Розглянемо докладніше основні множини застосування ЦВДЗ.

Спочатку розглянемо проблему піратства, або необмеженого неавторизованого копіювання. Наприклад, Аліса продає своє мультимедійне повідомлення Пітеру. Хоча інформація могла бути зашифрована під час передачі, ніщо не перешкодить Пітеру зайнятися її копіюванням після розшифровки. Отже, у цьому випадку потрібен додатковий рівень захисту від копіювання, що не може бути забезпечений традиційними методами. Існує можливість впровадження ЦВДЗ, що дозволяє відтворення та забороняє копіювання інформації.

Важливою проблемою є визначення дійсності отриманої інформації, тобто її автентифікація. Звичайно для автентифікації даних використовуються засоби цифрового підпису. Однак ці засоби не зовсім підходять для забезпечення автентифікації мультимедійної інформації. Справа в тому, що повідомлення, постачене електронним цифровим підписом, повинне зберігатися й передаватися абсолютно точно, "бітів у бітів". Мультимедійна ж інформація може незначно спотворюватися як при зберіганні (за рахунок стиску), так і при передачі (вплив одиночних або пакетних помилок у каналі зв’язку). При цьому її якість залишається припустимою для користувача, але цифрової підпис працювати не буде. Одержувач не зможе відрізнити справжнє, хоча і трохи перекручене повідомлення, від помилкового. Крім того, мультимедійні дані можуть бути перетворені з одного формату в інший. При цьому традиційні засоби

захисту цілісності працювати також не будуть. Можна сказати, що ЦВДЗ здатно захистити саме зміст аудіо-, відеоповідомлення, а не його цифрове подання у вигляді послідовності бітів. Крім того, важливим недоліком цифрового підпису є те, що його легко видалити із завіреного ним повідомлення, після чого прилаштувати до нього новий підпис. Видалення підпису дозволить зловмиснику відмовитися від авторства, або ввести в оману законного одержувача щодо авторства повідомлення. Система ЦВДЗ проектується таким чином, щоб виключити можливість подібних порушень. Як видно з рис. 1.3 застосування ЦВДЗ не обмежується додатками безпеки інформації.

Рис. 1.3. **Потенційні множини застосування стеганографії**

Основні множини використання технології ЦВДЗ можуть бути об’єднані в чотири групи: захист від копіювання (використання), прихована анотація документів, доказ автентичності інформації та прихований зв’язок.

Популярність мультимедіа-технологій викликало множину дослід-жень, пов’язаних з розробкою алгоритмів ЦВДЗ для використання в стандартах MP3, MPEG-4, JPEG2000, захисту DVD- дисків від копіювання.

1.3. Математична модель та структурна схема стеганографічної системи. Класифікація контейнерів

У загальному випадку стеганосистема може бути розглянута як система зв'язку [24]. Узагальнена структурна схема стеганосистеми наведена на рис. 1.4.

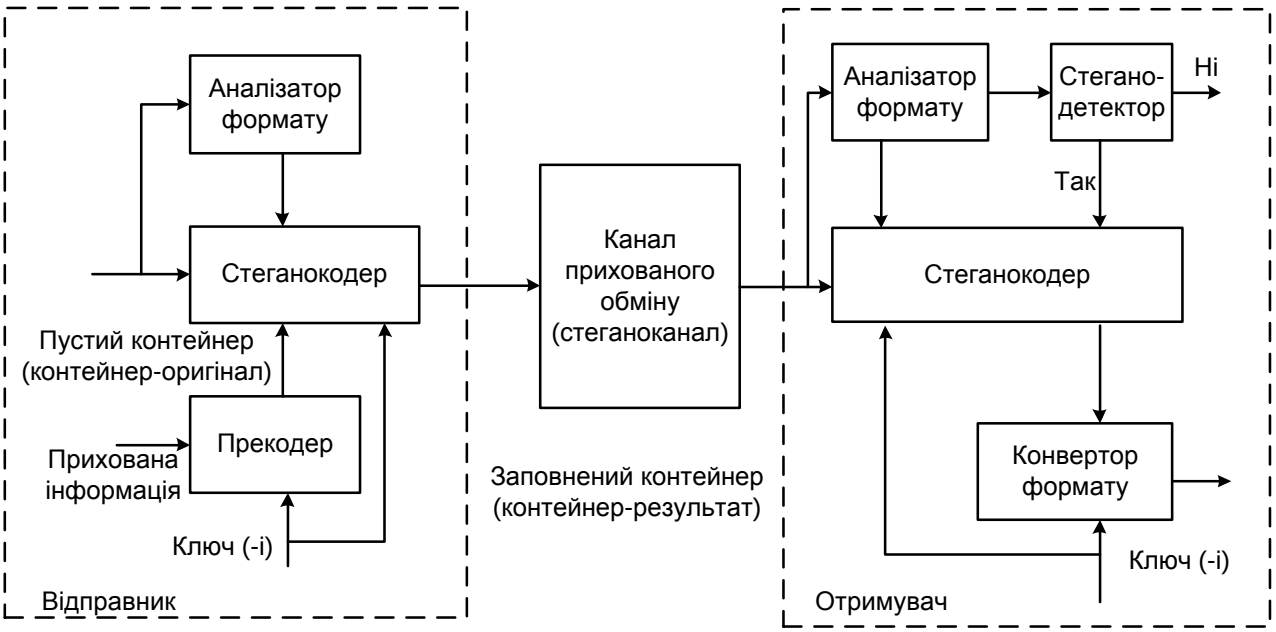


Рис. 1.4. **Структурна схема стеганосистеми як системи зв'язку**

Основними стеганографічними поняттями є повідомлення і контейнер. *Повідомлення* $m \in M$ – це секретна інформація, наявність якої необхідно приховати, $M = \{m_1, m_2, \ldots, m_n\}$ – множина всіх повідомлень.

Контейнером $c \in C$ називається несекретна інформація, яку можна використовувати для приховання повідомлення, $C = \{c_1, c_2, \ldots, c_q\}$ – множина всіх контейнерів, причому $q >> n$. Як повідомлення й контейнер можуть виступати як звичайний текст, так і файли мультимедійного формату.

Порожній контейнер (або так званий контейнер-оригінал) – це контейнер *c*, що не містить прихованої інформації. *Заповнений контейнер* (контейнер-результат) – контейнер *c*, що містить приховану інформацію *m* (*cm*). Одна з вимог, що при цьому висувається: контейнер-результат не повинен візуально відрізнятися від контейнера-оригіналу. Виділяють два основних типи контейнера: потоковий і фіксований.

Потоковий контейнер становить послідовність бітів, що безупинно змінюються. Повідомлення вбудовується в нього в реальному масштабі

часу, тому в кодері заздалегідь невідомо, чи вистачить розмірів контейнера для передачі всього повідомлення. В один контейнер великого розміру може бути убудовано кілька повідомлень. Інтервали між убудованими бітами визначаються генератором псевдовипадкової послідовності (ПВП) із рівномірним розподілом інтервалів між відліками.

Основна проблема полягає у виконанні синхронізації, визначенні початку та кінця послідовності. Якщо в даних контейнера існують біти синхронізації, заголовки пакетів тощо, то прихована інформація може випливати відразу ж після них. Складність організації синхронізації є перевагою з погляду забезпечення прихованості передачі. На жаль, на сьогоднішній день практично відсутні роботи, присвячені розробці стеганосистем з потоковим контейнером.

Як приклад перспективної реалізації потокового контейнера можна навести стеганоприставку до звичайного телефону. При цьому під прикриттям пересічної, несуттєвої телефонної розмови можна передавати іншу розмову, дані та ін. Не знаючи секретного ключа, не можна не тільки довідатися про зміст прихованої передачі, але й про самий факт її існування.

У фіксованому контейнері розміри і характеристики останнього заздалегідь відомі. Це дозволяє виконувати вкладення даних оптимальним (у визначеному змісті) чином. Далі будуть розглядатися переважно фіксовані контейнери (надалі – контейнери).

Контейнер може бути вибраним, випадковим або нав’язаним. *Вибраний контейнер* залежить від убудованого повідомлення, а в граничному випадку є його функцією. Такий тип контейнера більше характерний саме для стеганографії. *Нав’язаний контейнер* з’являється, коли особа, що надає контейнер, підозрює про можливу приховану переписку і бажає їй запобігти. На практиці ж найчастіше мають справу з випадковим контейнером.

Приховання інформації, що переважно має великий обсяг, висуває істотні вимоги до контейнера, розмір якого повинен щонайменше в кілька разів перевищувати розмір даних, що вбудовуються. Зрозуміло, що для збільшення прихованості зазначене співвідношення повинне бути як можна більшим.

Перед тим як виконати вкладення повідомлення в контейнер, його необхідно перетворити в певний зручний для впакування вид. Крім того, перед упакуванням у контейнер, для підвищення захищеності секретної інформації останню можна зашифрувати досить стійким криптографічним кодом. У багатьох випадках також бажана стійкість отриманого стегаповідомлення до перекручувань (у тому числі і злочинних).

У процесі передачі звук, зображення або яка-небудь інша інформація, використовувана як контейнер, може піддаватися різним трансформаціям (у тому числі з використанням алгоритмів із втратою даних): зміна обсягу, перетворення в інший формат і т. п., тому для збереження цілісності убудованого повідомлення може знадобитися використання коду з виправленням помилок (завадостійке кодування).

Початкову обробку приховуваної інформації виконує наведений на рис. 1.4 прекодер. Як одну з найважливіших попередніх обробок повідомлення (а також і контейнера) можна назвати обчислення його узагальненого перетворення Фур'є. Це дозволяє здійснити вбудовування даних у спектральній множині, що значно підвищує їх стійкість до перекручувань.

Слід зазначити, що для збільшення таємності вбудовування попередня обробка досить часто виконується з використанням ключа.

Упакування повідомлення в контейнер (з урахуванням формату даних, що представляють контейнер) виконуються за допомогою стегакодера. Вкладення відбувається, наприклад, шляхом модифікації найменших значущих бітів контейнера. Взагалі, саме алгоритм (стратегія) внесення елементів повідомлення в контейнер визначає методи стеганографії, які, у свою чергу, діляться на визначені групи, наприклад, залежно від того, файл якого формату був обраний як контейнер.

У більшості стеганосистем для впакування та витягнення повідомлень використовується ключ, що визначає секретний алгоритм, який визначає порядок внесення повідомлення в контейнер. За аналогією із криптографією, тип ключа спричиняє існування двох типів стеганосистем :

з *секретним ключем* – використовується один ключ, що визначається до початку обміну стеганограмою або передається захищеним каналом;

з *відкритим ключем* – для впакування та розпакування повідомлення використовуються різні ключі, які відрізняються таким чином, що за допомогою обчислень неможливо одержати один ключ із іншого, тому один із ключів (відкритий) може вільно передаватися по незахищеному каналу. Як секретний алгоритм може бути використаний генератор псевдовипадкової послідовності бітів. Якісний генератор ПВП, орієнтований на використання в системах захисту інформації, повинен відповідати певним вимогам. Перелічимо деякі з них.

Криптографічна стійкість – відсутність у зловмисника можливості передбачити наступний бітів на підставі відомих йому попередніх з імовірністю, відмінною від 1/2. На практиці криптографічна стійкість оцінюється статистичними методами. Національним інститутом стандар-

тів і технологій США (НІСТ) розроблений посібник із проведення статистичних випробувань генераторів ПВП, орієнтованих на використання в задачах криптографічного захисту інформації.

Статистичні властивості – ПВП за своїми статистичними властивостями не повинна істотно відрізнятися від істинно випадкової послідовності.

Великий період формованої послідовності.

Ефективна апаратно-програмна реалізація. Статистично (криптографічно) безпечний генератор ПВП повинен відповідати таким вимогам:

жоден статистичний тест не визначає в ПВП ніяких закономірностей, іншими словами, не відрізняє цю послідовність від істинно випадкової;

при ініціалізації випадковими значеннями генератор породжує статистично незалежні псевдовипадкові послідовності.

Як основа генератора може використовуватися, наприклад, лінійний рекурентний регістр. Тоді адресатам для забезпечення зв'язку повинне повідомлятися початкове заповнення цього регістра. Числа, породжувані генератором ПВП, можуть визначати позиції модифікованих відліків у випадку фіксованого контейнера або інтервали між ними у випадку потокового контейнера.

Слід зазначити, що метод випадкового вибору величини інтервалу між убудованими бітами не є досить ефективним за двома причинами. По-перше, приховані дані повинні бути розподілені по всьому контейнеру, тому рівномірний розподіл довжини інтервалів (від найменшого до найбільшого) може бути досягнуто тільки приблизно, оскільки повинна існувати впевненість у тому, що все повідомлення убудоване (тобто помістилося в контейнер). По-друге, довжина інтервалів між відліками шуму (у багатьох моделях сигнал-контейнер розглядається як адитивний шум) розподілена не за рівномірним, а за експонентним законом. Генератор ПВП із експонентним розподілом інтервалів складний у реалізації. Приховувана інформація заноситься відповідно до ключа в ті біти, модифікація яких не приводить до істотних перекручувань контейнера. Ці біти утворять так званий стеганошлях. Під "істотним" мається на увазі перекручування, що приводить до зростання ймовірності виявлення факту наявності прихованого повідомлення після проведення стеганоаналізу.

Стеганографічний канал – канал передачі контейнера-результату (взагалі, існування каналу як, власне кажучи, і одержувача – найбільш узагальнений випадок, оскільки заповнений контейнер може, наприклад, зберігатися у "відправника'', що поставив перед собою мету обмежити

неавторизований доступ до певної інформації. У цьому випадку відправник виступає в ролі одержувача). Під час перебування в стеганографічному каналі контейнер, що містить приховане повідомлення, може піддаватися навмисним атакам або випадковим перешкодам.

У стеганодетекторі визначається наявність у контейнері (можливо вже зміненому) прихованих даних. Ця зміна може бути обумовлена впливом помилок у каналі зв'язку, операцій обробки сигналу, навмисних атак порушників. Як вже відзначалося вище, у багатьох моделях стеганосистем сигнал-контейнер розглядається як адитивний шум. Тоді завдання виявлення й виділення стеганоспілкування є класичним для теорії зв'язку. Але такий підхід не враховує двох факторів: невипадкового характеру контейнера й вимог зі збереження його якостей. Ці моменти не зустрічаються у відомій теорії виявлення й виділення сигналів на тлі адитивного шуму. Очевидно, що їх облік дозволить побудувати більш ефективні стеганосистеми.

Розрізняють стеганодетектори, призначені тільки для виявлення факту наявності убудованого повідомлення, і пристрої, призначені для виділення цього повідомлення з контейнера, – стеганодекодери.

Отже, у стеганосистемі відбувається об'єднання двох типів інформації таким чином, щоб вони по-різному сприймалися принципово різними детекторами. У якості одного з детекторів виступає система виділення прихованого повідомлення, у якості іншого – людина.

Алгоритм вбудовування повідомлення в найпростішому випадку складається із двох основних етапів:

Вбудовування в стегакодері секретного повідомлення в контейнер-оригінал.

Виявлення (виділення) у стеганодетекторі (декодері) прихованого зашифрованого повідомлення з контейнера-результату.

Виходячи із цього, слід розглянути математичну модель стеганосистеми. Процес тривіального стеганографічного перетворення описується залежностями:

$$E : C \times M \to S \tag{1.1}$$

$$D : S \to M, \tag{1.2}$$

де $S = \{(c_1, m_1),(c_2, m_2),...,(c_n, m_n),...,(c_q, m_q)\} = \{s_1, s_2,..., s_q\}$ – множина контейнерів-результатів (стеганограм).

Залежність (1.1) описує процес приховання інформації, залежність (1.2) – витягнення прихованої інформації. Необхідною умовою при цьому є відсутність "перетинання" [11], тобто, якщо $m_a \neq m_b$, причому $m_a, m_b \in M$, а $(c_a, m_a), (c_b, m_b) \in S$, то $E(c_a, m_a) \cap E(c_b, m_b) = \varnothing$.

Крім того, необхідно, щоб потужність множини $|C| \geq |M|$. При цьому обидва адресати (відправник і одержувач) повинні знати алгоритм прямого (E) і зворотного (D) стеганографічних перетворень.

Отже, у загальному випадку стеганосистема – це сукупність $\Sigma = (C, M, S, E, D)$ контейнерів (оригіналів і результатів), повідомлень і перетворень, які їх пов'язують.

Для більшості стеганосистем множина контейнерів C вибирається таким чином, щоб у результаті стеганографичного перетворення (1.1) заповнений контейнер і контейнер-оригінал були подібні, що формально може бути оцінене за допомогою функції подоби [11; 46; 51].

Визначення 1.1. Нехай C – непуста множина, тоді функція $sim(C) \to (-\infty, 1)$ є функцією подоби на множині C, якщо для яких-небудь $x, y \in C$ справедливо, що $sim(x, y) = 1$ у випадку $x = y$ та $sim(x, y) < 1$ при $x \neq y$.

Стеганосистема може вважатися *надійною*, якщо $sim[c, E(c, m)] \approx 1$ для всіх m $\in M$ і $c \in C$, причому як контейнер c повинен обиратися такий, котрий раніше не використовувався. Крім того, неавторизована особа не повинна мати доступ до набору контейнерів, використовуваних для секретного зв'язку.

Вибір визначеного контейнера із набору можливих контейнерів C може здійснюватися довільно (так званий сурогатний метод вибору контейнера) або шляхом обрання найбільш придатного, котрий менше інших зміниться під час стеганоперетворення (селективний метод). В останньому випадку контейнер обирається відповідно до правила:

$$c = \max_{x \in C} sim[x, E(x, m)]. \qquad (1.3)$$

Також слід зазначити, що функції прямого (E) і зворотного (D) стеганографічних перетворень в загальному випадку можуть бути довільними (але, звичайно, відповідають одна одній), однак на практиці вимоги до стійкості прихованої інформації накладають на зазначені

функції визначені обмеження. Так, у переважній більшості випадків, $E(c,m) \approx E(c+\delta,m)$, або $D[E(c,m)] \approx D[E(c+\delta,m)] = m$ – тобто незначно *модифікований контейнер* (на величину δ) не повинен приводити до зміни прихованої в ньому інформації.

Контрольні запитання

1. Охарактеризуйте криптографічний захист інформації та приховання існування інформації.
2. Охарактеризуйте цифрову обробку сигналів. Що таке цифрова стеганографія?
3. Назвіть галузі застосування стеганографії.
4. Назвіть методи формування цифрових водяних знаків.
5. Що включає математична модель стеганографічної системи?
6. Побудуйте структурну схему стеганографічної системи.
7. Охарактеризуйте класифікацію стеганоконтейнерів.
8. Чим відрізняються прихований та відкритий стеганоконтейнери?
9. Чим відрізняються порожній та потоковий контейнери?
10. Назвіть основні вимоги до стеганосистем з секретним ключем.
11. Назвіть основні вимоги до стеганосистем з відкритим ключем.
12. Що таке криптографічна стійкість?
13. Назвіть основні статистичні властивості ПВП.
14. Що таке великий період формованої послідовності?
15. Охарактеризуйте ефективну апаратно-програмну реалізацію ПВП.
16. Що таке стеганографічний канал?
17. Назвіть основні вимоги до стеганографічного каналу.
18. Чим відрізняються стеганодетектори від стеганодекодерів?
19. У чому полягає афінне перетворення зображення?
20. Які етапи є в алгоритмі вбудовування повідомлення в найпростішому випадку?
21. Охарактеризуйте структурну схему стеганосистеми як системи зв’язку.
22. Охарактеризуйте методи стеганографії.
23. Побудуйте структурну схему типічної стеганосистеми цифрових водяних знаків.
24. Що таке цифрові водяні знаки?
25. Назвіть основні множини використання технології ЦВДЗ.

Розділ 2. Приховування даних у нерухомих зображеннях

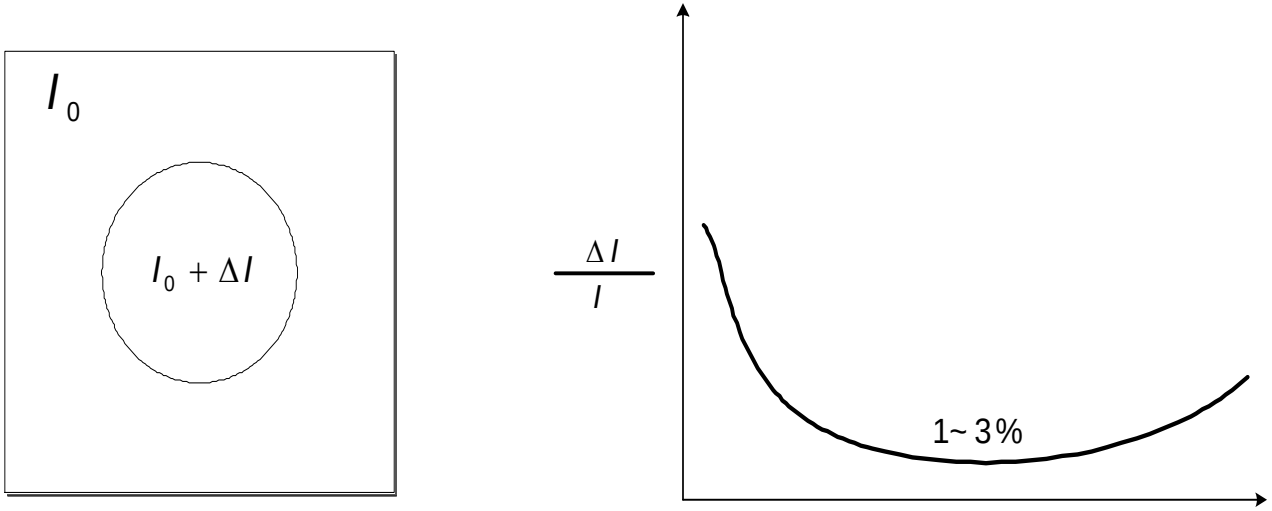
2.1. Особливості зорової системи людини (ЗСЛ). Основні властивості ЗСЛ, що використовуються при приховуванні даних у зображеннях

Властивості зорової системи людини можна розділити на дві групи: низькорівневі ("фізіологічні") і високорівневі ("психофізіологічні"). Аж до середини 1990-х років дослідники брали до уваги, головним чином, низькорівневі властивості зору. В останні роки намітилася тенденція побудови стеганоалгоритмів з обліком і високорівневих характеристик ЗСЛ.

Виділимо три найбільш важливі низькорівневі властивості, що впливають на помітність стороннього шуму в зображенні: чутливість до зміни яскравості зображення, частотна чутливість і ефект маскування.

Чутливість до зміни яскравості визначається в такий спосіб [33]. Випробуваному показують деяку однотонну картинку (рис. 2.1а). Після того, як око адаптувалося до її освітленості I, "настроїлося на неї", поступово змінюють яскравість навколо центральної плями. Зміну освітленості ΔI продовжують доти, поки вона не буде виявлена.

На рис. 2.1б показана залежність мінімального контрасту $\Delta I / I$ від яскравості I (для зручності поміняли звичне розташування осей).



а) б)

Рис. 2.1. **Властивості зорової системи людини**

Як видно з рис. 2.1, для середнього діапазону зміни яскравості контраст приблизно постійний (аналогія із кратномасштабним аналізом і вейвлетами), тоді як для малих і більших яскравостей значення порога нерозрізненості зростає. Було встановлено, що $\Delta I \approx 0.01 - 0.03 I$ для середніх значень яскравості.

Результати новітніх досліджень суперечать "класичній" точці зору і показують, що при малих значеннях яскравості ЗСЛ поріг нерозрізненості зменшується, тобто ЗСЛ більш чутлива до шуму в цьому діапазоні.

Частотна чутливість ЗСЛ проявляється в тому, що людина набагато більш сприйнятлива до низькочастотного (НЧ), ніж до високочастотного (ВЧ) шуму. Це пов’язане з нерівномірністю амплітудно-частотної характеристики системи зору людини. Експериментально її можна визначити за допомогою того ж досвіду, що і при яскравій чутливості. Але цього разу в центральному квадраті змінюються просторові частоти доти, поки зміни не стануть помітними.

Елементи ЗСЛ розділяють відеосигнал, що надходить, на окремі компоненти. Кожна складова збуджує нервові закінчення ока через ряд підканалів. Вирізнювані оком компоненти мають різні просторові й частотні характеристики, а також різну орієнтацію (горизонтальну, вертикальну, діагональну) [66]. У випадку одночасного впливу на око двох компонентів з подібними характеристиками збуджуються ті самі підканали. Це приводить до ефекту маскування, що полягає в збільшенні порога виявлення відеосигналу в присутності іншого сигналу, що володіє аналогічними характеристиками. Тому адитивний шум набагато помітніше на гладких ділянках зображення, ніж на високочастотних, тобто в останньому випадку спостерігається маскування. Найбільш сильно ефект маскування проявляється, коли обидва сигнали мають однакову орієнтацію й місце розташування.

Можна показати, що частотна чутливість тісно пов’язана з яскравісною. Відомо також і вираз для визначення порога маскування на основі відомої яскравісної чутливості, що дозволяє знайти метрику перекручування зображення, що враховує властивості ЗСЛ. Такого типу математичні моделі добре розроблені для випадку квантування коефіцієнтів дискретного косинусного перетворення зображення, тому що саме воно застосовується в стандарті JPEG.

Ефект маскування в просторовій множині може бути пояснений шляхом побудови стохастичних моделей зображення. При цьому

зображення представляється у вигляді марківського випадкового поля, розподіл імовірностей якого підкоряється, наприклад, узагальненому гауссівському закону.

Таким чином, можна запропонувати таку узагальнену схему впровадження даних у зображення:

1. Виконати фільтрацію зображення за допомогою орієнтованих смугових фільтрів. При цьому одержимо розподіл енергії по частотно-просторових компонентах.

2. Обчислити поріг маскування на основі знання локальної величини енергії.

3. Масштабувати значення енергії впроваджуваного ЦВДЗ у кожному компоненті так, щоб воно було менше порога маскування.

Багато алгоритмів вбудовування інформації, як ми побачимо, так чи інакше використовують цю схему.

Високорівневі властивості ЗСЛ поки рідко враховуються при побудові стеганоалгоритмів. Їх відмінністю від низькорівневих є те, що ці властивості проявляються "удруге", обробивши первинну інформацію від ЗСЛ, мозок видає команди на її "підстроювання" під зображення. Перелічимо основні із цих властивостей:

1. *Чутливість до контрасту*. Висококонтрастні ділянки зображення, перепади яскравості привертаються до себе значну увагу.

2. *Чутливість до розміру*. Більші ділянки зображення "помітніші" менших за розміром. Причому існує поріг насичення, коли подальше збільшення розміру не істотне.

3. *Чутливість до форми*. Довгі й тонкі об’єкти привертають більшу увагу, ніж круглі однорідні.

4. *Чутливість до кольору*. Деякі кольори (наприклад, червоний) "помітніші" інших. Цей ефект підсилюється, якщо тло заднього плану відрізняється від кольору фігур на ньому.

5. *Чутливість до місця розташування*. Людина схильна у першу чергу розглядати центр зображення.

6. Люди звичайно уважніше до зображень переднього плану, ніж заднього.

7. Якщо на зображенні є люди, у першу чергу людина зверне свою увагу на них. На фотографії людина звертає першочергову увагу на особу, очі, рот, руки.

8. *Чутливість до зовнішніх подразників*. Рух очей спостерігача залежить від конкретної обстановки, від отриманих їм перед переглядом або під час його інструкцій, додаткової інформації.

2.2. Цифрові формати нерухомих зображень (формати BMP, GIF, TIFF, JPEG). Особливості комп'ютерної обробки зображень

Формат BMP. За рішенням розроблювачів формат BMP-файла не прив'язаний до конкретної апаратної платформи. Цей файл складається із чотирьох частин: заголовка, інформаційного заголовка, таблиці кольорів (палітри) і даних зображення. Якщо у файлі зберігається зображення із глибиною кольору 24 біти (16 млн кольорів), то таблиця кольорів може бути відсутньою, однак у нашому 256-кольоровому випадку вона є. Структура кожної із частин файла, що зберігає 256-кольорове зображення, подана в табл. 2.1.

Заголовок файла починається із сигнатури "BM", а потім іде довжина файла, виражена в байтах. Наступні 4 байти зарезервовані для подальших розширень формату, а закінчується цей заголовок зсувом від початку файла до записаних у ньому даних зображення. При 256 кольорах цей зсув становить 1078 – саме стільки й доводиться пропустити, щоб добратися до даних.

Інформаційний заголовок починається із власної довжини (вона може змінюватися, але для 256-кольорового файла становить 40 байтів) і містить розміри зображення, роздільну здатність, характеристики подання кольору й інших параметрів.

Ширина та *висота зображення* задаються в точках растра й пояснень, мабуть, не вимагають.

Кількість площин, що можуть застосовуватися у файлах, які мають невелику глибину кольору. При кількості кольорів 256 і більше вона завжди дорівнює 1, тому зараз це поле вже можна вважати застарілим, але для сумісності воно зберігається.

Глибина кольору вважається найважливішою характеристикою способу подання кольору у файлі й виміряється в бітах на точку. У цьому випадку вона дорівнює 8.

Компресія в BMP-файлах звичайно не використовується, але поле в заголовку для неї передбачено. Звичайно вона дорівнює 0, і це означає, що зображення не стисле. Надалі будемо використовувати тільки такі файли.

Розмір зображення – кількість байтів пам'яті, що вимагаються для зберігання цього зображення, не вважаючи даної палітри.

Таблиця 2.1

Структура BMP-файла

Ім’я	Довжина	Зсув	Опис
Заголовок файла (BitMapFileHeader)			
Type	2	0	Сигнатура "BM"
Size	4	2	Розмір файла
Reserved 1	2	6	Зарезервовано
Reserved 2	2	8	Зарезервовано
OffsetBits	4	10	Зсув зображення від початку файла
Інформаційний заголовок (BitMapInfoHeader)			
Size	4	14	Довжина заголовка
Width	4	18	Ширина зображення, точка
Height	4	22	Висота зображення, точка
Planes	2	26	Кількість площин
BitCount	2	28	Глибина кольору, бітів на точку
Compression	4	30	Тип компресії (0 – незжате зображення)
SizeImage	4	34	Розмір зображення, байт
XpelsPerMeter	4	38	Горизонтальна роздільна здатність, точка на метр
YpelsPerMeter	4	42	Вертикальна роздільна здатність, точка на метр
ColorsUsed	4	46	Кількість використовуваних кольорів (0 – максимально можливе для даної глибини кольору)
ColorsImportant	4	50	Кількість основних кольорів
Таблиця кольорів (палітра) (ColorTable)			
ColorTable	1024	54	256 елементів по 4 байти
Дані зображення (BitMap Array)			
Image	Size	1078	Зображення, записане рядками злава направо і знизу вгору

Горизонтальний і *вертикальний роздільні здатності* виміряються в точках растра на метр. Вони особливо важливі для збереження масштабу відсканованих картинок. Зображення, створені за допомогою графічних редакторів, як правило, мають у цих полях нулі.

Кількість кольорів дозволяє скоротити розмір таблиці палітри, якщо в зображенні реально присутньо менше кольорів, чим це допускає обрана глибина кольору. Однак на практиці такі файли майже не

зустрічаються. Якщо кількість кольорів приймає значення, максимально припустиме глибиною кольору, наприклад 256 кольорів при 8 бітах, полі обнуляють.

Кількість основних кольорів – іде з початку палітри, і його бажано виводити без перекручувань. Дане поле буває важливе тоді, коли максимальна кількість кольорів дисплея була менше, ніж у палітрі BMP-файла. При розробці формату, мабуть, приймалося, що кольорі, які найбільш часто зустрічаються, будуть розташовуватися на початку таблиці. Зараз цієї вимоги практично не дотримуються, тобто кольори не впорядковуються по частоті, з якою вони зустрічаються у файлі. Це дуже важливо, оскільки палітри двох різних файлів, навіть складених з тих самих кольорів, містили б їх (кольори) у різному порядку, що могло істотно ускладнити одночасне виведення таких зображень на екран.

За інформаційним заголовком слідує таблиця кольорів, що становить масив з 256 (за кількістю кольорів) 4-байтових полів. Кожне поле відповідає своєму кольору в палітрі, а три байти із чотирьох – компонентам синьої, зеленої і червоної складових для цього кольору. Останній, найстарший байт кожного поля зарезервований і дорівнює 0.

Після таблиці кольорів знаходяться дані зображення, що по рядках растра записані знизу вгору, а усередині рядка – зліва направо. Оскільки на деяких платформах неможливо визначити одиницю даних, що менше 4 байтів, довжина кожного рядка вирівняна на границю в 4 байти, тобто при довжині рядка, некратній чотирьом, вона доповнюється нулями. Цю обставину обов’язково треба враховувати при зчитуванні файла, хоча, можливо, краще заздалегідь подбати, щоб горизонтальні розміри всіх зображень були кратні 4.

Формат файла був розроблений універсальним для різних платформ, тому немає нічого дивного в тому, що кольори палітри зберігаються в ньому інакше, чим прийнято для VGA. Під час виконання процедури читання виробляється необхідне перекодування.

Формат GIF. "GIF" (tm) – це стандарт фірми CompuServe для визначення растрових кольорових зображень. Цей формат дозволяє висвічувати на різному встаткуванні графічні високоякісні зображення з більшою роздільною здатністю і має на увазі механізм обміну й висвічування зображень. Описаний у справжньому документі формат зображень був розроблений для підтримки теперішньої й майбутньої технології обробки зображень і буде надалі основою для майбутніх графічних продуктів CompuServe.

Головне завдання справжнього документа полягає в тому, щоб забезпечити програмістів необхідною технічною інформацією для написання декодерів і кодерів GIF. Тому в документі використовується термінологія, пов'язана із загальними питаннями графіків і програмування. Перший розділ справжнього документа описує формат даних GIF і його компоненти в додатку до декодерів GIF, поза залежністю від того чи є вони окремою програмою або частиною пакета зв'язку. Додаток В відноситься до декодерів, що є частиною пакетів зв'язку й описує протокол, необхідний для входу та існування режиму GIF, і відповідає на ряд специфічних питань. Загальний формат файла GIF надано на рис. 2.2.

Рис. 2.2. **Загальний формат файла GIF**

Наявність на початку файла спеціального "підпису" указує, що наступні дані є дійсно потоком даних зображення у форматі GIF. Цей "підпис" складається з таких шести символів: GIF 87a.

Три останніх символи "87a" можуть розглядатися як номер версії для даного конкретного визначення GIF і будуть використовуватися надалі як посилання на документ із описом GIF залежно від номера версії.

Дескриптор екрана описує загальні параметри для всіх наступних зображень у форматі GIF. Він визначає розміри простору зображення

або необхідного логічного екрана, існування інформації про таблицю кольорів і "глибину" екрана. Ця інформація запам’ятовується у вигляді серії 8-бітових байтів, як показано на рис. 2.3.

Біти 7 6 5 4 3 2 1 0	Байти	
	1 Ширина екрана 2	Ширина растра у пікселах (спочатку LSB).
	3 Висота екрана 4	Висота растра у пікселах (спочатку LSB).
M cr 0 pixel	5	M = 1, за дескриптором слідує глобальна таблиця кольорів. cr + 1 – кількість бітів кольорової роздільної здатності. pixel + 1 – кількість бітів/піксел зображенні.
background	6 фон	Кольоровий індекс фону екрана (колір визначається з глобальної таблиці кольорів або з таблиці за замовчуванням).
0 0 0 0 0 0 0 0	7	

Рис. 2.3. **Дескриптор екрана формату GIF**

Ширина та висота логічного екрана може бути більше розмірів фізичного екрана. Спосіб висвічування зображень більших, ніж розміри фізичного екрана залежить від реалізації і може використовувати переваги конкретного встаткування (наприклад, вікна скролінга в Macintosh scrolling windows). У противному разі зображення буде усічено по краях екрана. Значення "pixel" також визначає кількість кольорів у зображенні.

Діапазон значень "pixel" становить від 0 до 7, що відповідає від 1 до 8 бітам. Це транслюється в діапазон від 2 (чорно-білі зображення) до 256 кольорів. Біт 3 у байті 5 зарезервований для майбутніх визначень і повинен бути нульовим.

Глобальна таблиця кольорів є необов’язковою і рекомендується для зображень, де потрібна точна передача кольорів. На існування цієї таблиці вказує поле "M" у 5 байті дескриптора екрана. Колірна таблиця може бути також пов’язана з кожним зображенням в GIF-файлі, що буде описано пізніше. Однак, звичайно, ця глобальна таблиця буде

використовуватися через обмеження, що існують у цей час у доступному устаткуванні. Прапор "M" у дескрипторі конкретного зображення звичайно дорівнює 0. Якщо глобальна таблиця кольорів присутня, її визначення слідує безпосередньо за дескриптором екрана. Кількість елементів колірної таблиці, що випливає за дескриптором екрана, дорівнює 2**(кількість бітів/піксел), причому кожний елемент складається із трьох байтів, значення яких описують, відповідно відносну інтенсивність червоного, зеленого і синього кольорів. Структура блоку колірної таблиці показана на рис. 2.4.

Біти 7 6 5 4 3 2 1 0	Байти	
Інтенсивність червоного	1	Значення червоного кольору для кольору 0
Інтенсивність зеленого	2	Значення зеленого кольору для кольору 0
Інтенсивність синього	3	Значення синього кольору для кольору 0
Інтенсивність червоного	4	Значення червоного кольору для кольору 0
Інтенсивність зеленого	5	Значення зеленого кольору для кольору 0
Інтенсивність синього	6	Значення синього кольору для кольору 0
…	…	…
Значення для подальших номерів різних кольорів		

Рис. 2.4. **Структура блоку колірної таблиці**

Одержуване значення кожного піксела при висвічувані зображення буде відповідати найближчому доступному кольору з колірної таблиці дисплея. Колірні компоненти становлять значення відносної інтенсивності від нульової (0) до полної (255). Білий колір може бути представлений як (255, 255, 255), чорний – як (0, 0, 0), а жовтий – як (180,180,0). При висвічуванні на дисплеях, які підтримують менше 8 бітів на колірний компонент, використовуються старші біти. При створенні елементів колірної таблиці GIF на апаратурі, що підтримує менше 8 бітів на компонент, значення апаратного компонента повинне бути конвертоване в 8-бітний формат за формулою:

<значення_в_таблиці> = <компонента>*255/(2**<кількість_бітів> – 1).

Це забезпечує точне переведення кольорів для всіх дисплеїв. У випадку створення зображення GIF на апаратурі без можливості колірної палітри, повинна бути створена фіксована палітра на основі доступних для даного встаткування кольорів. Якщо зазначено відсутність

глобальної таблиці кольорів, колірна таблиця за замовчуванням генерується внутрішнім чином так, що кожний колірний індекс дорівнює апаратному колірному індексу modulo <*n*>, де <*n*> – кількість доступних кольорів на встаткуванні.

Дескриптор зображення визначає дійсне розташування та розміри наступного зображення усередині простору, визначеного в дескрипторі екрана. Також визначаються прапори, що вказують на присутність локальної таблиці для пошуку кольорів і визначення послідовності пікселів. Кожний дескриптор зображення починається із символу-роздільника зображень.

Роль роздільника зображень полягає просто в синхронізації при вході в дескриптор зображення. Це бажано, якщо GIF-файл складається більш ніж з одного зображення. Цей символ визначений як шістнадцятирічне 0×2C або "," (кома). Як тільки цей символ зустрічається між зображеннями, безпосередньо за ним слідує дескриптор зображення.

Будь-який символ, що зустрічається між кінцем попереднього зображення й символом-роздільником зображення, ігнорується. Це дозволить при наступних модифікаціях GIF допускати присутність декількох форматів і правильно ігнорувати їх старими декодерами. Структура дескриптора зображення показана на рис. 2.5.

Біти 7 6 5 4 3 2 1 0						Байти	
0 0 1 0 1 1 0 0						1	’,’ - Символ-роздільник зображення.
						2 Лівий край 3	Початок зображення в пікселах відносно лівого краю екрана (спочатку LSB).
						4 Верхній край 5	Початок зображення в пікселах відносно верхнього краю екрана (спочатку LSB).
						6 Ширина 7	Ширина зображення в пікселах (спочатку LSB).
						8 Висота 9	Висота зображення в пікселах (спочатку LSB).
M	I	0	0	0	pixel	10	M = 0 – використовувати глобальну таблицю кольорів, ігнорувати “pixel”. M = 1 – далі слідує локальна таблиця кольорів, використовувати “pixel”. I=0 – зображення відформатоване в послідовному порядку. I=1 – зображення відформатоване в порядку переплетення. pixel+1 – кількість бітів на піксел у даному зображенні.

Рис. 2.5. **Структура дескриптора зображення**

Опис положення та розмірів екрана повинен перебувати усередині матриці, визначеної в дескрипторі екрана. З іншого боку, немає необхідності, щоб зображення повністю заповнювало весь екран.

Локальна таблиця кольорів необов'язкова і визначена тут для майбутнього використання. Якщо встановлено бітів "*M*" байта 10 у дескрипторі зображення, то за дескриптором зображення слідує локальна таблиця кольорів, що відноситься тільки до наступного зображення. Після обробки зображення колірну таблицю варто привести до тієї, котра була визначена після дескриптора екрана. Помітимо, що поле "pixel" байта 10 у дескрипторі зображення використовується тільки в тому випадку, якщо зазначено локальну таблицю кольорів. Вона визначає не тільки розмір піксела (кількість бітів у ньому), але кількість елементів наступної колірної таблиці. Кількість бітів на піксел також варто відновити до того значення, що було визначено в дескрипторі екрана, після того, як закінчиться обробка зображення.

Растрові дані – формат самого зображення визначений як серія значень номерів пікселів, які утворять зображення. Пікселі запам'ятовуються зліва направо послідовно рядками зображення.

За замовчуванням рядки записуються послідовно, зверху вниз. У тому випадку, якщо встановлено бітів "I" у байті 10 дескриптора зображення, то порядок рядків при записі зображення відповідає чотирипрохідному процесу. При першому проході записується кожний 8-й рядок, починаючи з верхнього рядка вікна зображення. При другому проході записується кожний 8-й рядок, починаючи з п'ятого рядка зверху. На третьому проході записується кожний 4-й рядок, починаючи із третього рядка вікна. Четвертий прохід завершує зображення, записуючи кожний другий рядок, починаючи із другого рядка зверху. На рис. 2.6 наведено графічний опис цього процесу.

Значення пікселів зображення обробляються як колірні індекси, що вказують на існуючу таблицю кольорів. У результаті виходить колірне значення з таблиці, що реально відтворюється на екрані. Ці серії колірних індексів, кількість яких дорівнює ширині зображення, перемноженій на висоту зображення, пропускаються через потік даних зображення GIF по одному значенню на піксел, стискуються та упаковуються відповідно до версії алгоритму стиску LZW.

Зображення
Ряд. Прох.1 Прох.2 Прох.3 Прох.4 Результат

```
 0 **1a**          **1a**
 1       **4a**    **4a**
 2     **3a**      **3a**
 3       **4b**    **4b**
 4   **2a**        **2a**
 5       **4c**    **4c**
 6     **3b**      **3b**
 7       **4d**    **4d**
 8 **1b**          **1b**
 9       **4e**    **4e**
10     **3c**       **3c**
11       **4f**    **4f**
12   **2b**          **2b**
. . .
```

Рис. 2.6. **Графічний опис заповнення растра**

Термінатор GIF. Для того щоб забезпечити синхронізацію із закінченням файла зображення GIF, декодер GIF повинен обробляти закінчення режиму GIF по символі шістнадцятирічне 0×3B або ";", знайденому після закінчення обробки зображення. За згодою декодувальні програми повинні робити паузу і чекати дій, які вказують, що користувач готовий до продовження. Це може бути повернення каретки, уведення із клавіатури або клацання кнопкою миші. Для інтерактивних додатків ці дії користувача повинні бути передані в ядро програми як переведення каретки, для того, щоб обчислювальний процес міг тривати. Звичайно декодувальна програма залишає графічний режим і вертається до попереднього процесу.

Розширений блок GIF. Для того щоб забезпечити акуратне розширення визначення GIF, необхідний механізм для визначення впакування усередині потоку даних GIF. Значення розширення було визначено і документоване CompuServe для того, щоб передбачити керований спосіб удосконалень. Розширений блок GIF пакується способом, схожим на той, котрий використовувався для растрових даних, але не стискується. Основна структура блоку наведена на рис. 2.7.

Біти 7 6 5 4 3 2 1 0	Байти	
0 0 1 0 0 0 0 1	1	”!” – Ідентифікатор розширювального блоку.
Функціональний код	2	Розширений функціональний код (0-255).
Байт-лічильник	3	
Функціональні байти даних		Повторюється стільки разів, скільки необхідно.
… … …		
0 0 0 0 0 0 0 0		Нулевий байт-лічильник (термінатор блоку).

Рис. 2.7. **Розширений блок GIF**

Розширений блок GIF може безпосередньо передувати дескриптору зображення або перебувати перед термінатором GIF.

Всі декодери GIF повинні бути здатні розпізнавати присутність розширеного блоку GIF і потім читати його, якщо вони не можуть обробити функціональний код. Це гарантує, що старі декодери зможуть обробляти файли зображень GIF у майбутньому, хоча і без додаткових функціональних можливостей.

Формат TIFF. У TIFF конкретні поля ідентифікуються за допомогою унікального тегу. Це допускає присутність або відсутність конкретних полів у файлі залежно від вимог конкретного завдання.

TIFF-файл починається з 8-байтового заголовка файла (Image File Header), що вказує на одну або кілька директорій файла (Image File Directories). Директорії містять інформацію про зображення і покажчики на дані самого зображення. Слід розглянути ці структури більш докладно.

Заголовок файла (Image File Header – IFD). TIFF-файл починається з 8-байтового заголовка, що містить таку інформацію:

Байти 0 – 1. Перше слово файла визначає порядок байтів, використовуваний у файлі. Припустимими його значеннями є:

II (hex 4949),
MM (hex 4D4D).

У форматі II в 16-бітних і 32-бітних цілих числах порядок байтів завжди йде від молодших (менш значущих) до старшого (більше

значущого). У форматі MM для тих же чисел порядок байтів іде від старших до молодшого. В обох форматах символьні рядки запам'ятовуються як послідовність байтів у їх природному порядку.

Байти 2 – 3. Друге слово TIFF-файла – це номер версії. Це число, рівне 42 (2A hex), але воно не дорівнює номеру редакції поточної специфікації TIFF (у цьому випадку номер редакції поточної специфікації – це 5.0). Фактично номер версії TIFF (42) ніколи не міняється і, можливо, ніколи не зміниться. Але якщо це трапиться, то буде означати, що TIFF змінився настільки радикально, що програма читання TIFF повинна негайно припинити роботу. Число 42 було обрано через його глибокий філософський зміст. Воно може і повинне використовуватися для додаткової перевірки того, що це дійсно TIFF-файл.

TIFF-файли не мають явного номера редакції (тобто 5.0 – це поточна редакція). Це рішення при розробці було прийнято свідомо. У багатьох форматах поля можуть приймати різні значення залежно від номера версії. Проблема полягає в тому, що як тільки формат починає "старіти", зростають труднощі з документування того, які поля що означають у даній версії, і старі програми звичайно не здатні до роботи з файлами, що містять новий номер версії. Поля-TIFF мають постійне і визначене значення, тому що "старі" програми як правило можуть читати "нові" TIFF-файли. Останнє знижує вартість розробки програмного забезпечення та робить його більш надійним.

Байти 4 – 7. Це слово типу long, що містить зсув у байтах першої директорії файла (Image File Directory). Директорія може розташовуватися в будь-якому місці файла слідом за заголовком, але її початок повинен бути вирівняним на границю слова. Зокрема, директорія може випливати за даними зображення, що вона описує. Програми читання повинні просто переміщатися по цьому покажчику, поза залежністю від того, куди він указує. Термін байтовий зсув (byte offset) завжди використовується в цьому документі, щоб посилатися на положення відносно початку файла. Перший байт файла має зсув, рівний 0.

Директорії файла (Image File Directory – IFD) складаються з 2-байтового лічильника числа елементів (тобто числа тегів у даній директорії), слідом за яким розташована послідовність 12-байтових тегів і далі 4-байтовий зсув для наступної директорії або 0, якщо така відсутня. Не забувайте записувати 4 нульові байти наприкінці останньої директорії. Кожний 12-байтовий елемент IFD має такий формат:

Байти 0 – 1 містять Тег (Tag) поля.

Байти 2 – 3 містять Тип (Type) поля.

Байти 4 – 7 містять Довжину (Length) поля (тут, можливо, більш вдалим терміном є Count – лічильник).

Байти 8 – 11 містять Зсув для значення (Value Offset), тобто байтовий зсув того місця у файлі, де розташоване самозначення. Передбачається, що цей зсув повинен бути вирівняний на границю слова, тобто Value Offset повинен бути парним числом.

Цей зсув може вказувати на будь-яке місце у файлі.

Елементи в IFD повинні бути відсортовані в порядку зростання поля Tag. Помітимо, що цей порядок відмінний від того, у якому поля описані в даному документі. Значення, на які вказують елементи директорії, можуть випливати у файлі в будь-якому порядку.

Для економії часу та простору поле Value Offset інтерпретується як саме значення, а не як покажчик на значення, якщо значення вміщається в 4 байтах. Якщо значення менше 4 байтів, то воно вирівнюється по лівому краю 4-байтового поля, тобто запам’ятовується в байтах з молодшими номерами. Для того щоб визначити, вміщається чи ні значення в 4 байтах, варто перевірити значення полів Type і Length.

Поле Length описує дані в термінах типів даних, а не кількістю байтів у полі. Наприклад, одиночне 16-бітне слово (SHORT) має Length рівне 1, а не 2. Нижче наведені типи даних і їх довжини:

1 = BYTE – 8-бітне беззнакове ціле.

2 = ASCII – 8-бітні байти, які містять ASCII-коди, останній байт повинен бути нульовим.

3 = SHORT – 16-бітне (2-байтове) беззнакове ціле.

4 = LONG – 32-бітне (4-байтове) беззнакове ціле.

5 = RATIONAL – два числа типу LONG: перше становить чисельник дробу, друге – її знаменник.

Значення поля Length для даних типу ASCII включає нульовий байт. Якщо необхідне вирівнювання (наприклад, на границю слова), то поле Length не включає байти, що додаються при вирівнюванні. Відзначимо, що тут не потрібний байт-лічильник як у паскалівських рядках. Наявність поля Length робить його непотрібним. Строго говорячи, нульовий байт наприкінці рядків не є необхідним, але його присутність значно спрощує життя для програмістів, що пишуть на C.

Програми читання повинні перевіряти тип даних, щоб переконатися, що він такий, як вони очікують. У цей час TIFF допускає використання декількох типів даних для того самого поля. Наприклад, поля ImageWidth (ширина зображення) і ImageLength (довжина зображення) описані як ті, що мають тип SHORT. На деяких пристроях, що існують уже сьогодні, можливі дуже великі зображення, що мають більше 64K рядків або стовпчиків. Замість додавання паралельного LONG-тегу для цих полів, простіше допустити можливість використання типів і SHORT і LONG для поля ImageWidth і подібного йому.

Відмітимо, що у файлі може існувати біти IFD. Говорять, що кожний IFD визначає субфайл (subfile). Одна з потенційних можливостей використання послідовних субфайлів полягає в описі субзображень, кожне з яких пов'язане з головним, наприклад, є його версією зі зменшеною роздільною здатністю.

Формат JPEG. JPEG з'явився методом стиску, що дозволяє стискати дані повнокольорових багатоградаційних зображень із глибиною від 6 до 24 бітів/піксел з досить високою швидкістю та ефективністю. Сьогодні JPEG – це схема стиску зображень, що дозволяє досягти дуже високих коефіцієнтів стиску. Правда максимальний стиск графічної інформації, як правило, пов'язаний з певною втратою інформації. Тобто для досягнення високого ступеня стиску алгоритм так змінює вихідні дані, що одержуване після відновлення зображення буде відрізнятися від вихідного (стисливого). Цей метод стиску використовується для роботи з повнокольоровими зображеннями високої фотографічної якості. JPEG не був визначений як стандартний формат файлів зображень, однак на його основі були створені нові або модифіковані існуючі файлові формати.

Алгоритм обробки даних JPEG. Специфікація JPEG визначає мінімальні вимоги стандарту, які повинні підтримуватися всіма програмами, що використовують цей метод. JPEG заснований на схемі кодування, що базується на дискретних косинус-перетвореннях (DCT). DCT – це загальне ім'я визначеного класу операцій, дані про які були опубліковані кілька років назад. Алгоритми, що базуються на DCT, стали основою різних методів стиску. Ці алгоритми стиску базуються не на пошуці однакових атрибутів пікселів (як в RLE і LZW), а на різниці між ними.

У силу своєї природи вони завжди кодують із втратами, але здатні забезпечити високий ступінь стиску при мінімальних втратах даних. Схема JPEG ефективна тільки при стиску багатоградаційних зображень,

у яких розходження між сусідніми пікселями, як правило, досить незначні. Практично JPEG добре працює тільки із зображеннями, що мають глибину хоча б 4 або 5 бітів/піксел на колірний канал. Основи стандарту визначають глибину вхідного зразка в 8 бітів/піксел. Дані з меншою бітовою глибиною можуть бути оброблені за допомогою масштабування до 8 бітів/піксел, але результат для вихідних даних з низькою глибиною кольору може бути незадовільним, оскільки між атрибутами сусідніх пікселів будуть істотні розходження. За подібними причинами погано обробляються вихідні дані на основі кольорових таблиць, особливо якщо зображення представляється в розмитому вигляді.

Процес стиску за схемою JPEG включає ряд етапів (рис. 2.8):

Перетворення зображення в оптимальний колірний простір.

Субдискретизація компонентів кольоровості усередненням груп пікселей.

Застосування дискретних косинус-перетворень для зменшення надмірності даних зображення.

Квантування кожного блоку коефіцієнтів DCT із застосуванням вагових функцій, оптимізованих з урахуванням візуального сприйняття людиною.

Кодування результуючих коефіцієнтів (даних зображення) із застосуванням алгоритму Хаффмена для видалення надмірності інформації.

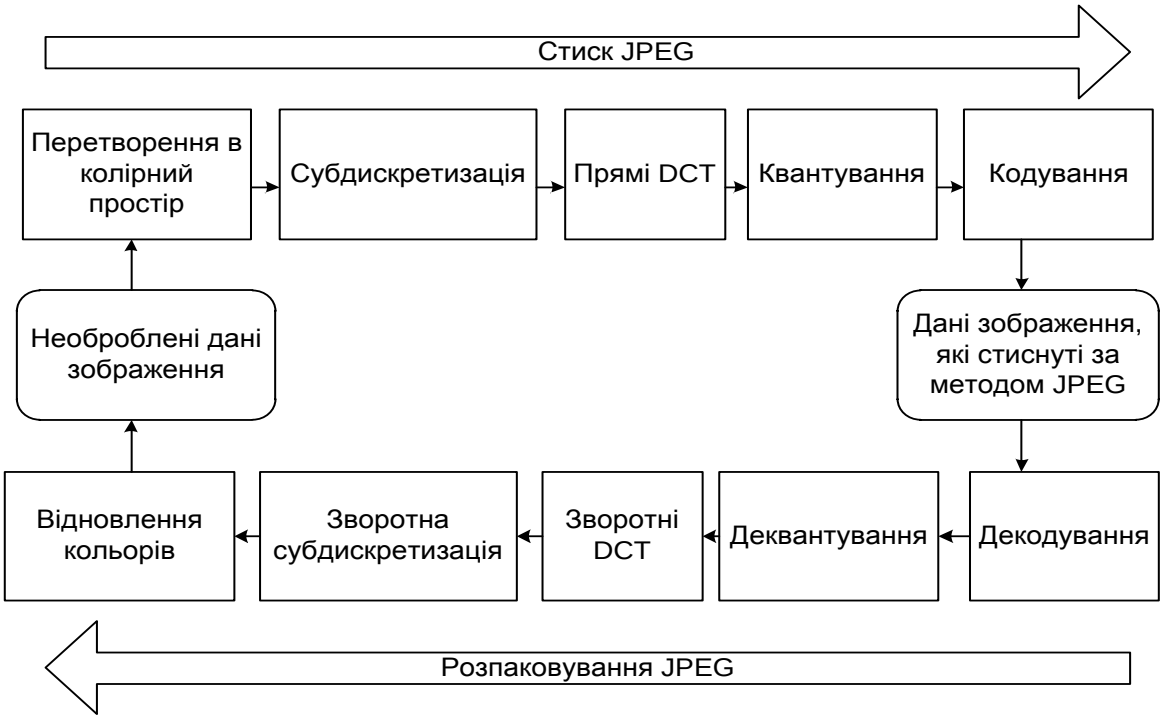


Рис. 2.8. **Структура JPEG-перетворень**

Розглянемо особливості кожного з перерахованих етапів. При цьому слід звернути увагу на те, що декодування JPEG здійснюється у зворотному порядку.

Колірний простір. У принципі алгоритм JPEG здатний кодувати зображення, засновані на будь-якому типі колірного простору. JPEG кодує кожний компонент колірної моделі окремо, що забезпечує його повну незалежність від будь-якої моделі колірного простору (наприклад, від RGB, HSI або CMYK).

У випадку застосування колірного простору яскравість/кольоровість, наприклад такого, як YUV або YCbCr, досягається кращий ступінь стиску. Компонента Y є інтенсивністю, а U і V – кольоровістю. Ця модель може бути переведена в RGB за допомогою перетворення без якої-небудь корекції насиченості. Для напівтонових зображень (у градаціях сірого) використовується тільки одна складова Y.

Перетворення колірної моделі RGB у модель YCbCr здійснюється за допомогою таких співвідношень:

$$Y = 0{,}299\ R + 0{,}587\ G + 0{,}114\ B;$$
$$Cb = -\ 0{,}1687\ R - 0{,}3313\ G + 0{,}5\ B + 128;$$
$$Cr = 0{,}5\ R - 0{,}4187\ G - 0{,}0813\ B + 128.$$

Зворотне перетворення моделі YCbCr у модель RGB здійснюється за допомогою подібних співвідношень:

$$R = Y + 1{,}402\ (Cr-128);$$
$$G = Y - 0{,}34414\ (Cb-128) - 0{,}71414\ (Cr-128);$$
$$B = Y + 1{,}772\ (Cb-128).$$

Субдискретизація компонентів кольоровості. На екрані комп'ютера практично ніколи не видно реально повнокольорових зображень реального світу. Це пояснюється обмеженими можливостями по цифровому поданню в пам'яті ПЕОМ, перекручуваннями при відтворенні кольору монітором і відеокартою. У результаті на моніторі ПЕОМ відтворюються, залежно від обраного відеорежиму, кольори, найбільш близькі до реального.

Більша частина візуальної інформації, до якої найбільш чутливі очі людини, складається з високочастотних, напівтонових компонентів яскравості (Y) колірного простору YCbCr. Дві інші складові кольоровості (Cb і Cr) містять високочастотну колірну інформацію, до якої око людини

менш чутливе. Отже, певна її частина може бути відкинута і, тим самим, можна зменшити кількість пікселей, що враховуються для каналів кольоровості. Наприклад, у зображенні розміром 1000×1000 пікселей можна використовувати яскравості всіх 1000×1000 пікселей, але тільки 500×500 пікселей для кожного компонента кольоровості. При такому поданні кожний піксел кольоровості буде охоплювати ту ж множину, що і блок 2×2 пікселі (для яскравості). У результаті зберігається для кожного блоку 2×2 усього 6 піксельних значень (4 значення яскравості та по 1 значенню для кожного із двох каналів кольоровості) замість того, щоб використовувати 12 значень при звичайному описі. Практика показала, що зменшення обсягу даних на 50 % майже непомітно відбивається на якості більшості зображень.

Однак у випадку загальноприйнятих колірних моделей типу RGB таке подання даних неможливе, оскільки кожний колірний канал RGB несе деяку інформацію яскравості та будь-яка втрата роздільної здатності досить помітна. Зменшення роздільної здатності каналів кольоровості шляхом субдискретизації, або усереднення груп пікселей здійснюється компресором JPEG.

Стандарт JPEG пропонує кілька різних варіантів визначення коефіцієнтів дискретизації, або відносних розмірів каналів субдискретизації. Канал яскравості завжди залишається з повною роздільною здатністю (дискретизація 1:1). Для обох каналів кольоровості звичайно виробляється субдискретизація 2 – у горизонтальному напрямі й 1:1 або 2:1 – у вертикальному. При цьому мається на увазі, що кольоровість пікселів буде охоплювати ту ж множину, що і блок 2×1 або 2×2 яскравості пікселей. Відповідно до термінології JPEG, ці процеси називаються 2h1v- і 2h2v – дискретизацією, відповідно.

Іншою загальноприйнятою специфікацією дискретизації 2hlv є 4:2:2, а дискретизації 2h2v – 4:2:0; остання специфікація пов’язана з телебаченням. Дискретизація 2hlv використовується досить широко, оскільки відповідає стандарту NTSC (Національний комітет з телевізійних стандартів США), однак при досить незначному виграші в якості вона забезпечує менший ступінь стиску, чим дискретизація 2h2v.

Колірні моделі RGB, HSI і CMY використовують у корисній візуальній інформації всі три колірних компоненти. Це утрудняє її вибіркове відкидання, а отже, і зменшує ступінь стиску даних. Напівтонові зображення не мають колірного простору і, отже, не мають потреби в перетворенні.

Оскільки засоби візуалізації та подання інформації є цифровими, у них споконвічно закладена певна система квантування сигналу. В остаточному підсумку рівень квантування визначається глибиною кольору, тобто кількістю кодових комбінацій, які використовуються для кодування кольору. Цей процес і називається квантуванням кольору.

Оскільки у квантованій матриці відсутня значна частка високочастотної інформації, наявної у вихідній матриці, перша часто стискується до половини свого первісного розміру або навіть ще більше. Реальні фотографічні зображення часто зовсім неможливо стиснути за допомогою методів стиску без втрат, тому 50 %-й стиск вважається досить хорошим.

Після конвертації графічних даних у колірний простір типу LAB, відкидається частина інформації про колір (залежно від конкретної реалізації алгоритму).

Спочатку в специфікаціях формату не була передбачена субтрактивна колірна модель CMYK. Фірма Adobe увела підтримку функції кольороподілу. Однак використання колірної моделі CMYK в JPEG для багатьох програм проблематично. Більш надійним рішенням уважається використання JPEG-стиску в EPS-файлах (Photoshop).

Сегментація зображення застосовується з метою розподілу його на два й більше сегменти (підзображення). Це полегшує буферизацію даних зображення в пам’яті ПЕОМ, прискорює їх довільну вибірку з диску і дозволяє зберігати зображення розміром понад 64x64 Кб. JPEG підтримує три типи сегментації зображень: просту, пірамідальну й комбіновану.

При простій сегментації зображення ділиться на два або більше сегменти фіксованого розміру. Всі прості сегменти кодуються зліва направо і зверху вниз, є суміжними і не перекриваються. Сегменти повинні мати однакову кількість вибірок і ідентифікаторів компонентів, і бути закодованими за однією схемою. Сегменти в нижній і правій частинах зображення можуть бути меншого розміру, ніж "внутрішні" сегменти, оскільки величина зображення не обов’язково повинна бути кратною розмірам сегмента.

При пірамідальній сегментації зображення також ділиться на сегменти, а кожний з них, у свою чергу, – на ще більш дрібні сегменти. При цьому використовуються різні рівні роздільної здатності. Моделлю такого процесу є сегментування піраміди зображення JPEG (JPEG Tiled Image Pyramid, JTIP), що відбиває процедуру створення пірамідального JPEG-зображення з декількома рівнями роздільної здатності.

У *схемі JTIP* послідовні шари одного зображення зберігаються з різною розподільною здатністю. Перше зображення, записуване на вершині піраміди, займає одну шістнадцяту частину встановленого розміру екрана й називається віньєткою. Застосовується воно для швидкого відтворення вмісту зображення. Це здобуває особливу значущість при роботі із програмами перегляду (браузерами). Наступне зображення займає одну четверту частину екрана й називається мажеткою. Звичайно вона використовується в тих випадках, коли на екрані необхідно одночасно відобразити два й більше зображення. Далі випливають повноекранне зображення з низькою розподільною здатністю, зображення з розподільною здатністю, що послідовно підвищується, і, нарешті, оригінал зображення.

При *пірамідальній сегментації* доцільний процес внутрішньої сегментації, коли кожний сегмент кодується як частина одного потоку JPEG-даних. Іноді може застосовуватися процес зовнішньої сегментації, при якому кожний сегмент становить окремо кодований потік JPEG-даних. Зовнішня сегментація прискорює доступ до даних зображення, полегшує його шифрування та поліпшує сумісність із деякими JPEG-декодерами.

Комбінована сегментація дозволяє зберігати і відтворювати версії зображень із декількома рівнями дозволу у вигляді мозаїки. Комбінована сегментація допускає наявність сегментів, що перекриваються, різних розмірів, з різними коефіцієнтами масштабування й параметрами стиску. Кожний сегмент кодується окремо й може комбінуватися з іншими сегментами без повторної дискретизації.

Наприклад, у випадку використання сегментів розміром 8×8 пікселів для кожного блоку формується набір чисел. Перші кілька чисел представляють колір блоку в цілому, у той час, як наступні числа відбивають більш тонкі деталі. Спектр деталей базується на зоровому сприйнятті людини, тому великі деталі більш помітні. На наступному етапі, залежно від обраного рівня якості, відкидається певна частина чисел, що представляють тонкі деталі. Таким чином, чим вище рівень компресії, тим більше даних відкидається й тим нижче якість зображення. Використовуючи JPEG, можна одержати файл у 1 500 разів менше, ніж BMP. Формат апаратно незалежний, повністю підтримується на PC і Macintosh.

Дискретне косінусне перетворення. Ключовим компонентом роботи алгоритму є дискретне косінусне перетворення. Дискретне

косінусне перетворення становить різновид перетворення Фур'є і, так само як і воно, має зворотне перетворення. Графічне зображення можна розглядати як сукупність просторових хвиль, причому осі *X* і *Y* збігаються із шириною і висотою картинки, а по осі *Z* відкладається значення кольору відповідного піксела зображення. Дискретне косинусне перетворення дозволяє переходити від просторового подання картинки до її спектрального подання та назад. Впливаючи на спектральне подання картинки, що складається з "гармонік", тобто відкидаючи найменш значущі з них, можна балансувати між якістю відтворення та ступенем стиску.

Формули прямого і зворотного дискретного косинусного перетворення наведені нижче. *Дискретне косинусне перетворення* (ДКП) перетворить матрицю пікселів у матрицю частотних коефіцієнтів відповідного розміру. Незважаючи на видиму складність, закодувати ці формули досить просто. Формула дискретного косинусного перетворення:

$$ДКП(i,j)=\frac{1}{\sqrt{2N}}C(i)C(j)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}pixel(x,y)\cos\left[\frac{(2x+1)i\pi}{2N}\right]\cos\left[\frac{(2y+1)i\pi}{2N}\right]$$

$$C(x)=\begin{cases}\frac{1}{\sqrt{2}}, x=0\\ 1, x>0\end{cases}.$$

Формула зворотного дискретного косинусного перетворення:

$$ДКП(i,j)=\frac{1}{\sqrt{2N}}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}C(i)C(j)ДКП(i,j)\cos\left[\frac{(2x+1)i\pi}{2N}\right]\cos\left[\frac{(2y+1)i\pi}{2N}\right]$$

$$C(x)=\begin{cases}\frac{1}{\sqrt{2}}, x=0\\ 1, x>0\end{cases}$$

У матриці коефіцієнтів, що вийшла, низькочастотні компоненти розташовані ближче до лівого верхнього кута, а високочастотні – праворуч і внизу. Це важливо тому, що більшість графічних образів на екрані комп'ютера складається з низькочастотної інформації. Високочастотні компоненти не так важливі для передачі зображення. Таким чином, дискретне косинусне перетворення дозволяє визначити, яку частину інформації можна безболісно викинути, не вносячи серйозних перекручувань у картинку.

Реалізація дискретного косинусного перетворення. Час, необхідний для обчислення кожного елемента матриці дискретного косинусного перетворення, сильно залежить від її розміру, тому що використовуються два вкладених цикли. Однією з особливостей є те, що практично неможливо виконати дискретне косинусне перетворення для всього зображення відразу. Як рішення цієї проблеми група розроблювачів JPEG запропонувала розбивати зображення на блоки розміром 8×8 точок.

Збільшуючи розміри блоку дискретного косинусного перетворення, можна домогтися деякого збільшення результатів стиску. Обмеження в коефіцієнті стиску пояснюються малою ймовірністю того, що вилучені на значну відстань точки зображення мають однакові атрибути.

За визначенням дискретного косинусного перетворення для його реалізації потрібно два вкладених цикли, і тіло циклів буде виконуватися Nx разів для кожного елемента матриці дискретного косинусного перетворення. Значно більш ефективний варіант обчислення коефіцієнтів дискретного косинусного перетворення реалізований через перемножування матриць. При такому підході формула дискретного косинусного перетворення може бути записана в такому вигляді:

$$ДКП = КП \cdot Точки \cdot К_{пт},$$

де *ДКП* – дискретне косінусне перетворення;

КП – матриця косінусного перетворення розміром Nx, елементи якої визначаються за формулою:

$$C(i,j) = \begin{cases} \frac{1}{\sqrt{N}}, i = 0 \\ \sqrt{\frac{2}{N}} \cos\left[\frac{(2j+1)i\pi}{2N}\right], i > 0 \end{cases},$$

Точка – матриця розміром Nx, що складається з пікселів зображення;

$К_{пт}$ – транспонована матриця КП.

При перемножуванні матриць "ціна" обчислення одного елемента результуючої матриці становлять *N* множень і *N* додавань, при обчисленні матриці дискретного косінусного перетворення – 2^x, відповідно. Порівняно з $O(N^x)$ це помітне підвищення продуктивності. Оскільки дискретне косінусне перетворення є різновидом перетворення Фур'є, то всі методи прискорення перетворення Фур'є можуть бути застосовані і у цьому випадку.

Округлення коефіцієнтів. Дискретне косінусне перетворення становить перетворення інформації без втрат і не здійснює ніякого стиску. Навпаки, дискретне косінусне перетворення підготовляє інформацію для етапу стиску із втратами або округлення.

Округлення є процесом зменшення кількості бітів, необхідних для зберігання коефіцієнтів матриці дискретного косінусного перетворення за рахунок втрати точності.

Стандарт JPEG реалізує цю процедуру через матрицю округлення. Для кожного елемента матриці дискретного косінусного перетворення існує відповідний елемент матриці округлення. Результуюча матриця виходить розподілом кожного елемента матриці дискретного косінусного перетворення на відповідний елемент матриці округлення і наступним округленням результату до найближчого цілого числа. Як правило, значення елементів матриці округлення зростають за напрямом зліва направо і зверху вниз.

Вибір матриці округлення. Від вибору матриці округлення залежить баланс між ступенем стиску зображення і його якістю після відновлення. Стандарт JPEG дозволяє використовувати будь-яку матрицю округлення, однак ISO розробила набір матриць округлення.

Матриця округлення будується за допомогою дуже простого алгоритму. Для того щоб визначити крок зростання значень у матриці округлення, задається одне значення в діапазоні [1; 13;16], називане фактором якості. Потім матриця заповнюється в такий спосіб:

```
for (i = 0; i < N; i++)
for (j = 0; j < N; j++)
Matrix[i][j] = 1 + (1 + i + j) * QualityFactor;
```

Фактор якості задає інтервал між сусідніми рівнями матриці округлення, розташованими на її діагоналях. Необхідно відзначити, що при таких значеннях матриці округлення коефіцієнт у матриці дискретного косінусного перетворення, розташований в осередку (7,7), повинен приймати значення не менше 16, щоб після округлення мати значення, відмінне від 0, і впливати на декадоване зображення. Таким чином, операція округлення є єдиною фазою роботи JPEG, де відбувається втрата інформації.

Обчислення при використанні методу DCT надзвичайно складні; фактично, це найбільш трудомісткий етап стиску JPEG. Виконавши його, практично розділяємо високочастотну та низькочастотну інформацію, з яких складається зображення. Після цього можна відкинути високо час-

тотні дані без втрати низькочастотних. Сам по собі етап перетворення DCT не передбачає втрат, за винятком помилок округлення.

Перш ніж відкинути певний обсяг інформації, компресор ділить кожне вихідне значення DCT на "коефіцієнт квантування", округляючи результат до цілого. Чим більше коефіцієнт квантування, тим більше даних губиться, оскільки реальне DCT-значення представляється усе менш і менш точно. Кожна з 64 позицій вихідного блоку DCT має власний коефіцієнт квантування. Причому терми більшого порядку квантуються з більшим коефіцієнтом, ніж терми меншого порядку. Крім того, для даних яскравості й кольоровості застосовуються окремі таблиці квантування, що дозволяють квантувати дані кольоровості з більшими коефіцієнтами, чим дані яскравості. Таким чином, JPEG використовує різну чутливість ока до яскравості й кольоровості зображення.

На цьому етапі більшість JPEG-компресорів управляються за допомогою установки якості. Компресор використовує убудовану таблицю, розраховану на середню якість, і нарощує або зменшує значення кожного елемента таблиці назад пропорційно необхідній якості.

Застосовувані таблиці квантування записуються в стислий файл, щоб декомпресор знав, як відновити коефіцієнти DCT (приблизно).

Вибір відповідної таблиці квантування є "високим мистецтвом". Більшість існуючих компресорів використовують таблицю, розроблену Комітетом JPEG ISO. Можливо, згодом будуть запропоновані таблиці, що дозволяють здійснювати стиск більш ефективно й при збереженні якості зображення.

Кодування. Заключна стадія роботи кодера JPEG – це власне кодування. Воно включає три дії над округленою матрицею дискретного косінусного перетворення, для того, щоб підвищити ступінь стиску.

Перша дія – це заміна абсолютного значення коефіцієнта, розташованого в осередку (0,0) матриці, на відносне. Оскільки сусідні блоки зображення значною мірою "схожі" один на одного, то кодування чергового (0,0) елемента як різниці з попереднім дає менше значення.

Коефіцієнти матриці дискретного косінусного перетворення обходяться зиґзаґом. Після чого нульові значення кодуються з використанням алгоритму кодування повторів (RLE), а потім результат обробляється за допомогою "кодування ентропії", тобто алгоритмів Хаффмана або арифметичного кодування, залежно від реалізації.

"Кодування ентропії". Результатом роботи спрощеної схеми кодування є трійки такого вигляду:

<Кількість нулів, Кількість бітів, Коефіцієнт>,

де Кількість нулів – кількість повторюваних нулів, що передують поточному (ненульовому) елементу матриці дискретного косинусного перетворення;

Кількість бітів – кількість бітів, що випливають далі і кодують значення коефіцієнта;

Коефіцієнт – значення ненульового елемента матриці дискретного косинусного перетворення.

Відповідність між полями кількості бітів і коефіцієнтів наведена в табл. 2.2.

Таблиця 2.2

Відповідність між полями кількості бітів і коефіцієнтів

Кількість бітів	Коефіцієнт
1	[-1, 1]
2	[-3, -2], [2, 3]
3	[-7, -4], [4, 7]
4	[-15, -8], [8, 15]
5	[-31, -16], [16, 31]
6	[-63, -32], [32, 63]
7	[-127, -64], [64, 127]
8	[-255, -128], [128, 255]
9	[-511, -256], [256, 511]
10	[-1023, -512], [512, 1023]

Таке кодування не настільки ефективне, як кодування Хаффмана, але на певних даних воно дає аналогічні результати.

Після завершення цього етапу потік даних JPEG готовий до передачі по комунікаційних каналах або інкапсуляції у формат файла зображення.

Формат JFIF. JPEG позначає розглянутий вище алгоритм стиску, а не конкретний формат подання графічної інформації. Практично будь-яку графічну інформацію можна стиснути за таким алгоритмом. Формат файлів, що використовують алгоритм JPEG, формально називають JFIF (JPEG File Interchange Format). На практиці дуже часто файли, що використовують JPEG-стиск, називають JPEG-файлами.

На основі JPEG-методу стиску побудовані численні формати, наприклад, формат TIFF/JPEG, відомий як TIFF 6.0, TIFF, QuickTime та ін.

*Файли із графікою у форматі JPEG мають розширення *.jpg*.

Формат JPEG є TrueColor-форматом, тобто може зберігати зображення із глибиною кольору 24 біти/піксел. Такої глибини кольору досить для практично точного відтворення зображень будь-якої складності

на екрані монітора. У випадку перегляду кольорового зображення на екрані монітора більша глибина кольору (наприклад, 32 біти/піксел) практично не відрізняється від зображення із глибиною кольору в 24 біти/піксел. Той же результат спостерігається й при роздруківці зображення на більшості доступних принтерів. Глибина кольору в 32 біти/піксел, як правило, використовується у видавничій діяльності.

JPEG має більш високий ступінь стиску зображень, ніж GIF, але не має можливість зберігати кілька зображень в одному файлі. Правда, відома модифікація формату JPEG, що одержала назву Progressive JPEG, яка призначена для тих же завдань, що і міжрядкове відображення GIF-зображень. Це зробило формат JPEG більш привабливим як мережний стандарт.

JPEG орієнтований, насамперед, на реалістичні зображення, тобто зображення фотографічної спрямованості, і якість стиску значно погіршується при обробці зображень із чітко обкресленими лініями й границями кольорів.

Алгоритм JPEG і побудовані на його основі формати передбачають реалізацію функціональної можливості, що одержала назву етикетки. Фактично це зменшена копія зображення. Етикетку можна розглядати як свого роду аналог запропонованого у форматі GIF прийому міжрядкового розгорнення зображення. Тобто при наявності великої кількості файлів JPEG можна виводити їх на екран у вигляді етикеток, що дозволяє відобразити їх досить швидко або у великій кількості (списком) і, тим самим, дати користувачеві подання про вміст кожного файлу. Етикетки можуть бути закодовані методом JPEG; збережені у форматі 1 байт/піксел (тобто у вигляді напівтонового зображення) або представлені у вигляді повнокольорового зображення з 16,7 млн кольорів (24 біти/піксел).

JPEG може розглядатися як набір методів стиску зображень, придатних для задоволення потреб користувача. JPEG може настроюватися на відтворення дуже маленьких, стислих зображень з поганою якістю, але прийнятих для необхідних цілей. У той же час він дозволяє робити стиск зображень дуже високої якості, обсяг даних яких набагато менше, ніж в оригінальних незжатих даних.

JPEG, як правило, супроводжується втратами. Схеми стиску JPEG засновані на відкиданні інформації, що важко помітити візуально. Відомо, що невеликі зміни кольору погано розпізнаються оком людини.

Схема JPEG була спеціально розроблена для стиску кольорових і напівтонових багатоградаційних зображень фотографій, телевізійних

заставок, іншої складної графіки. Кінцевий користувач може "відрегулювати" якість кодувальника JPEG, використавши параметр, що іноді називають установкою якості, або *Q*-фактором. Різні реалізації даного методу мають різні діапазони *Q*-фактора, але типовим вважається від 1 до 100. При значенні фактора, рівному 1, створюється стисле зображення найменшого розміру, але поганої якості; при значенні фактора, рівному 100, можна одержати стисле зображення більшого розміру, але й кращої якості. Оптимальне значення Q-фактора залежить від умісту зображення й, отже, підбирається індивідуально. Особливим мистецтвом при стиску JPEG є вибір мінімального значення Q-фактора, що дозволяє створити зображення прийнятної якості і найбільш близьке до оригіналу.

Поряд з вищесказаним необхідно відзначити, що графічна анімація, чорно-білі ілюстрації, документи, а також типова векторна графіка, як правило, JPEG стискуються погано.

У цей час JPEG стали використовувати для стиску відеоінформації, однак авторам не відомі отримані результати.

Формат JPEG одержав велике поширення в Web-публікаціях для подання графічних елементів Web-сторінки, у тих випадках, коли потрібно багатобарвне якісне зображення.

2.3. Основні етапи алгоритму стиску зображень JPEG. Атаки на стеганосистеми із застосуванням JPEG

Під стиском розуміється зменшення числа бітів, що вимагаються для цифрового подання зображень. В основі стиску лежать два фундаментальних явища: зменшення статистичної і психовізуальної надмірності. Можна виділити три типи статистичної надмірності:

просторова, або кореляція між сусідніми пікселами;

спектральна, або кореляція між сусідніми частотними смугами;

часова, або кореляція між сусідніми кадрами (для відео).

Чи велика статистична надмірність у нерухливому зображенні? Для відповіді на це запитання спробуйте стиснути картинку яким-небудь архіватором – результати вас розчарують. Високі коефіцієнти стиску досяжні лише з використанням психовізуальної надмірності зображення, тобто зневаги його візуально незначущими частинами. І отут уже не обійтися без знання системи людського зору. "Викинуті" частини зображення заміняють нулями (константами), і якщо їх багато – застосовують кодер довжин серій. У реальних алгоритмах стиску здійснюють

обнуління не пікселів зображення, а спектральних коефіцієнтів. Перевага такого підходу полягає в тому, що близькі до нуля спектральні коефіцієнти мають тенденцію розташовуватися в заздалегідь передбачуваних областях, що приводить до появи довгих серій нулів і підвищення ефективності кодування. Більші за величиною коефіцієнти ("значущі") піддають більш-менш точному квантуванню й також стискають кодером довжин серій. Останнім етапом алгоритму стиску є застосування ентропійного кодера (Хаффмана або арифметичного).

Відновлене після стиску зображення, природно, відрізняється від вихідного. За інших рівних умов, чим більше стиск, тим більше перекручування. Для оцінки якості відновленого зображення можна використовувати міру середньоквадратичного перекручування, обумовлену як:

$$СКП = \frac{1}{N}\sum_{i=1}^{N}\left(x_i - \hat{x}_i\right)^2, \qquad (2.1)$$

де N – кількість пікселів у зображенні: $x_i, \hat{x}_i$ – значення пікселів вихідного і відновленого зображень. Набагато частіше застосовується модифікація цієї засоби – пікове відношення сигнал/шум, обумовлене як:

$$ПВСШ = 10\log_2 \frac{N255^2}{\sum_{i=1}^{N}\left(x_i - x_i\right)^2}, \qquad (2.2)$$

де 255 – максимальне значення яскравості напівтонового зображення (тобто 8 бітів/піксел). Відновлене зображення вважається прийнятним, якщо $ПВСШ \geq 28 - 30$ дБ (у середньому). Перераховані об'єктивні засоби перекручування не завжди корелюють із суб'єктивним сприйняттям зображень, однак нічого кращого дотепер не придумано.

ПВСШ не завжди добре поводиться з візуально спостережуваною помилкою. Нехай є два зображення, які повністю однакові, крім невеликої множини. Хоча візуально різниця між цими зображеннями добре помітна, ПВСШ буде приблизно однаковим. Облік системи людського зору в схемі стиску є важким завданням. Було проведено множину досліджень, але в силу труднощів з математичним описом системи зору людини більш прийнятної міри знайдено не було.

Вище було показано, що в людському оці виконується операція короткомасштабного подання зображень. Око більш чутливе до перекручувань у низькочастотній множині. Звідси існує можливість поліпшення візуальної якості реконструйованого зображення шляхом звуження середньоквадратичнього відхілення (СКВ) субсмуг відповідно до чутливості ока в різних частотних діапазонах.

Процес впровадження приховуваної інформації в зображення в якомусь сенсі дуалний процесу їх стиску. Вбудовування інформації найчастіше здійснюють у незначні множини, щоб не змінити візуальне подання зображення. Оптимальний метод стиску видалить цю інформацію. На щастя, сучасні алгоритми стиску залишають досить можливостей для реалізації витончених способів упровадження даних.

Розглянемо деякі алгоритми стиску зображень.

Стандарт стиску JPEG є в цей час найпоширенішим і своєрідним benchmark'ом для алгоритмів цифрових відеозображень (ЦВДЗ), тобто стійкість системи ЦВДЗ до стиску JPEG перевіряється звичайно в першу чергу. Відповідно до цього стандарту зображення розбивається спочатку на блоки 8×8 елементів, до кожного з яких застосовується дискретне косинусне перетворення (ДКП). Призначенням ДКП є здійснення перерозподілу енергії: значущі коефіцієнти групуються в лівому верхньому куті квадрата спектральних коефіцієнтів, тому що сусідні піксели зображення корелбовані. Далі випливають рівномірні табличні квантування коефіцієнтів, кодування довжин серій і кодування Хаффмана.

В останні роки увагу фахівців у множині ефективного кодування притягнуто до стиску зображень із застосуванням вейвлет-перетворення. У даному напрямі ведуться активні дослідження й уже отримані перші результати, що показують ефективність застосування вейвлет-перетворення для стиску зображень. Розроблено велику кількість алгоритмів стиску з використанням цього перетворення.

Вейвлет-перетворення, як і ДКП, перерозподіляє енергію зображення. Ця компактність енергії веде до ефективного застосування скалярних квантователей. Однак вони не враховують залишкову структуру, що зберігається у вейвлет-коефіцієнтах, особливо високочастотних субсмуг. Сучасні алгоритми стиску всі тим або іншим способом використовують цю структуру для підвищення ефективності стиску.

Одним з найбільш природних способів є облік взаємозв'язків між коефіцієнтами з різних субсмуг. У високочастотних субсмугах є звичайно більші множини з нульовою або малою енергією. Множини з високою енергією повторюють від субсмуги до субсмуги свої обриси й місце розташування. І це не дивно – адже вони з'являються навколо контурів у вихідному зображенні – там, де вейвлет-перетворення не може адекватно представити сигнал, що приводить до "витоку" частини енергії у високої частоти (ВЧ) субсмуги. Повільно змінюючись, гладкі множини вихідного

зображення добре описують низької частоти (НЧ) вейвлет-базиси, що приводить до "упакування" енергії в малому числі коефіцієнтів НЧ множини. Цей процес приблизно повторюється на всіх рівнях декомпозиції, що і приводить до візуального "подібності" різних субсмуг (рис. 2.9).

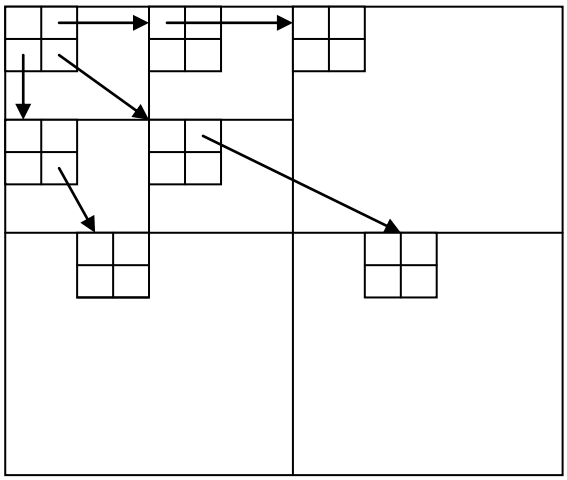


Рис. 2.9. **Залежності між коефіцістами вейвлет-перетворення зображення, які використовуються в алгоритмі нуль-дерева**

Отже, апріорне знання того, що зображення складається із гладких галузей, текстур і контурів, допомагає враховувати цю міжсмужну структуру. Кодери, що використовують структуру нуль-дерева, сполучать облік структури коефіцієнтів зі спільним кодуванням нулів, у результаті чого виходить дуже ефективний алгоритм стиску.

Вперше ідея нуль-дерева була запропонована в роботі [42]. В алгоритмі застосовувалася деревоподібна структура даних для опису вейвлет-коефіцієнтів (див. рис. 2.9).

Така структура виходить у результаті застосування двоканального роздільного вейвлет-перетворення. Кореневий вузол дерева представляє коефіцієнт функції, що масштабує, в самій НЧ множині й має три нащадки. Вузли дерева відповідають вейвлет-коефіцієнтам масштабу, рівного їх висоті в дереві. Кожний з вузлів має чотири нащадки, що відповідають вейвлет-коефіцієнтам наступного рівня й того ж просторового розташування. Низом дерева є листові вузли, що не мають нащадків.

Для кожного з коефіцієнтів самої НЧ множини існує три таких дерева, що відповідають трьом порядкам фільтрації.

Квантування нуль-деревом засноване на спостереженні, що якщо коефіцієнт малий, його нащадки на дереві найчастіше теж малі.

Це пояснюється тим, що значущі коефіцієнти виникають поблизу контурів і текстур, які локальні. Неважко побачити, що це є різновидом пророкування. Можна припустити, що якщо який-небудь коефіцієнт незначущий, то всі його нащадки також будуть незначущими. Дерево або субдерево, що містить (принаймні, так передбачається) тільки незначущі коефіцієнти, називається нуль-деревом.

У роботі [42] був запропонований такий алгоритм квантування вейвлет-коефіцієнтів. Спочатку кожний вузол квантується квантователем, оптимальним для щільності розподілу Лапласа. Якщо значення вузла менше деякого порога, його нащадки ігноруються. Ці нащадки будуть відновлені декодером як нулі. Інакше здійснюється перехід до чотирьох нащадків вузла, і процедура повторюється. Якщо вузол не має нащадків (є аркушем), починає оброблятися наступний кореневий вузол і т. д.

Даний алгоритм є ефективним у силу двох причин. По-перше, у силу гарного "упакування" енергії вейвлет-перетворенням і, по-друге, за рахунок спільного кодування нулів. Для кодування нулів звичайно застосовується кодер довжин серій. Для підвищення ефективності на вхід цього кодера коефіцієнти повинні подаватися у визначеному порядку. Наприклад, в JPEG застосоване зиґзаґоподібне сканування. Напевно, найбільш важливим внеском цієї роботи була демонстрація того, що область вейвлет-коефіцієнтів прекрасно пристосована для роботи кодера довжин серій. Справді, генеруються більші серії нулів і не треба передавати їх довжину, тому що висота дерева відома. Аналогічно JPEG, даний алгоритм є різновидом скалярного/векторного квантування. Кожний (значущий) коефіцієнт квантується окремо, а символи, що відповідають малим коефіцієнтам, утворять вектор. Цей вектор складається із символу нуль-дерева й послідовності нулів довжиною до кінця дерева.

У більшості алгоритмів стиску зображень на основі вейвлет-перетворення є можливість виділити дві складові швидкості й дві складові перекручування. В алгоритмах виконується оптимізація розподілу бітів між цими складовими з урахуванням обмеження на загальну швидкість кодування зображення.

Одна зі складових пов’язана з "обнуліннням" коефіцієнтів, що не перевершують деякий поріг, інша – із квантуванням більших коефіцієнтів ("значущих") і передачею їх місця розташування. Ефективність алгоритму стиску залежить від правильного визначення порога ухвалення рішення про значущість коефіцієнтів, а також від обраного способу

квантування значущих коефіцієнтів і від методу передачі інформації про їх місце розташування.

Для передачі інформації про позиції значущих коефіцієнтів відомий винятково ефективний алгоритм "вкладеного нуль-дерева" (EZW) [59], а також його різновиду – SPIHT [54] та ін.

Стандарт JPEG добре придатний для стиску зображень в 30 – 40 разів. При більш сильному стиску якість різко падає. Ця й багато інших причин послужили приводом до розробки нового стандарту на стиск зображень – JPEG-2000. У новому стандарті реалізовані такі опції, як послідовна передача, кодування конкретного блоку, що цікавить, зображення, його масштабованість, захищеність від помилок передачі, довільний доступ до стислого зображення. У *стандарті JPEG-2000* як первинне перетворення застосовується вейвлет-перетворення. Вейвлет-коефіцієнти піддаються квантуванню за алгоритмом, відомим як "ієрархічне кодування блоків з оптимізованим усіканням" (EBCOT), запропонованим в роботі [31]. Основна відмінність цього алгоритму від EZW і SPIHT полягає в тому, що EBCOT працює з незалежними блоками, що не перекриваються, які кодуються ітеративно. У такий спосіб замість структури даних нуль-дерева тут використовується структура квадро-дерева. У результаті виходить багаторівневий легко масштабований потік бітів. Кожний рівень відповідає якомусь ступеню перекручування. Розподіл бітів між рівнями здійснюється рішенням оптимізаційної завдання із застосуванням методу множників Лагранжа [60].

У стеганографії використовується багато ідей з множини компресії зображень. Крім того, знання алгоритмів стиску відео допомагає конструювати до цих алгоритмів ЦВДЗ.

Навіть за умови неможливості виділення й читання прихованого повідомлення, факт наявності останнього можна відносно легко виявити. Ще більш простою є операція знищення даного повідомлення. Наприклад, якщо повідомлення приховане у файлі BMP-формату мето-дом заміни палітри, то вплив на цей файл випадковою заміною кольорів у палітрі зробить повідомлення, що невитягається, іншими словами – знищить його.

Під час проектування й дослідження стеганографічних систем особлива увага повинна бути приділена вивченню впливу на них активних і зловмисних атак. Активні атаки здатні змінити контейнер під час зв’язку: порушник може перехопити стеганограму, що була послана переда-вальною стороною до приймаючої, змінити її (шляхом певної обробки контейнера) і відправити результат приймаючій стороні. У цьому випадку

передбачається, що при активній атаці неможливо повністю замінити контейнер і його семантику, а можна тільки внести незначні зміни таким чином, щоб оригінал і змінений контейнер залишилися візуально і семантично подібними.

Стеганографічні системи надзвичайно чутливі до модифікації контейнера (наприклад, для зображення – це згладжування і фільтрація, для звуку – фільтрація і переквантування відліків). Так, простий стиск із втратами може привести до повної втрати інформації, оскільки при цьому витягають незначні компоненти сигналу і цим знищується прихована в них секретна інформація.

У сучасних комп'ютерних системах реалізуються стеганографічні перетворення з високою надмірністю, стійкі до трансформації контейнера (поворот, масштабування, друк з наступним скануванням і т. д.), тому одна з важливих вимог до прикладної стегасистеми – це забезпечення стійкості до випадкових або навмисних атак.

2.4. Стійкість стеганосистеми до активних атак

Виходячи з розглянутих вище особливостей атак на стеганосистеми, можна зробити висновок, що протидія статистичному стеганоаналізу повинна полягати в побудові математичних моделей сигналів-контейнерів, пошуці на їх підставі "дозволених" для модифікації галузей і вбудовуванні в них приховуваної інформації, статистика якої не відрізняється від статистики контейнера. Така нерозрізненість визначає стійкість стеганосистеми.

Як і для криптографічних систем захисту інформації, безпека стеганосистем описується й оцінюється їх стійкістю (стеганографічною стійкістю), однак визначення стійкості та злому даних систем різні. У криптографії система захисту інформації є стійкою, якщо, володіючи перехопленою криптограмою, зловмисник не здатний витягти інформацію, що втримується в ній.

Неформально визначимо, що стеганосистема є стійкою, якщо зловмисник, спостерігаючи за інформаційним обміном між відправником і одержувачем, нездатний виявити, що під прикриттям контейнерів передаються приховувані повідомлення, і тим більше довідатися про зміст останніх. У більш широкому змісті, під *стійкістю стеганосистем* розуміється їх здатність приховувати від кваліфікованого зловмисника факт прихованої передачі даних, здатність протистояти спробам

зловмисника зруйнувати, спотворити, видалити приховано передані повідомлення, а також можливість підтвердити або спростувати автентичність приховано переданої інформації.

Стеганосистема є стійкою до активних атак, якщо приховувана з її допомогою інформація не може бути замінена без значних змін контейнера, у результаті яких останній втратить свою функціональність [5].

Визначення 2.1. Нехай Σ — стеганографічна система та φ – клас відображень $C \to S$. Тоді система Σ буде φ-стійкою, якщо у випадку стеганосистем із секретним ключем для всіх $f \in \varphi$ справедливо

$$D\{f[E(c,m,k),k]\} = D[E(c,m,k),k] = m, \qquad (2.3)$$

а у випадку безключових стеганосистем, незалежно від вибору $m \in M, c \in C$ та $k \in K$:

$$D\{f[E(c,m,k),k]\} = D[E(c,m,k),k] = m. \qquad (2.4)$$

Очевидно, що існує і зворотний взаємозв'язок між надійністю стеганосистеми і її стійкістю: чим більш стійкою до модифікації контейнера буде стеганосистема, тим вона буде менш надійною, оскільки стійкість може бути досягнута завадостійким кодуванням, що може привести до істотних перекручувань контейнера й, можливо, до зміни ймовірності P_s.

Багато стеганосистем стійкі тільки до визначеного класу відображень (ущільнення із втратами, геометричні перетворення, фільтрація, переквантування відліку, адитивний білий шум, перетворення ЦАП→АЦП і т. д.). Ідеальна стеганосистема повинна бути стійкою до всіх відображень типу "збереження α-подоби", тобто відображення виду $f : C \to S$ із властивістю $sim[c, f(c)] \to \lambda$ та $\lambda \approx 1$ [5; 11]. Однак такі системи складні в проектуванні і, через необхідність додаткового застосування завадостійкого кодування, мають занадто низьку пропускну здатність. З іншого боку, система є "λ-слабкою", якщо для кожного контейнера існує таке відображення "збереження-подоби", що прихована інформація буде невідновлюваною з погляду співвідношень (2.3) або (2.4).

У загальному випадку існує два підходи до створення стійких стеганосистем [11; 35]:

передбачаючи можливі атаки на стеганограми з боку порушників, стеганографічне перетворення споконвічно проектується стійким до знищення прихованих даних певним класом модифікацій;

використовуються перетворення, які мають властивість оборотності до можливих модифікацій з метою відновлення початкового вигляду стеганограми. Так, у роботі [5] запропонований метод "афінного кодування" для протидії афінним перетворенням зображення. При цьому передбачається оцінка параметрів перетворень, вимір змін форми, розмірів деяких кодованих образів.

Стійкі алгоритми повинні приховувати дані в найбільш істотних фрагментах контейнера, оскільки інформація, що кодується в шумовому компоненті, легко може бути витягнута або зруйнована. Наприклад, відомо [5; 16], що стеганографічні перетворення, які працюють із частотною областю контейнера, переважно більш стійкі до модифікацій порівняно з алгоритмами, що працюють у просторовій (для зображення) або часовій (для звуку) областях. Використовуючи ці властивості, можна створити стійкі стеганосистеми, які, наприклад, будуть зберігати приховувану інформацію в коефіцієнтах дискретного косинусного перетворення (ДКП) зображення.

2.5. Приховування даних у просторій множині зображень. Методи приховування в найменш значущому біті даних

Метод заміни найменш значущого біта (НЗБ, LSB – Least Significant Bit) найпоширеніший серед методів заміни в просторовій множині. Молодший значущий бітів зображення несе в собі найменше інформації. Відомо, що людина в більшості випадків не здатна помітити змін у цьому біті. Фактично, НЗБ – це шум, тому його можна використовувати для вбудовування інформації шляхом заміни менш значущих бітів пікселей зображення бітами секретного повідомлення. При цьому для зображення в градаціях сірого (кожний піксель зображення кодується одним байтом) обсяг убудованих даних може становити 1/8 від загального обсягу контейнера. Наприклад, у зображення розміром 512×512 можна вмонтувати ~32 кбайт інформації. Якщо ж модифікувати два молодших біти (що також практично непомітно), то дану пропускну здатність можна збільшити ще вдвічі (додаток А).

Популярність даного методу обумовлена його простотою й тим, що він дозволяє приховувати у відносно невеликих файлах досить великі обсяги інформації (пропускна здатність створюваного прихованого каналу зв’язку становить при цьому від 12,5 до 30 %). Метод найчастіше

працює з растровими зображеннями, представленими у форматі без компресії (наприклад, GIF і BMP) [41].

Метод НЗБ має низьку стеганографічну стійкість до атак пасивного і активного порушників. Основний його недолік – висока чутливість до найменших перекручувань контейнера. Для ослаблення цієї чутливості часто додатково застосовують завадостійке кодування.

Перед імпортом зображення-контейнера в документ MathCAD його необхідно підготувати у відповідному редакторі та записати у вигляді файла в поточний (для формованого документа MathCAD) каталог (слід зазначити, що для того, щоб уникнути можливих проблем з підтримкою кирилиці, бажано, щоб адреса розміщення файла на диску, як власне, і ім'я файла, складалися з латинських символів). MathCAD підтримує формати BMP, JPEG, GIF, PCX і TGA. Як було зазначено вище, формати BMP і GIF дозволяють зберігати зображення практично без втрати їх якості та тому більш придатні в ролі носіїв інформації.

Крок 1. Розглянемо структуру BMP-файла: він містить точкове (растрове) зображення і складається із трьох основних розділів: заголовка файла, заголовка растра та растрових даних.

Заголовок файла містить інформацію про файл (його тип, обсяг і т. п.). У заголовок растра винесена інформація про ширину й висоту зображення, кількість бітів на піксель, розмір растра, глибину кольору, коефіцієнт компресії і т. д.

У першу чергу цікавять растрові дані – інформація про колір кожного пікселя зображення. Колір пікселя визначається об'єднанням трьох основних колірних складових: червоної, зеленої та синьої (скорочено, RGB). Кожній з них відповідає своє значення інтенсивності, що може змінюватися від 0 до 255. Отже, за кожний з колірних каналів відповідає 8 бітів (1 байт), а глибина кольору зображення в цілому – 24 біти (3 байти).

Імпорт графічного файла виконується операцією Picture з позиції Insert головного меню програми. У модулі, що при цьому з'явився, необхідно заповнити шаблон даних у лівому нижньому куті, для чого в подвійних лапках варто ввести ім'я файла (або ж, при необхідності, – повний шлях його розміщення на диску) і нажати клавішу <Enter>.

Взагалі, формати BMP і GIF використовують алгоритми компресії, але ці алгоритми найпростіші, що дозволяє зберігати зображення практично без втрати його якості. Застосовується функція READRGB

("ім’я файла"), що повертає масив із трьох підмасивів, які, у свою чергу, несуть інформацію про розкладання кольорового зображення на колірні компоненти R, G і B:

C:= READRGB("C.bmp").

При цьому три колірних компоненти розміщаються один за іншим у загальному масиві C.

Для виділення колірних складових можна використовувати убудовані функції виділення відповідних колірних компонентів, кожна з яких повертає масив, що відповідає визначеному колірному компоненту графічного файла:

R := READ_RED("C.bmp"); G := READ_GREEN("C.bmp");
B := READ_BLUE("C bmp").

Крок 2. Текст повідомлення збережемо у файлі M.txt каталогу, для формованого документа MathCAD, у такому форматі:

тип файла – звичайний текст (*.txt);

кодування – кирилиця (Windows) .

Крок 3. Існує можливість приховання файлів будь-якого формату. Єдина умова, що при цьому повинна виконуватися, – вибір файла-контейнера належного обсягу (наприклад, орієнтовне співвідношення між обсягами файла-контейнера й файла-повідомлення для методу НЗБ – 8:1).

Імпорт текстового повідомлення можна виконати за допомогою функції READBIN "ім’я_файла", "тип_формату_даних". У цьому випадку дані представлені як 8-бітне беззнакове ціле число (байт):

M:= READBIN("M.txt", "byte").

Результат обчислення даного виразу – матриця-стовпець (вектор), кожний елемент якої відповідає розширеному ASCII-коду відповідного символу (букви) імпортованого повідомлення. У десятковому вигляді коди символів можуть приймати значення від 0 до 255; у двійковому вигляді для цього досить використовувати 8 бітів на один символ – так зване однобайтове кодування, на що вказує параметр byte як аргумент функції READBIN.

Необхідно відзначити, що за замовчуванням нижня границя індексації масивів дорівнює 0. У прикладах даного посібника індексація

починається з 1 (якщо не буде застережене інше), що, зокрема, можна встановити за допомогою оператора ORIGIN := 1 на початку документа або ввести 1 у поле Array Origin на вкладці Built-in Variables діалогового вікна Worksheet Options, що викликається з меню Tools системи MathCAD.

Перевірку імпортування файла повідомлення (якщо як повідомлення використовується звичайний текст) можна виконати за допомогою виклику функції vec2str(M). Ця функція повертає рядок символів, що відповідають вектору M ASCII-кодів.

Крок 4. Для того щоб при розпакуванні контейнера з отриманої множини символів можна було чітко визначити початок і кінець саме прихованого повідомлення, доцільно ввести відповідні секретні мітки, які обмежували б цей корисний зміст.

Мітки повинні складатися з достатньої кількості символів, щоб не приймати за мітки символи випадкового утворення. Крім того, для зменшення ймовірності виявлення міток при проведенні стеганоаналізу бажано, щоб коди цих символів були досить рознесені на ASCII-осі (наприклад, використовувати поряд з латинськими символами символи кирилиці і службових символів – так звана транслітерація; використання псевдовипадкових послідовностей кодів символів і т. п.). Нехай мітки мають такий вигляд:

$$\mu_S = "n0ч@m0k" \qquad \mu_E = "KIHeu,6"$$

Обмежуючі мітки додаємо в текст закодованого повідомлення, для чого використовуємо функцію stack(A,B,...), що дозволяє поєднувати записані через кому масиви. Об’єднання відбувається шляхом "насадження" матриці A на матрицю B; отриманої в такий спосіб матриці – на наступну матрицю (якщо така присутня) і т. д.

Зрозуміло, що початкові матриці повинні мати однакову кількість стовпців, тому необхідно перетворити мітки з рядків у вектори ASCII-кодів. Отже, sMe :stack (str2vec(μ_S), M_cod, str2vec(μ_e)).

Загальна кількість символів у схованому повідомленні: rows(sMe) = 5404. Кількість НЗБ контейнера, що для цього необхідно (8 бітів/символ): 8*rows(sMe) = 43232 біти. Загальна кількість НЗБ контейнера:

rows(C) cols(C) = 3-128-128 = 49152 > 43232 біти. Таким чином, файл зображення має достатній обсяг для того, щоб приховати повідомлення.

Крок 5. Для подальших обчислень буде потрібно перетворення десяткового числа (яким за замовчуванням кодується кожний символ) у формат двійкового. Також знадобиться і зворотне перетворення.

Зворотний процес, тобто добування повідомлення із зображення, досягається зворотним шляхом, а саме добуванням молодших бітів зображення та перетворенням їх у повідомлення.

Метод псевдовипадкового інтервалу. У розглянутому вище найпростішому випадку виконується заміна НЗБ всіх послідовно розміщених пікселей зображення. Інший підхід – метод випадкового інтервалу [5], полягає у випадковому розподілі бітів секретного повідомлення по контейнеру, у результаті чого відстань між двома убудованими бітами визначається псевдовипадково. Ця методика особливо ефективна у випадку, коли бітова довжина секретного повідомлення істотно менше кількості пікселей зображення.

Розглянемо найпростіший випадок цього методу, коли інтервал між двома послідовними вбудовуваннями бітів повідомлення є функцією координат попереднього модифікованого піксела.

Крок 1. Нехай повідомлення, яке необхідно приховати: M := "© Пузыренко А.Ю., 2005 р.". Як контейнер *C* використовуємо підмасив *B* синього колірного компонента зображення.

Крок 2. Визначимо мітки, які будуть установлювати границі корисного повідомлення в контейнері. На відміну від попереднього методу, стартова мітка буде визначати порядковий номер елемента контейнера, починаючи з якого в останній будуть заноситися дані. Нехай μ_s := 154. Мітка μ_e буде сигналізувати про завершення корисної частини серед витягнутих символів: μ_e^* := "KiHeu,6"

Крок 3. Приймемо, що при внесенні бітів повідомлення в контейнер зі змінним кроком величина останнього обумовлена кількістю одиниць у двійковому значенні номера елемента контейнера, що модифікувався попередньо. Для підрахунку величини кроку (інтервалу) скористаємося модулем $step(x) : K \sum_{i=1}^{rows(x)} x_i$, що підсумує кількість елементів матриці-стовпця x, значення яких дорівнює 1. Коефіцієнт *K* у даному випадку виступає в ролі найпростішого ключа, що може приймати будь-які цілі значення (у тому числі і негативні, але в цьому випадку стартова мітка повинна мати значення, близьке до найбільшого значення індексу елементів контейнера). Також при виборі *K* варто брати до уваги

загальну кількість бітів, необхідних для приховання повідомлення, а також наявну кількість елементів масиву контейнера. Нехай $K := 9$.

Обмежуючу мітку μe за допомогою рядкової функції concat, що поєднує рядки, які виступають у якості її аргументів, додамо до тексту повідомлення, які підлягає прихованню. Результат об'єднання перетворимо у вектор ASCII-кодів: Me := str2vec(concat(M, μe)). Загальна кількість символів в отриманому повідомленні: rows(Me) = 32.

Кількість НЗБ контейнера, що для цього необхідно (8 бітів/символ): 8 rows(Me) = 256 бітів.

Крок 4. Розгорнемо масив У в вектор, на основі якого сформуємо новий вектор, що містить приховане повідомлення.

Кожний символ повідомлення Me (операція циклу for $\mu \in$ 1..rows(Me)) переводиться у двійковий формат (змінна *b*), кожний розряд якого записується замість самого молодшого біта числа *P*, що відповідає значенню інтенсивності синього кольору визначеного піксела. При цьому елементи масиву Cv перебираються не послідовно, а зі змінним кроком, величина якого обумовлена функцією step(·).

При обчисленні функції D2B(z) необхідно змінити в граничному значенні змінної циклу *i* з 8 на таке, котре дозволить переводити у двійковий формат найстарший індекс елементів вектора *Cv*.

Стартовий елемент задається міткою μ_s. Після проведеної зміни, модифіковане двійкове число *P* переводиться у формат десяткового й записується у відповідну позицію вектора *Sv*, що на початку модуля був прийнятий рівним вектору *Cv*.

Крок 5. Зворотне згортання вектора *Sv* у масив, що має розмірність контейнера, виконується з тією лише відмінністю, що аргументом функцій розмірності масиву (rows і cols) є масив *B*. Щоб оцінити ступінь "розсіювання" бітів прихованого повідомлення по масиву контейнера, як приклад розглянемо результат присвоєння пікселу, у який планувалося ввести бітів повідомлення, нульового значення інтенсивності (чорний колір) при попередньому загальному посвітлінні зображення.

Крок 6. Результуюче кольорове зображення буде визначатися масивом об'єднання колірних масивів: S := augment(R, G, S').

Крок 7. При добуванні прихованого повідомлення повинні бути відомі параметри μ^*_s, μ^*_e, *K** та масив *B**, що, як передбачається, містить приховані дані.

Розгортання масиву *B** у вектор *Sv** виконується за допомогою аналогічного модуля. Добування повідомлення з вектора *Sv** виконується у зворотному, стосовно операції вбудовування, порядку.

З отриманого вектора *Mf** шляхом порівняння з міткою μ^*_e виділеного фрагмента витягає корисне повідомлення *M**.

M* = "© Пузыренко А.Ю., 2005 р.".

Результати обчислення візуального перекручування об'єднані в табл. 2.3.

Таблиця 2.3

Приклад обчислення ключів К

1	2	3	4	5	6	7	8	9	10	11	12
125	156	243	59	34	115	132	174	30	90	81	65

Метод псевдовипадкової перестановки. Недоліком методу псевдовипадкового інтервалу є те, що біти повідомлення в контейнері розміщені в тій же послідовності, що й у самому повідомленні, і тільки інтервал між ними змінюється псевдовипадково. Тому для контейнерів фіксованого розміру більш доцільним є використання методу псевдовипадкової перестановки (вибору) [5], зміст якого полягає в тому, що генератор ПВЧ утворить послідовність індексів $j_1, j_2, ..., j_{l\ M}$ і зберігає *–k-й* бітів повідомлення в пікселі з індексом j_k (додаток Б).

Нехай *N* – загальна кількість бітів (наймолодших) у наявному контейнері; P^N – перестановка чисел {1,2, ...,*N*}. Тоді, якщо в нас є для приховання конфіденційне повідомлення довжиною *n* бітів, то ці біти можна просто вмонтувати замість бітів контейнера P^N (1), P^N (2),..., *PN*(*n*).

Функція перестановки повинна бути псевдовипадковою, іншими словами, вона повинна забезпечувати вибір бітів контейнера приблизно випадковим чином. Таким чином, секретні біти будуть рівномірно розподілені по всьому бітовому простору контейнера. Однак при цьому індекс визначеного біта контейнера може з'явитися в послідовності більше одного разу і в цьому випадку може відбутися "перетинання" – перекручування вже убудованого біта. Якщо кількість бітів повідомлення набагато менше кількості молодших бітів зображення, то ймовірність перетинання є незначною, і перекручені біти надалі можуть бути відновлені за допомогою коригувальних кодів.

Імовірність принаймні одного перетинання оцінюється як:

$$p \approx 1 - \exp\left[\frac{l_м \cdot (l_м - 1)}{2 \cdot l_c}\right], \quad l_м \langle\langle\, l_c. \tag{2.5}$$

При збільшенні $l_М$ і $l_С$ = const дана ймовірність прагне до одиниці.

Для запобігання перетинань можна запам'ятовувати всі індекси використаних елементів *B*, і перед модифікацією нового піксела виконувати попередню його перевірку на повторюваність. Також можна застосовувати генератори ПВЧ без повторюваності чисел. Останній випадок розглянемо більш докладно.

Для наших цілей функція перестановки також залежить від секретного ключа *K*. При цьому *генератор псевдовипадкової перестановки* P^N – це функція, що для кожного значення *K* виробляє різні псевдовипадкові перестановки чисел {1,2, ...,*N*}.

Позначимо через p_K^N – генератор перестановок з відповідним ключем *K*.

Якщо перестановка p_K^N захищена по обчисленню (тобто злом алгоритму вимагає невиправдано більших витрат обчислювальних ресурсів зловмисника), то можливість розкриття змісту або припущення самого тільки виду перестановок без володіння інформацією про секретний ключ *K* практично рівняється нулю.

Секретний генератор псевдовипадкової перестановки (ГПВП) може бути ефективно реалізований на основі генератора псевдовипадкової функції (ГПВФ), котрий, як і ГПВП, виробляє різні, не підлягаючі прогнозуванню функції при кожному окремому значенні ключа; однак множина значень функцій не повинне рівнятися множині її визначення. ГПВФ легко реалізується із секретної геш-функції *H* шляхом об'єднання аргументу *i* із секретним ключем *K* та взяття від результуючого бітового рядка функції *H*:

$$F_k\ (i)=H(K_i), \tag{2.6}$$

де K_i – об'єднання (конкатенація) бітових рядків K_i;

$f_K(i)$ – результуюча псевдовипадкова функція від *i*, що залежить від параметра *K*.

Генератор Лубі (Luby) і Рекоффа (Reckoff) побудований у такий спосіб. Запис виду $a \oplus b$ має на увазі побітове додавання за модулем 2 аргумента *a* з аргументом *b*, причому результат додавання має ту ж розмірність, що і *a*.

Нехай *i* – рядок двійкових даних довжиною $2 \cdot l$. Розділимо *i* на дві частини: *x* та *y* довжиною *l* кожна, а ключ *K* на чотири частини: K_1, K_2, K_3, K_4. Тоді,

$$y = y \oplus f_{K_1}(x) = y \oplus H(K_1 \circ x);$$
$$x = x \oplus f_{K_2}(y) = x \oplus H(K_2 \circ y);$$
$$y = y \oplus f_{K_3}(x) = y \oplus H(K_3 \circ x);$$
$$x = x \oplus f_{K_4}(y) = x \oplus H(K_4 \circ y);$$

повернення в $\circ x$.

Для кожного значення ключа *K* алгоритм повертає псевдовипадкову перестановку із чисел {1, ...,2^{2l}}. Лубі й Рекофф показали, що пропонована перестановка є настільки ж секретною, наскільки й генератор ПВФ. Вони також запропонували простий алгоритм перестановки з {1,..., 2^{2l+1}}. Якщо значення функції f_k становить досить довгі бітові послідовності, той же ефект можна одержати, прийнявши, що *y* – перші *l* біти рядка *i*, а *x* – останні (*l+1*) біти.

Представлена вище конструкція дозволяє одержати перестановку $P_K^{2^k}$ з {1, ..., 2^k } для довільного *K*. Однак у випадку, коли кількість бітів контейнера становить *N*, виникає необхідність перестановки P_K^N з *{1, ...,N).*

Перевага методу полягає в тому, що існує можливість обмежитися тільки наявними для P_K^N аргументами. Нехай $k = [\log_2(N)]$ (квадратні дужки означають округлення до найменшого цілого, що більше або дорівнює аргументу). Тоді $2^{k-1} < N \leq 2^k$. При цьому підраховуються значення $P_K^{2^k}(1)$, $P_K^{2^k}(2)$,... і з послідовності віддаляються будь-які числа, що перевищують *N*. Таким чином, одержують значення $P_K^N(1)$, $P_K^N(2)$... . Помітимо, що це стає можливим, коли функція перестановки обчислена для зростаючих значень аргументів, починаючи з одиниці. Отже, генератор ПВП P^N для довільного *N* може бути побудований на основі алгоритму

Лубі й Рекоффа. Якщо ж *N* є складовим (як у випадку зображення), існує більше зручний спосіб побудови ГПВП. Наведений нижче алгоритм заснований на блоковому кодуванні з довільним розміром блоку.

Кількість бітів контейнера повинна становити складене число із двох співмножників приблизно однакового порядку, тобто $N = X \cdot Y$ для деяких X і Y.

У випадку, коли дані ховаються в НЗБ пікселей цифрового зображення, параметри X і Y є розмірами даного зображення. Для одержання координат i-го піксела зображення при прихованні біта повідомлення ($i \in \{1, ..., N\}$) необхідно виконати такі обчислення:

$$x = div(i, Y) + 1;$$
$$y = \mathrm{mod}(i, Y) + 1;$$
$$x = \mathrm{mod}(x + f_{K_1}(y), X) + 1;$$
$$y = (y + f_{K_2}(x), Y) + 1;$$
$$x = \mathrm{mod}(x + f_{K_3}(y), X) + 1;$$
$$i = (x, y) \text{ або } \mathrm{i} = (\mathrm{x} - 1) * \mathrm{Y} + \mathrm{y},$$

де *div(i, X)* і *mod(i, X)* – функції, які повертають, відповідно, ціле й залишок від розподілу / на X. Інший варіант формули застосуємо у випадку, якщо масив зображення попередньо був розгорнутий у вектор (по рядках). Додаток одиниці необхідний при індексації елементів масиву зображення, починаючи з одиниці.

Перші два раунди алгоритму необхідні для того, щоб "розсіяти" біти приховуваного повідомлення серед найменш значущих бітів контейнера. При цьому перший раунд надає випадковий характер j_c-координатам піксела-контейнера, а другий у-координатам. Третій раунд необхідний для протидії атаці на відкритий (незашифрований) текст.

У випадку використання тільки двох раундів, нехай $i = (b - 1) \cdot Y + a$, а $P_K^N(i)$ – значення перестановки. Якщо криптоаналітик здатний припустити значення a й може одержати пару "відкритий текст – кодований текст" ($i' = (z - 1) \cdot Y + a, P_K^N(i')$) для деякого z, то він здатний установити b.

Крок 1

Повідомлення, яке необхідно приховати: M := "© Пузоренко А.Ю., 2009 р.".

Контейнер C – підмасив B синьої кольорової складової зображення. При цьому кількість бітів у повідомленні: L_M := strlen(M), L_M = 200 бітів; геометричні розміри контейнера: X := rows(C), X = 128 пікселей; Y := cols(C), Y = 128 пікселей; $N := X\,Y$, N = 16384.

Крок 2

Для формування ключа використовуємо модуль, що дозволяє на підставі первинного ключа $K_O > 2$ сформувати вектор, що буде містити R пар ключів (кожна пара ключів буде використовуватися у відповідному раунді обчислення координат *x* і *y*).

Крок 3

Вбудовування бітів повідомлення в псевдовипадкові пікселі контейнера виконаємо за допомогою модуля, що реалізує алгоритм (5.3). На початку модуля масиву *S* привласнюються значення вихідного масиву *C*. Також виконується конвертування повідомлення зі строкового формату у вектор двійкових даних Mvec_bin. При обчисленні координат *x* і *y* використовується операція векторизації, що дозволяє поелементно складати за модулем 2 двійкові вектори *K* і *y* (або *x*). При цьому розмірність зазначених векторів повинна бути однаковою, для чого використовується функція submatrix.

Крок 4

На приймаючій стороні повинні бути відомі первинний ключ K_O * *i* масив кольоровості, у який виконувалося вбудовування (*S**). З останнього виходять значення *X**, *Y**, *N**.

Алгоритми, описувані в даному пункті, впроваджують ЦВДЗ в множини вихідного зображення. Їх перевагою є те, що для впровадження ЦВДЗ немає необхідності виконувати обчислювально громіздкі лінійні перетворення зображень. ЦВДЗ впроваджується за рахунок маніпуляцій яскравістю $I(x,y) \in \{1,...,L\}$ або колірними складовими $(r(x,y), b(x,y), g(x,y))$.

Алгоритм Kutter. Нехай зображення має RGB-кодування. Вбудовування виконується в канал синього кольору, тому що до синього кольору система людського зору найменш чутлива. Розглянемо алгоритм передачі одного біта секретної інформації.

Нехай s_i – бітів, що, вбудовується $I = \{R,G,B\}$ – контейнер, $p = (x,y)$ – псевдовипадкова позиція, у якій виконується вкладення. Секретний бітів вбудовується в канал синього кольору шляхом модифікації яскравості $I(p) = 0{,}299r(p) + 0{,}587g(p) + 0{,}114b(p)$:

$$b'(p) = \begin{cases} b(p) + qI(p), & \text{якщо } s_i = 0, \\ b(p) - qI(p), & \text{якщо } s_i = 1, \end{cases} \qquad (2.7)$$

де *q* – константа, що визначає енергію сигналу, що вбудовується. Її величина залежить від призначення схеми. Чим більше *q*, тим вище працездатність вкладення, але тим сильніше його помітність.

Витяг біта одержувачем здійснюється без наявності в нього вихідного зображення, тобто наосліп. Для цього виконується пророкування значення вихідного, немодифікованого піксела на підставі значень його сусідів. У роботі [5] пропонується для одержання оцінки піксела використовувати значення декількох пікселів, розташованих у тому же стовпці й тому же рядку. Автори використовували "хрест" пікселів розміром 7×7. Оцінка $\hat{b}''(p)$ виходить у вигляді:

$$\hat{b}''(p)=\frac{1}{4c}\left(-2b''(p)\sum_{i=-c}^{+c}b''(x+i,y)+\sum_{k=-c}^{+c}b''(x,y+k)\right), \qquad (2.8)$$

де c – кількість пікселів зверху (знизу, ліворуч, праворуч) від оцінюваного піксела $(c=3)$. Тому що в процесі вбудовування ЦВДЗ кожний бітів був повторений c_r разів, одержимо c_r оцінок одного біта ЦВДЗ. Секретний бітів перебуває після усереднення різниці оцінки піксела і його реального значення

$$\delta=\frac{1}{cr}\sum_{i=1}^{cr}\hat{b}_i(p)-b_i(p). \qquad (2.9)$$

Знак цієї різниці визначає значення убудованого біта.

Чи можна гарантувати завжди правильне визначення значення секретного біта? Ні, тому що функція витягування біта не є зворотної функції вбудовування. Для підвищення надійності необхідне застосування додаткових заходів.

У роботі [41] розглянута також і модифікація даного алгоритму для вбудовування декількох бітів. Показано, що алгоритм є працездатним до багатьох з відомих атак: низькочастотної фільтрації зображення, його стиску відповідно до алгоритму JPEG, обрізанню країв.

Алгоритм Bruyndonckx [43]. ЦВДЗ становить рядок бітів. Для підвищення завадостійкості застосовується код БЧХ. Впровадження здійснюється за рахунок модифікації яскравості блоку 8 × 8 пікселів.

Процес вбудовування здійснюється в три етапи:

1. Класифікація, або поділ пікселів усередині блоку на дві групи із приблизно однорідною яскравістю.

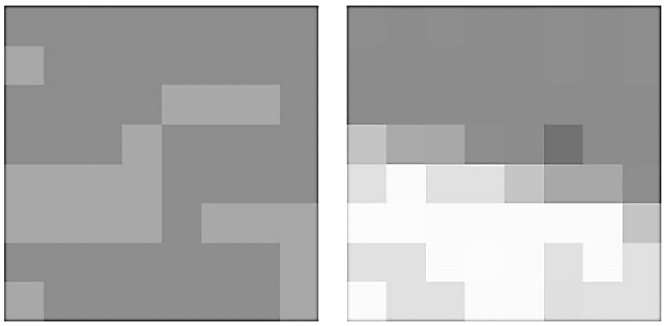
2. Розбивка кожної групи на категорії, обумовлені даною сіткою.

3. Модифікація середніх значень яскравості кожної категорії в кожній групі.

Розглянемо докладніше кожний із цих етапів.

1. При класифікації автори виділяють два типи блоків: блоки з "шумовим контрастом" (рис. 2.10а) і блоки з різко вираженими перепадами яскравості (рис. 2.10б).

У блоках другого типу зони з яскравістю, що відрізняється, не обов'язково повинні розташовуватися впритул друг до друга, не обов'язково повинні містити рівну кількість пікселів. Більше того, деякі пікселі взагалі можуть не належати жодній зоні. У блоках першого типу класифікація особливо утруднена.



а) б)

Рис. 2.10. **Два типи блоку:**
а) з нечітким контрастом; б) з різко вираженим контрастом

Для виконання класифікації значення яскравості сортуються за зростанням (рис. 2.11а і б). Далі перебуває точка, у якій нахил дотичної до кривої, що вийшла, максимальний (α). Ця точка є границею, що розділяє дві зони в тому випадку, якщо нахил більше деякого порога. У противному разі пікселі діляться між зонами нарівно.

2. Для сортування пікселів по категоріях на блоки накладаються маски, різні для кожної зони й кожного блоку. Призначення масок полягає в забезпеченні таємності впровадження. Приклад масок для двох зон наведений на рис. 2.11а і б.

A	A	B	B	A	A	B	B
A	A	B	B	A	A	B	B
B	B	A	A	B	B	A	A
B	B	A	A	B	B	A	A
A	A	B	B	A	A	B	B
A	A	B	B	A	A	B	B
B	B	A	A	B	B	A	A
B	B	A	A	B	B	A	A

B	B	B	B	A	A	A	A
B	B	B	B	A	A	A	A
B	B	B	B	A	A	A	A
B	B	B	B	A	A	A	A
A	A	A	A	B	B	B	B
A	A	A	A	B	B	B	B
A	A	A	A	B	B	B	B
A	A	A	A	B	B	B	B

а) б)

Рис. 2.11. **Приклади використовуваних масок**

3. *Модифікація*. Отже, множина пікселів виявилася розділеною на п'ять підмножин: дві зони: дві категорії + пікселі, що не належать якій-небудь зоні (для блоків першого типу). Позначимо середнє значення яскравості для пікселів двох зон і категорій через $l_{1A}, l_{2A}, l_{1B}, l_{2B}$. Відомо, що $l_{1A} < l_{2A}$, $l_{1B} < l_{2B}$. Вбудовування біта ЦВДЗ s здійснюється за таким правилом:

$$s = \begin{cases} 1, & \begin{cases} l'_{1A} > l'_{1B}, \\ l'_{2A} > l'_{2B}, \end{cases} \\ 0, & \begin{cases} l'_{1A} < l'_{1B}, \\ l'_{2A} < l'_{2B}. \end{cases} \end{cases} \qquad (2.10)$$

З іншого боку, необхідно забезпечити рівність значень яскравості в кожній зоні:

$$\frac{n_{1A} l'_{1A} + n_{1B} l'_{1B}}{n_{1A} + n_{1B}} = l_1 \text{ та } \frac{n_{2A} l'_{2A} + n_{2B} l'_{2B}}{n_{2A} + n_{2B}} = l_2. \qquad (2.11)$$

Для досягнення цього яскравість всіх пікселів однієї зони міняється однаково. Наприклад, для зони 1 категорії *A* ця зміна складе $l'_{1A} - l_{1A}$.

Алгоритм витягу ЦВДЗ є зворотним алгоритму впровадження. При цьому обчислюються середні значення яскравостей і знаходяться різниці

$$s'' = \begin{cases} 0, & \text{якщо } l''_{1A} - l''_{1B} < 0 \text{ та } l''_{2A} - l''_{2B} < 0, \\ 1, & \text{якщо } l''_{1A} - l''_{1B} > 0 \text{ та } l''_{2A} - l''_{2B} > 0. \end{cases} \qquad (2.12)$$

Алгоритм Langelaar. Даний алгоритм також працює із блоками 8x8. Спочатку створюється псевдовипадкова маска нулів і одиниць такого ж розміру $pat(x,y) \in \{0,1\}$. Далі кожний блок B ділиться на два субблоки B_0 й B_1, залежно від значення маски. Для кожного субблоку обчислюється середнє значення яскравості, l_0 і l_1. Далі вибирається деякий поріг α, і бітів ЦВДЗ вбудовується в такий спосіб:

$$s = \begin{cases} 1, & l_0 - l_1 > +\alpha, \\ 0, & l_0 - l_1 < -\alpha. \end{cases} \qquad (2.13)$$

Якщо умова (2.13) не виконується, змінюємо значення яскравості пікселів субблоку B_1. Для витягнення біта ЦВДЗ обчислюються середні

значення яскравості субблоків I''_0 і I''_1. Різниця між ними дозволяє визначити шуканий бітів:

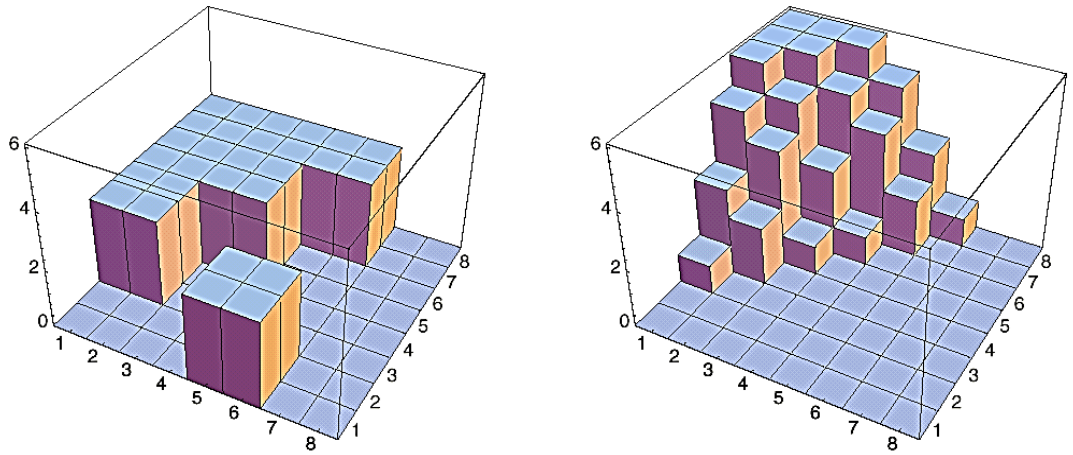
$$s = \begin{cases} 1, & I_0 - I_1 > 0, \\ 0, & I_0 - I_1 < 0. \end{cases} \qquad (2.14)$$

ЦВДЗ становить двовимірний масив бітів розміром із зображення, причому число одиниць у ньому дорівнює кількості нулів. Існує кілька версій алгоритму, запропонованого Пітасом. Спочатку пропонувалося вбудовувати бітів ЦВДЗ у кожний піксел зображення, але пізніше розсудливо було вирішено використовувати для цієї мети блоки розміром 2×2 або 3×3 пікселів, що робить алгоритм більш працездатним до стиску або фільтрації. ЦВДЗ складається із зображенням:

$$I'(x,y) = I(x,y) + \alpha s(x,y). \qquad (2.15)$$

У випадку використання для впровадження блоків детектор ЦВДЗ обчислює середнє значення яскравості цього блоку. Звідси з'являється можливість нерівномірного впровадження ЦВДЗ у піксели, тобто величина $\alpha \neq const$.

У такий спосіб можна одержати ЦВДЗ, оптимізований за критерієм працездатності до процедури стиску алгоритмом JPEG. Для цього в блоці 8×8 елементів заздалегідь обчислюють "ємність" кожного піксела з обліком ДКП і матриці квантування JPEG. Потім ЦВДЗ впроваджують відповідно до обчисленої ємності. Ця оптимізація виробляється раз і назавжди, і знайдена маска застосовується для будь-якого зображення. На рис. 2.12а і б показаний ЦВДЗ до й після оптимізації.



а) б)

Рис.2.12. **Оптимізація ЦВДЗ: а) до оптимізації; б) після оптимізації**

Алгоритм Rongen [45]. В алгоритмі ЦВДЗ становить двовимірну матрицю одиниць і нулів із приблизно рівною їх кількістю. Піксели, у які можна впроваджувати одиниці, тобто робастні до перекручувань, визначаються на основі деякої характеристичної функції (характеристичні пікселі). Ця функція обчислюється локально, на основі аналізу сусідніх пікселів. Характеристичні пікселі становлять приблизно 1/100 від загальної кількості, так що не всі одиниці ЦВДЗ вбудовуються саме в ці позиції. Для підвищення кількості характеристичних пікселів якщо буде потреба, пропонується здійснювати невелике переспотворення зображення.

Детектор знаходить значення характеристичних пікселів і порівнює з наявним у нього ЦВДЗ. Якщо в зображенні ЦВДЗ не втримується, то в характеристичних пікселах кількість одиниць і нулів буде приблизно нарівно. Автори розрахували значення порога ухвалення рішення, мінімізуючого ймовірність фіктивної тривоги.

Алгоритм Patchwork [65]. В основі алгоритму Patchwork лежить статистичний підхід. Спочатку псевдовипадковим чином на основі ключа вибираються два піксели зображення. Потім значення яскравості одного з них збільшується на деяке значення (від 1 до 5), значення яскравості іншого зменшується на те ж значення. Далі цей процес повторюється велику кількість разів (~10000) і знаходиться сума значень всіх різниць. За значенням цієї суми судять про наявність або відсутність ЦВДЗ у зображенні.

Для пояснення роботи алгоритму введемо ряд позначень. Нехай значення обираних на кожному кроці пікселів – a_i і b_i, величина збільшення – δ. Тоді сума різниць значень пікселів:

$$S_n = \sum_{i=1}^{n}\left[(a_i + \delta) - (b_i - \delta)\right] = 2\delta n + \sum_{i=1}^{n}(a_i - b_i) \qquad (2.16)$$

Маточікування величини $\sum_{i=1}^{n}(a_i - b_i)$ (суми різниці значень пікселів у незаповненому контейнері) близьке до нуля при досить великому n. Маточікування величини S_n буде більше 2δ. S_n має гаусівський розподіл. Таким чином, у стеганодетекторі відповідно до ключа перевіряється значення S_n й у тому випадку, якщо воно значно відрізняється від нуля, виноситься рішення про наявність ЦВДЗ.

Авторами також запропоновані поліпшення основного алгоритму для підвищення його робастності. Замість окремих пікселів пропонується використовувати блоки, або patches. Звідси й назва алгоритму. Використання блоків різного розміру може розглядатися як формування спектра внесеного ЦВДЗ шуму (шейпінг), аналогічно тому, як це застосовується в сучасних модемах. Оскільки найбільш імовірною модифікацією стега є компресія JPEG, то доцільно, щоб спектр ЦВДЗ перебував у множині низьких частот. З іншого боку, якщо характер можливих модифікацій стега заздалегідь невідомий, доцільне застосування сигналів з розширеним спектром. Від форми блоку залежить невидимість внесених перекручувань.

Алгоритм Patchwork є досить стійким до операцій стиску зображення, його усікання, зміни контрастності. Основним недоліком алгоритму є його нестійкість до афінних перетворень, тобто поворотів, зрушень, масштабування. Інший недолік полягає в малій пропускній здатності. Так, у базовій версії алгоритму для передачі 1 біта прихованого повідомлення потрібно 20 000 пікселів.

Алгоритм Bender [65]. Алгоритм, заснований на копіюванні блоків з випадково обраної текстурної множини в іншу, що має подібні статистичні характеристики. Це приводить до появи в зображенні повністю однакових блоків. Ці блоки можуть бути виявлені в такий спосіб:

1. Аналіз функції автокореляції стеганозображення й знаходження її піків.

2. Зрушення зображення відповідно до цих піків і вилучення зображення з його зрушеної копії.

3. Різниця в місцях розташування копійованих блоків повинна бути близька до нуля. Тому можна вибрати деякий поріг і значення, менші цього порога за абсолютною величиною, уважати шуканими блоками.

Оскільки копії блоків ідентичні, то вони змінюються однаково при перетвореннях усього зображення. Якщо зробити розмір блоків досить більшим, то алгоритм буде стійким стосовно більшості з негеометричних перекручувань. У проведених експериментах показана працездатність алгоритму до фільтрації, стиску, поворотів зображення.

Основним недоліком алгоритму є виняткова складність знаходження галузей, блоки з яких можуть бути замінені без помітного погіршення якості зображення. Крім того, у даному алгоритмі як контейнер можуть використовуватися тільки досить текстурні зображення.

Один з перших запропонованих способів для перевірки автентичності зображень одержав назву методу перевірочних сум. Відповідно до цього методу відбиралися сім старших бітів восьми прилеглих пікселів. Виходило 56-бітне слово. Виконавши цю операцію для всього зображення, мали $N \times N / 8$ таких слів, де $N \times N$ – кількість пікселів у зображенні. Потім вони порозрядно складалися за модулем 2, тобто обчислювалася перевірочна сума довжиною 56 бітів. Ця сума записувалася в молодші значущі біти обраних відповідно до ключа пікселів. У детекторі здійснювалася перевірка цих бітів, перевірочна сума рівнялася з еталонною, і виносилося рішення про наявність або відсутність модифікації зображення. Таким чином, у даному алгоритмі як ключ використовується місце розташування перевірочної суми пікселів і ця перевірочна сума.

Більшість запропонованих алгоритмів вбудовування ЦВДЗ у просторову область зображень засновані на використанні широкосмугових сигналів (ШСС). Цей метод добре зарекомендував себе в радіозв'язку, при передачі вузькосмугових сигналів каналами з шумами. Основною ідеєю застосування ШСС у стеганографії є те, що дані впроваджуються в шумовий сигнал малої потужності. Оскільки сигнал малої потужності, то для захисту ЦВДЗ застосовують завадостійкі коди. Розглянемо приклад.

Алгоритм Marvel [47]. Стеганокодер із застосуванням ШСС наведений на рис. 2.13.

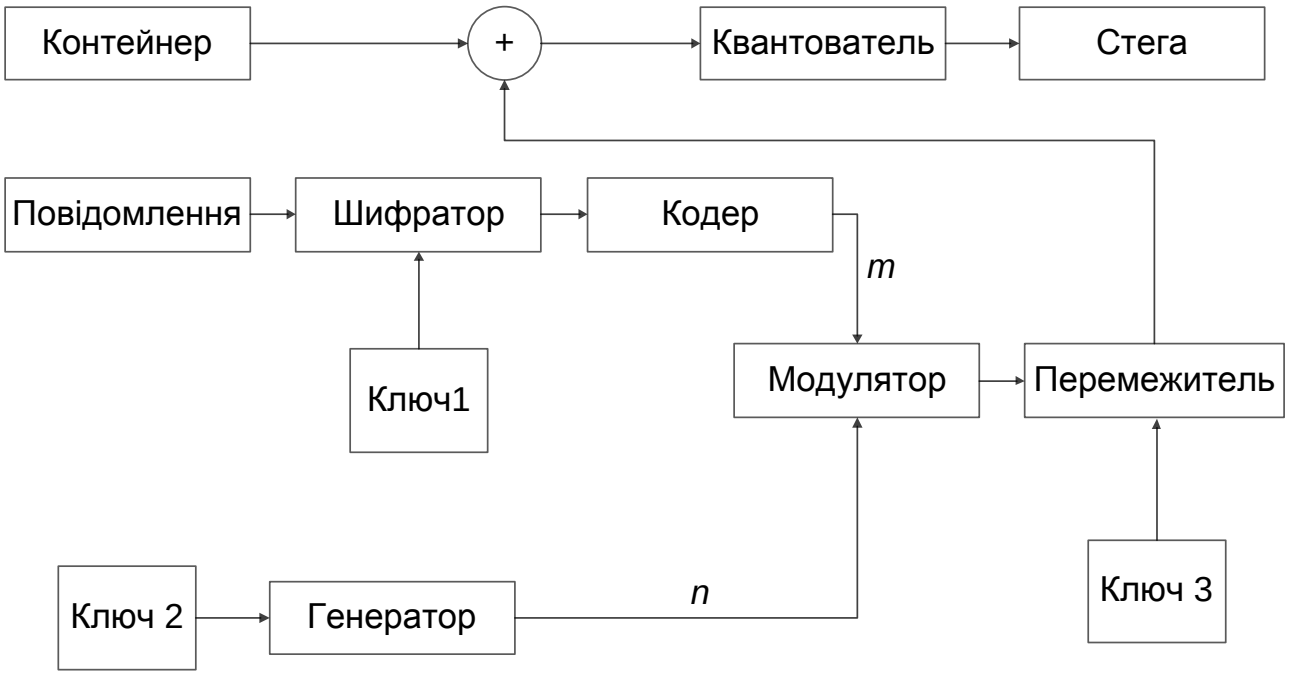


Рис. 2.13. **Стеганокодер на основі ШСС**

Приховуване повідомлення шифрується на ключі k_1 й кодується завадостійким кодом, у результаті чого виходить кодоване повідомлення m. Це повідомлення модулюється псевдовипадковою послідовністю з виходу генератора, початкове заповнення якого дорівнює k_2 сигнал, що вийшов, з розширеним спектром піддається перестановкам відповідно до ключа k_3 й складається із зображенням-контейнером. У декодері виконуються зворотні операції. Як детектор ЦВДЗ використовують кореляційний приймач.

Як датчик псевдовипадкової послідовності найчастіше пропонується використовувати генератор *M*-послідовності в силу гарних кореляційних властивостей цієї послідовності.

2.6. Приховування даних у просторі множини зображень (блокове приховування, метод квантування, метод "хреста")

Метод блокового прихування – це ще один підхід до реалізації методу заміни, він полягає в такому [5; 11; 32; 34]. Зображення-оригінал розбивається на l_M непересічних блоків $\Delta_i \left(1 \le i \le l_M\right)$ довільної конфігурації, для кожного з яких обчислюється бітів парності $b(\Delta_i)$: $b(\Delta i) = \sum_{j \in \Delta i}^{\mathrm{mod}\ 2} LSB(C_J)$.

У кожному блоці виконується приховання одного секретного біта M_t. Якщо бітів парності $b\left(\Delta_i\right) \ne M_i$, то відбувається інвертування одного із НЗБ блоку Δ_i, у результаті чого $b\left(\Delta_i\right) = M_i$. Вибір блоку може відбуватися псевдовипадково з використанням стеганоключа.

Хоча цей метод має таку ж низьку стійкість до перекручувань, як і всі попередні, у нього є ряд переваг. По-перше, існує можливість модифікувати значення такого піксела в блоці, зміна якого приведе до мінімальної зміни статистики контейнера. По-друге, вплив наслідків вбудовування секретних даних у контейнер можна зменшити за рахунок збільшення розміру блоку.

Розглянемо приклад програми в MathCAD, що дозволяє виконати стеганографічний захист текстового повідомлення методом блокового прихування.

Крок 1

Вихідні дані відповідають прийнятим при моделюванні попереднього методу.

Крок 2

Розбивка масиву контейнера на блоки виконаємо в такий спосіб: якщо кількість бітів у повідомленні (L_M) не перевищує кількості стовпців *Y* масиву C, той один блок відповідає окремому стовпцю масиву *C*. Якщо ж $L_M > Y$, той один блок дорівнює $1/\chi$ від окремого стовпця масиву, де $\chi = ceil\ (L_M / Y)$. Значення *X* повинне бути відомо одержувачеві.

Лічильник σ дозволяє виділяти відповідному співвідношенню χ частину від загальної розмірності стовпця масиву. При цьому визначаються індекси рядків, починаючи з якого (r_1) і по який (r_2) виділяється фрагмент Δ *b*-го стовпця.

Для кожного блоку Δ виконується обчислення біта парності *b*. Якщо *b* не дорівнює поточному значенню біта повідомлення, то із блоку Δ випадковим чином вибирається індекс піксела, інтенсивність кольору якого збільшується або зменшується на одиницю, залежно від того, парним або непарним є його первинне значення. За допомогою функції putregion(*S,* Δ*, r_1, y*) виконується вбудовування модифікованого масиву Δ у загальний масив *S*, починаючи з рядка r_1 і стовпця в убік найстарших індексів рядків і стовпців відповідно.

2.7. Приховування даних у частотній множині зображень. Метод Коха – Жао та його модифікації

Один з найпоширеніших на сьогодні методів приховання конфіденційної інформації в частотній множині зображення полягає у відносній заміні величин коефіцієнтів ДКП, що у свій час описали Е. Кох (E. Koch) і Дж. Жао (J. Zhao) [39].

На початковому етапі первинне зображення розбивається на блоки розмірністю 8×8 пікселей. ДКП застосовується до кожного блоку, у результаті чого одержують матриці 8×8 коефіцієнтів ДКП, які найчастіше позначають $Q_{ft}(u,v)$, де *b* – номер блоку контейнера *C*, *a (o,v)* – позиція коефіцієнта в цьому блоці. Кожний блок при цьому призначений для приховання одного біта даних (дод. В).

Було запропоновано дві реалізації алгоритму: псевдовипадково можуть вибиратися два або три коефіцієнти ДКП. Розглянемо перший варіант.

Під час організації секретного каналу абоненти повинні попередньо домовитися про конкретні коефіцієнти ДКП із кожного блоку, які будуть використовуватися для приховання даних. Задамо дані коефіцієнти їх координатами в масивах коефіцієнтів ДКП: (υ_1,ν_1) і (υ_2,ν_2). Крім цього, зазначені коефіцієнти повинні відповідати косинус-функціям із середніми частотами, що забезпечить прихованість інформації в істотних для ЗСЧ областях сигналу, до того ж інформація не буде спотворюватися при JPEG-компресії з малим коефіцієнтом стиску.

Безпосередньо процес приховання починається з випадкового вибору блоку C_b зображення, призначеного для кодування *b*-го біта повідомлення. Вбудовування інформації здійснюється в такий спосіб: для передачі біта "0" прагнуть, щоб різниця абсолютних значень коефіцієнтів ДКП перевищувала деяку позитивну величину, а для передачі біта "1" ця різниця робиться меншої порівняно з деякою негативною величиною.

Таким чином, первинне зображення спотворюється за рахунок внесення змін у коефіцієнти ДКП, якщо їх відносна величина не відповідає приховуваному біту. Чим більш значення, тим стеганосистема, створена на основі даного методу, є більше стійкою до компресії, однак якість зображення при цьому значно погіршується.

Після відповідного внесення корекції в значення коефіцієнтів, які повинні задовольняти нерівності, проводиться зворотне ДКП.

Для витягнення даних у декодері виконується аналогічна процедура вибору коефіцієнтів, а рішення про переданий біт приймається відповідно до такого правила:

Крок 1

Виділяємо масиви колірних компонентів зображення:

У зв’язку з низькою чутливістю ЗСЧ до каналу синього кольору й можливим при певних обставинах досить значним перекручуванням контейнера при вбудовуванні, секретне повідомлення будемо вбудовувати в масив B.

Визначимо розмірність масиву B и задамо розмірність сегментів (блоків), на які він буде розбиватися: кількість рядків *X := rows{B)*, *X = 128*; кількість стовпців *Y := cols(B)*, *Y = 128*; розмірність сегментів *N ’= 8* пікселей.

Загальна кількість сегментів, на яку розбивається контейнер: $Nc := X – Y/N^2$, *Nc = 256* сегментів.

Крок 2

Розбивку масиву *B* на сегменти *Ic* проводимо за допомогою модуля (2.45).

Кожний сегмент призначений для приховання одного біта конфіденційного повідомлення *M*. Тому попередньо необхідно перевірити достатність кількості сегментів для цієї операції.

Нехай, як і в попередніх випадках, повідомлення має вигляд *M := "©Назаренко А.Ю., 2005 р."*. При цьому кількість бітів у приховуваному повідомленні (один символ повідомлення кодується одним байтом):*1-м := 8 strlen(M) = 200 бітів* < N_c *= 256*. Отже, обсяг повідомлення є прийнятним для вбудовування.

Крок 3

Застосуємо до кожного із сегментів пряме дискретне косинусне перетворення. Програмний модуль прямого ДКП складається із двох частин: перша визначає значення коефіцієнтів *£* для поточного значення аргументу *X*, друга проводить обчислення спектральних коефіцієнтів ДКП для кожного сегмента C_D. при цьому вертається відповідна матриця розмірністю N_x.

Коефіцієнт у лівому верхньому куті матриці Ω_b (нагадаємо, що в нашому випадку нумерація елементів масивів починається з одиниці) $(\Omega_b)_{1,1}$ містить інформацію про яскравість усього сегмента (його найчастіше називають DC-коефіцієнтом). Інші коефіцієнти називаються AC-коефіцієнтами. Відзначимо також, що коефіцієнти НЧ компонентів розміщені ближче до лівого верхнього кута, а ВЧ компонентів – ближче до правого нижнього кута.

2.8. Приховування даних у частотній множині зображень. Метод Хсу – Ву та метод Фрідріха

Хсу (Chiou-Ting Hsu) і Ву (Ja-Ling Wu) [7; 25] був запропонований алгоритм вбудовування цифрового водного знака в масив коефіцієнтів ДКП блоків зображення-контейнера. Наведемо основні положення, закладені авторами в основу алгоритму.

Нехай *C* – напівтонове зображення розміром *X*×*F*, а *W* ЦВДЗн, який представляє собою двійкове зображення розміром *A*×*Z*. У ЦВДЗн піксел може приймати значення або "1", або "0". Зрозуміло, що безпосереднє

спостереження такого зображення неможливе, оскільки інтенсивності 0 і 1 відповідають чорному кольору (останній у деякому наближенні). Зображення ЦВДЗн можна створити чорно-білим, а перед приховуванням замінити інтенсивність білих пікселей (255) на одиницю, наприклад, шляхом розподілу всього масиву ЦВДЗн на 255. При добуванні, навпаки, для візуального спостереження масив ЦВДЗн необхідно помножити на 255.

Оскільки, як буде показано надалі, під час вбудовування ЦВДЗн буде оброблятися тільки середньочастотний діапазон сигналу-контейнера, необхідною передумовою є те, що ЦВДЗн повинен мати менший порівняно з контейнером розмір. Так, наприклад, для контейнера, розбитого на блоки 8х8, при вбудовуванні ЦВДЗн оптимальним буде використання *64*A*Z/(X*Y)* коефіцієнтів ДКП. Відношення *A*Z/(X*Y)* у цьому випадку визначає та кількість інформації, що може бути убудована в обране як контейнер зображення (у наведеному прикладі – до 64 коефіцієнтів блоку 8×8). Для більшої стійкості й прихованості результатів використання розглянутого стегано-методу кількість убудованої інформа-ції на практиці намагаються зменшити.

Зображення-контейнер *C* і ЦВДЗн *W* представимо як:

$$C = \{c(x,y);1 \le x \le X;1 \le y \le Y\},$$
$$W = \{w(a,z);1 \le a \le A;1 \le z \le Z\}.$$

де $c(x,y) \in \{0,...,2^L - 1\}$ – інтенсивність піксела *(x,y)*; *L* – кількість бітів, що використовується для квантування інтенсивностей; $w(a,z) \in \{0,1\}$ – двійкові значення піксела *(a,z)* ЦВДЗн.

Контейнер *C* можна розбити на $\frac{X}{8} x \frac{Y}{8}$ блоків розмірністю 8×8. Для одержання цієї ж кількості блоків ЦВДЗн розбивається на блоки розмірністю $\frac{8A}{X} x \frac{8Z}{Y}$. Наприклад, якщо *A =X/2 і Z= Y/2*, розмірність блоку ЦВДЗ складе 4×4; якщо ж *A = X/4 и Z = Y/4* – 2×2 і т. д.

Для створення контейнера або ЦВДЗн необхідної розмірності до останнього можуть бути додані додаткові стовпці або рядки.

Псевдовипадкова перестановка пікселей ЦВДЗн

У деякому наближенні кожний блок ЦВДЗн вбудовується в середньочастотні коефіцієнти ДКП кожного блоку контейнера шляхом використання блокового перетворення всього контейнера. Тому замість контейнера в цілому кожний блок ЦВДЗн буде неуважним тільки лише по відповідний

йому блок. При цьому очевидно, що у випадку відсутності належного регулювання просторових зв'язків ЦВДЗн, звичайне масштабування контейнера може привести до руйнування ЦВДЗн (див. додаток Б).

Для забезпечення стійкості до масштабування, з метою зміни порядку ЦВДЗн для розосередження його просторових зв'язків, авторами було запропоновано використовувати швидкий метод генерації двовимірного ПВЧ:

$$W_{md} = permute \quad (W);$$
$$W_{md} = \{w_{md}(a,z) = w_{md}(a',z'); 1 \le a \le A; 1 \le z \le Z\},$$

де піксель *(a', z')* становить переставлений відповідно до псевдовипадкової перестановки (оператор *permute***)** піксель *(a, z).*

Перестановка блоків ЦВДЗ залежно від характеристик блоків контейнера. Відповідно до збільшення рівня прихованості, повинні бути враховані характеристики контейнера, наприклад, відомо, що модифікація високочастотного діапазону або ж ділянок з більшою яскравістю є менш помітною. Подібні, залежні від контейнера, властивості можуть бути використані для перестановки псевдовипадково змішаного ЦВДЗ для одержання більшої відповідності чутливості ЗСЧ.

Автори пропонують упорядкувати блоки контейнера відповідно до зміни дисперсій інтенсивностей пікселей (наприклад, за їх зменшенням). У свою чергу, блоки ЦВДЗ сортуються за кількістю інформації, тобто кількістю значущих (одиничних) пікселей. Вид сортування блоків ЦВДЗ (за зростанням або убуванням) повинен відповідати аналогічний операції над блоками контейнера.

Таким чином, кожному блоку контейнера відповідає свій блок ЦВДЗ, тобто *Wsort* = permute(*Wrnd).*

На рис. 2.14 наведений приклад сортування й перестановки блоків.

Перетворення блоків контейнера

Оскільки ДКП, використовуване при JPEG-компресії, оперує із блоками розмірністю 8×8, вважається доцільним розбити контейнер *З* на блоки зазначеної розмірності. До кожного блоку застосовується операція прямого ДКП (оператор FDCT):

$$\Omega = FDCT\ (C). \qquad (2.17)$$

Індекс блоку контейнера	Значення дисперсії
7	67.1
2	61.2
3	53.0
6	41.9
1	32.3
5	14.5
4	7.4

Індекс блоку ЦВЗ	Кількість одиниць
7	67.1
2	61.2
3	53.0
6	41.9
1	32.3
5	14.5
4	7.4

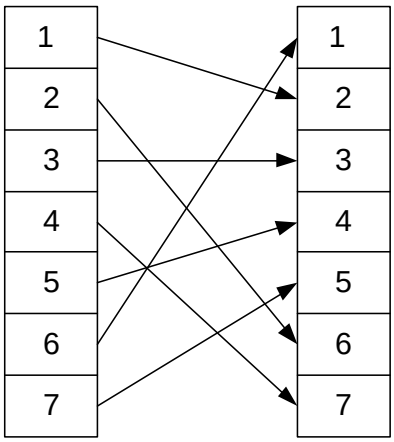


Рис. 2.14. **Приклад перестановки блоків ЦВДЗ залежно від характеристик блоків контейнера**

Вибір середньочастотних коефіцієнтів ДКП

Для того щоб убудований ЦВДЗ візуально був непомітним і залишався при цьому досить стійким і, стиску даних із втратами, очевидним компромісом є його вбудовування в діапазон середніх частот контейнера. При цьому для кожного блоку 8х8 контейнера з можливих 64-х відбираються *64*A*Z/(X*Y)* коефіцієнтів ДКП, розташованих уздовж другорядної діагоналі матриці ДКП.

Відібрані коефіцієнти для зручності наступних дій перетворюються в зменшену матрицю розмірністю $\frac{8A}{X} \times \frac{8Z}{Y}$ (оператор *reduce*):

$$\Omega_{mid} = reduce\,(\Omega)\,. \qquad (2.18)$$

Зокрема, якщо *A=X/2* і *Z=Y/2*, під час вбудовування ЦВДЗ обробляються тільки 16 коефіцієнтів ДКП, а інші 48 залишаються незміненими.

Розглянутий вище процес формування масиву СЧ-коефіцієнтів ДКП проілюстрований на рис. 2.15.

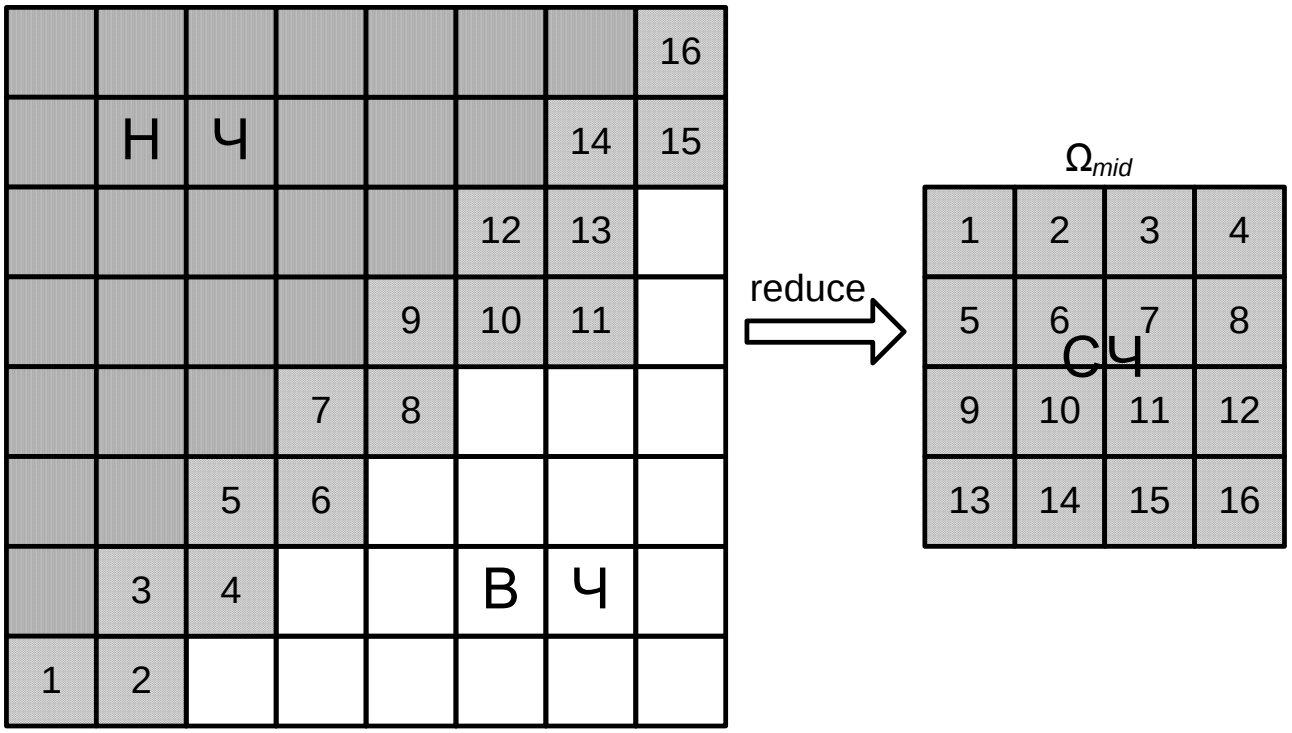


Рис. 2.15. **Формування масиву СЧ-коефіцієнтів ДКП**

Модифікація коефіцієнтів ДКП

У результаті попередніх дій маємо: переставлений у псевдо-випадковому порядку й наведений у відповідність до блоку контейнера блок ЦВДЗ, а також наведене частотне відображення контейнера (утримуюче тільки СЧ-компоненти первинного зображення), – розмірністю $\frac{8A}{X} \times \frac{8Z}{Y}$.

Елементи масиву Ω_{mid} можуть бути модифіковані відповідно до пікселів убудованого блоку ЦВДЗ.

На думку авторів алгоритму, ефективним засобом досягнення непомітності ЦВДЗ і стійкості стеганосистеми при низьких коефіцієнтах JPEG-компресії є вбудовування кожного піксела ЦВДЗ шляхом модифікації полярності між відповідними пікселами сусідніх блоків. Однак при цьому обмовляється, що такий метод не буде стійким до атак при високому (≤ 6) коефіцієнті компресії.

Пропонується розглянути технічні аспекти проблеми вбудовування при зазначеному підході, а також удосконалений метод, що є більш стійким до атак компресії.

Вбудовування шляхом модифікації відносин між значеннями коефіцієнтів сусідніх блоків

Для розрахунку полярності обраних СЧ-коефіцієнтів сусідніх блоків використовується так звана "залишкова" маска. На рис. 2.16 наведений приклад такої маски, де кожний елемент (від "А" до "И") містить у собі наведене відображення коефіцієнтів ДКП контейнера Ω_{mid} деякого блоку, причому позиції "Д" відповідає поточне відображення ДКП.

А	Б	В
Г	Д	Е
Ж	З	И

Рис. 2.16. **Приклад залишкової маски**

Наприклад, якщо А = Б = В = Е = Ж = З = И = 0,

Г = –1, а Д = 1, то полярність буде становити двійковий образ – масив нулів і одиниць, який вказує на той факт, що коефіцієнт ДКП поточної позиції даного блоку відображення коефіцієнтів ДКП є більшим (полярність дорівнює 1) або меншим (полярність дорівнює 0) порівняно з коефіцієнтом на відповідній позиції попереднього блоку. Тобто для наведеного прикладу

$$P = polarity(\Omega_{mid}) = \begin{cases} 1, \text{при } \omega_{\text{mid}_b}(\upsilon,\nu) > \omega_{mid_{b-1}}(\upsilon,\nu); \\ 0, \text{при } \omega_{\text{mid}_b}(\upsilon,\nu) \le \omega_{mid_{b-1}}(\upsilon,\nu); \end{cases} \quad (2.19)$$

де $\omega_{\text{mid}_b}(\upsilon,\nu)$ – середньочастотний коефіцієнт ДКП *b-го* блоку;

polarity – оператор полярності.

При інших значеннях елементів залишкової маски відповідно змінюється й вираз (2.19). При обчисленні полярності порівняння значення коефіцієнта ДКП поточного блоку зі значеннями відповідних коефіцієнтів декількох сусідніх блоків у більшості випадків дозволяє, крім збільшення рівня захищеності стеганосистеми від зламу, одержати менше перекручування контейнера. Після одержання відображень полярності *P* для всіх блоків контейнера, проводиться виявлення тих коефіцієнтів ДКП, які вимагають модифікації для приховання окремого піксела псевдовипадково переставленого ЦВДЗ. Пошук проводиться відповідно до залишкової маски шляхом зміни поточної полярності (оператор XOR або знак "©" – додавання за модулем 2):

$$\begin{aligned} &\hat{P} = XOR(P, W_{sort});\\ &\hat{P} = \left\{ \hat{p}(\upsilon,\nu); 1 \le \upsilon \le \frac{8 \cdot A}{X}; 1 \le \nu \le \frac{8 \cdot Z}{Y} \right\}, \end{aligned} \qquad (2.20)$$

де

$$\hat{p}(\upsilon,\nu) = \begin{cases} 1 - p(\upsilon,\nu), & \text{при} \quad w_{sort}(\upsilon,\nu) = 1; \\ p(\upsilon,\nu), & \text{при} \quad w_{sort}(\upsilon,\nu) = 0. \end{cases} = p(\upsilon,\nu) \oplus w_{sort}(\upsilon,\nu).$$

Далі на основі масивів полярності *P* для кожного блоку контейнера формують масив:

$$\hat{\Omega}_{mid} = \exp and(\hat{P}), \quad \text{при} \quad \sum_{\upsilon,\nu} \left[w_{sort}(\upsilon,\nu) - \hat{w}_{sort}(\upsilon,\nu) \right]^2 < \eta.$$

Наприклад, задаючись початковим коефіцієнтом $\hat{w}_{sort}(\upsilon_1,\nu_1) = w_{sort}(\upsilon_1,\nu_1)$, необхідно додавати/віднімати коефіцієнти сусідніх блоків (відповідно до залишкової маски) таким чином, щоб, провівши згодом операцію, аналогічну, можна було одержати відповідну полярність $\hat{p}(\upsilon_1,\nu_1)$.

Далі варто перейти до наступних коефіцієнтів, змінюючи тільки ті з них, які не будуть впливати на полярність попередньо оброблених коефіцієнтів.

Для того щоб зменшити деградацію зображення (як результат вбудовування ЦВДЗ), автори методу пропонують обчислювати полярність для абсолютних значень коефіцієнтів ДКП, що дозволить гарантовано зберегти знак (плюс або мінус) модифікованого коефіцієнта. Крім того, для збільшення стійкості стеганосистеми до JPEG-компресії із

втратами повинен бути врахований ефект квантування, що використовується в технології JPEG. На рис. 2.17а, наведена таблиця квантування яскравості, запропонована стандартом JPEG, що найчастіше викликає помітні перекручування (так звані "артефакти") зображення. На рис. 2.17б зображена інша таблиця квантування, використовувана в більшості сучасних програм, що працюють із JPEG. Видно, що значення при цьому майже вдвічі менше відповідних їм у попередній таблиці.

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

а)

8	6	5	8	12	20	26	31
6	6	7	10	13	29	30	28
7	7	8	12	20	29	35	28
7	9	11	15	26	44	40	31
9	11	19	28	34	55	52	39
12	18	28	32	41	52	57	46
25	32	39	44	52	61	60	51
36	47	48	49	56	50	52	50

б)

Рис. 2.17. **Приклади таблиць квантування яскравості**

Заснована на таблиці квантування полярність становить результат обчислення різниці між квантованими й згодом деквантованими коефіцієнтами ДКП відповідних блоків. Для тривіального випадку, коли порівняння ведеться з коефіцієнтами попереднього блоку, формула здобуває вигляд:

$$P_b = \begin{cases} 1, \quad \text{при} \quad \left[\dfrac{\left|\omega_{mid_b}(\upsilon,\nu)\right|}{Q_{mid}(\upsilon,\nu)}\right]\cdot Q_{mid}(\upsilon,\nu) > \left[\dfrac{\left|\omega_{mid_{b-1}}(\upsilon,\nu)\right|}{Q_{mid}(\upsilon,\nu)}\right]\cdot Q_{mid}(\upsilon,\nu); \\ 0, \quad \text{при} \quad \left[\dfrac{\left|\omega_{mid_b}(\upsilon,\nu)\right|}{Q_{mid}(\upsilon,\nu)}\right]\cdot Q_{mid}(\upsilon,\nu) \le \left[\dfrac{\left|\omega_{mid_{b-1}}(\upsilon,\nu)\right|}{Q_{mid}(\upsilon,\nu)}\right]\cdot Q_{mid}(\upsilon,\nu), \end{cases}$$

де $Q_{mid}(\upsilon,\nu)$ – значення результата квантування для СЧ-коефіцієнта з координатами (υ,ν); квадратні дужки вказують на те, що вертається ціла частина від результату розподілу. При цьому у випадку атаки квантування попередній облік ефекту останнього значно збільшує ймовірність правильного розпізнання ознак пікселей при добуванні. Однак оскільки квантування має тенденцію до зведення значень

багатьох коефіцієнтів у нуль (що особливо характерно більш високочастотних коефіцієнтів), деяка частина СЧ-коефіцієнтів ДКП також у результаті буде рівнятися нулю. Крім того, для збереження встановленої полярності й після проведення квантування тим же значенням повинні бути модифіковані не тільки визначені СЧ-коефіцієнти в поточному блоці, але й у всіх сусідніх блоках відповідно до маски залишковості.

Вбудовування шляхом модифікації відносин між значеннями коефіцієнтів у межах одного блоку

Для подолання описаних вище технічних труднощів. Хсу й Ву запропонували замість порівняння зі СЧ-коефіцієнтами ДКП сусідніх блоків використовувати DC-коефіцієнт поточного блоку. У цьому випадку:

$$P_b = \begin{cases} 1, \quad \text{при} \quad \left[\dfrac{|\omega_{mid_b}(\upsilon,\nu)|}{Q_{mid}(\upsilon,\nu)}\right]\cdot Q_{mid}(\upsilon,\nu) > \left[\dfrac{|\omega_b(1,1)|}{\Psi\cdot Q(1,1)}\right]\cdot Q(1,1); \\ 0, \quad \text{при} \quad \left[\dfrac{|\omega_{mid_b}(\upsilon,\nu)|}{Q_{mid}(\upsilon,\nu)}\right]\cdot Q_{mid}(\upsilon,\nu) \le \left[\dfrac{|\omega_b(1,1)|}{\Psi\cdot Q(1,1)}\right]\cdot Q(1,1), \end{cases}$$

де Ψ масштабний коефіцієнт; Q(1-1) значення квантування для DC-коефіцієнта.

Зворотне перетворення блоків контейнера

Модифіковані матриці СЧ-коефіцієнтів ($\hat{\Omega}$) відображаються в загальні матриці коефіцієнтів ДКП (Q) (оператор **put):**

$$\hat{\Omega} = put(\hat{\Omega}_{mid}).$$

До результату проведеного об'єднання застосовується зворотне ДКП (оператор **IDCT):**

$$\hat{C} = IDCT(\hat{\Omega}).$$

Добування ЦВДЗ із контейнера

Процес добування вимагає наявності оригінального зображення-контейнера, зображення з убудованим ЦВДЗ, а також зображення, що виступає в ролі ЦВДЗ.

Обидва зображення (оригінальне – C, і досліджуване на наявність убудованого ЦВДЗ – C^*) піддаються прямому ДКП:

$$\Omega = FDCT(C), \quad \Omega^* = FDCT(C^*).$$

З отриманих масивів коефіцієнтів ДКП проводиться виділення матриць СЧ-коефіцієнтів, які, у свою чергу, використовуються для одержання шаблонів полярності:

$$\Omega_{mid} = reduce(\Omega), \quad P = polarity(\Omega_{mid});$$
$$\Omega^*_{mid} = reduce(\Omega^*), \quad P^* = polarity(\Omega^*_{mid}).$$

Застосовуючи до отриманих масивів полярностей операцію додавання за модулем 2, одержують двійкові дані (поки ще переставлені в просторі й псевдовипадково змішані):

$$W^*_{sort} = XOR\ (P, P^*),$$

де $w^*_{sort}(\upsilon,\nu) = p(\upsilon,\nu) \oplus p^*(\upsilon,\nu)$.

Виконується зворотна просторова перестановка блоків отриманих даних (оператор re-permute). Індекси відповідних пар блоків і даних, що витягаються, можуть бути отримані або шляхом їхнього зчитування з попередньо збережених у файлі на етапі вбудовування ЦВДЗ, або ж безпосередньо при вбудовуванні шляхом аналогічних дій над зображенням-оригіналом і зображенням-ЦВДЗ:

$$W^*_{md} = re - permute\ (W^*_{sort}).$$

Аналогічно проводиться зворотна псевдовипадкова перестановка даних в отриманому масиві: $W^* = re - permute\ (W^*_{sort})$.

У розглянутому алгоритмі можна виділити три особливості, які можна використовувати як секретний ключ:

1) початкове число генератора ПСЧ, що буде визначати перший елемент псевдовипадкової перестановки (будь-яке ціле число з діапазону [1, *A***Z* – 1]);

2) вибір СЧ-коефіцієнтів ДКП (необхідно вибрати 64*A**Z/(X*Y)* коефіцієнти з 64-х для кожного блоку, отже, за кожним блоком можна закріпити свій набір коефіцієнтів);

3) алгоритм відомості (відображення) обраних СЧ-коефіцієнтів в окрему матрицю (на рис. 2.18 показаний тільки один з можливих способів такого відображення).

Розглянемо приклад реалізації методу Хсу й Ву у системі MathCAD.

Крок 1

Нехай зображення-контейнер і зображення-ЦВДЗ становлять графічні файли *C*.bmp і *W*.bmp, відповідно (рис. 2.18):

C:= READBMP("C.bmp");

W := READBMP("W.bmp").

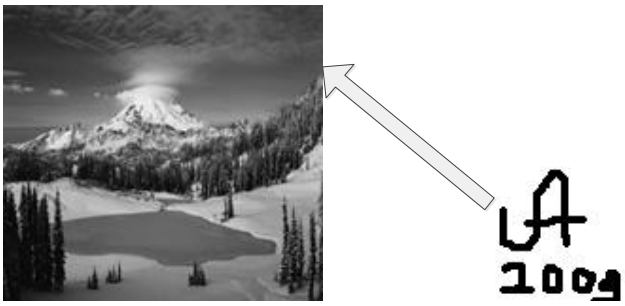
При цьому відповідні характеристики зазначених зображень становлять:

X := rows(C), X = 128;

Y:= cols(C), Y = 128;

A:= rows(W), A = 64;

Z:= cols(W), Z = 64.



C W

Рис. 2.18. **Зображення-оригінал та прихований у ньому ЦВДЗ**

Крок 2

Проводимо нормування масиву ЦВДЗ:

$$W := \frac{}{round\,(W \div \max(W))} \rightarrow$$

Елементи вихідного масиву W при цьому приймають значення 0 або 1.

Крок 3

Розмірність блоків, на які розбивається контейнер, приймаємо рівною *N* := 8. Кількість отриманих у цьому випадку блоків: *KG := X Y – r – N . KG* = 256. Розмірність, що повинні мати блоки ЦВДЗ Для одержання їх кількості *KW = Кс*, становить *n := A – N – Гх, n* = 4.

Виконуємо перевірку кількості одержуваних блоків ЦВДЗ: *KW := A – Z – n , KW* = 256 блоків.

Крок 4

Розбиття контейнера *C* на *Кс* блоків розмірністю *Nx* виконуємо за допомогою програмного модуля (2.21).

```
B_C := | c1 ← 1
       | c2 ← N
       | for b ∈ 1..X_C
       |     r1 ← mod[N·(b − 1) + 1, X]
       |     r2 ← r1 + N − 1
       |     B_(C_b) ← submatrix(C, r1, r2, c1, c2)
       |     if r2 = X
       |         c1 ← c1 + N
       |         c2 ← c2 + N
       | B_C
```

(2.21)

Крок 5

Псевдовипадкову перестановку елементів ЦВДЗ проведемо в тому порядку, що був запропонований Хсу й Ву. По-перше, проведемо індексацію кожного піксела ЦВДЗ (від 1 до *A* – *Z* = 4096), для чого просто розгорнемо масив ЦВДЗ у вектор (програмний модуль (2.22)).

```
W_vec := | W_vec ← W^⟨1⟩
         | for z ∈ 2..Z
         |   W_vec ← stack(W_vec, W^⟨z⟩)
```

(2.22)

По-друге, отримані індекси розставимо в довільному (псевдо-випадковому) порядку, для чого використовуємо *лінійний регістр зсуву зі зворотним зв'язком* (ЛРЗЗЗ, або LFSR – Linear Feedback Shift Register). Як відомо, ЛРЗЗЗ складається із двох частин: власне регістра зсуву й функції зворотного зв'язку (рис. 2.19) [12]. Регістр зсуву становить послідовність бітів (розрядів), кількість яких *d* визначається довжиною регістра зсуву. Зворотний зв'язок є сумою за модулем 2 певних бітів регістра (ці біти називаються *відвідною послідовністю).*

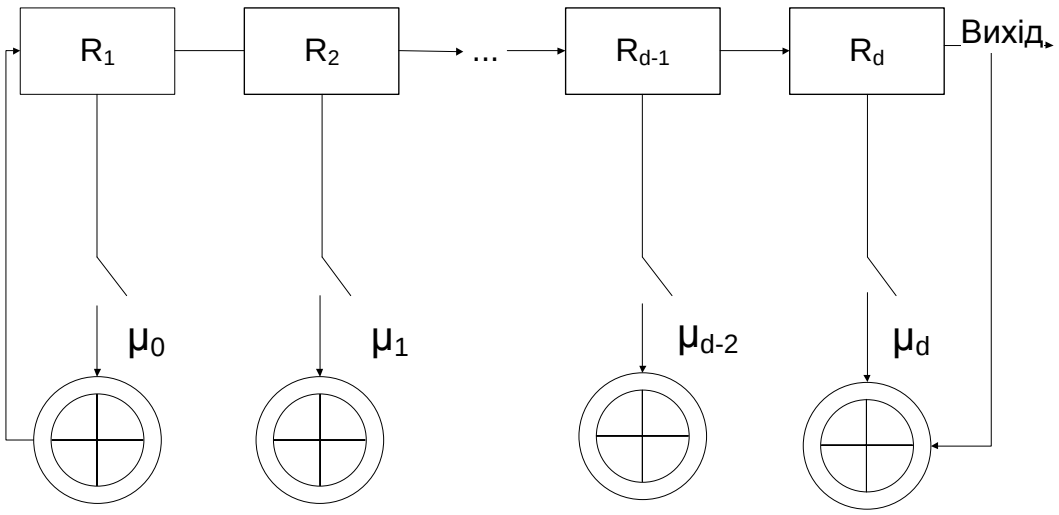


Рис. 2.19. **Узагальнений лінійний регістр зсуву зі зворотним зв'язком**

Теоретично, бітовий ЛРЗЗЗ може перебувати в одному з *2* внутрішніх станів, тобто може генерувати ПВП із періодом у *T= 2* біти. Всі *T* внутрішніх станів регістр пройде тільки при певних відвідних послідовностях. Такі ЛРЗЗЗ мають максимальний період, а отриманий при цьому результат називають *M*-послідовністю*.*

На рис. 2.19 значення (*i* = 0, 1, ..., *d)* є ваговими коефіцієнтами полінома, асоційованого з послідовністю

$$\rho(x) = \mu_0 \cdot x^0 + \mu_1 \cdot x^1 + \mu_2 \cdot x^2 + \ldots + \mu_{d-2} \cdot x^{d-2} + \mu_{d-1} \cdot x^{d-1} + \mu_d \cdot x^d.$$

Якщо μ = 1, то відповідний ключ замкнуть. У випадку *μ* = 0 – розімкнуть. Невдале включення суматорів у ланцюг зворотного зв'язку може привести до одержання ПВП, період повторення якої буде меншим максимально можливого при наявній розрядності регістра. Для того щоб конкретний ЛРЗЗЗ мав максимальний період, поліном *p(x)* повинен бути примітивним за модулем 2, тобто не розкладатися на добуток двійкових поліномів меншого степеня. При цьому коефіцієнти μ_0 й μ_1 завжди рівняються 1, інші коефіцієнти обраного полінома й будуть визначати схему формувача ПВП. У нашому випадку, для перестановки чисел у діапазоні від 1 до *A – Z* необхідним і достатнім є кількість розрядів регістра *d '.= log(A – Z, 2), d* = 12. При цьому період повторення ПВП складе *A – Z* – 1 = 4095. Для *d*-розрядного регістра як примітивний поліном за модулем 2 виберемо такий: p(x) = 1 + $x + x^2 + x^8 + x^{10}$*.* Цей і деякі інші можливі види примітивних поліномів степеня *d* = 12 наведені в табл. 2.4.

Таблиця 2.4

Приклади примітивних за модулем 2 поліномів степеня d = 12

x^0	x^1	x^2	x^3	x^4	x^5	x^6	x^7	x^8	x^9	x^{10}	x^{11}	x^{12}
1	1	1	0	0	0	0	0	1	0	0	0	1
1	1	1	0	0	0	0	0	0	0	1	0	1
1	0	1	0	0	0	0	1	1	0	0	0	1
1	0	0	1	1	0	1	0	0	1	1	0	1
1	0	0	1	0	0	1	1	1	0	0	1	1
1	1	0	1	1	0	0	0	0	1	0	1	1

ЛРЗЗЗ, що має *d* розрядів, реалізує програмний модуль (2.23), у якому аргумент s визначає початковий стан регістра (у десятковому поданні) – довільне ціле число в межах від 1 до $A - Z - 1$.

```
Vrnd(s) := | μ ← (1 1 0 0 0 0 0 1 0 0 0 1)^T
           | for i ∈ 1..2^d − 1
           |   if i = 1
           |     R(dec_i) ← s
           |     R_bin ← D2B[R(dec_i)]
           |   if i ≥ 2
           |     bit ← 0
           |     for j ∈ 1..d
           |       bit ← R(bin_j) ⊕ bit  if μ_j = 1
           |     R ← R_bin
           |     for j ∈ 1..d
           |       R(bin_j) ← R_(j−1)  if j > 1
           |       R(bin_j) ← bit  if j = 1
           |     R(dec_j) ← B2D(R_bin)
           | r_(2^d−1) ← 2^d
           | r
```

(2.23)

На початку модуля задається вектор вагових коефіцієнтів примітивного полінома *p(x)* для елементів відвідної послідовності (для наочності даний вектор наведений у вигляді матриці-рядка з наступним транспонуванням). Циклом зміни і проводиться зміна стану регістра.

Кожний *i*-ий стан із двійкового формату конвертується в десятковий і зберігається як значення відповідного елемента вектора Rdec – оскільки період послідовності,що генерується даним регістром, рівняється *2d – 1*, а псевдовипадкова перестановка буде застосовуватися до вектора, кількість елементів якого рівняється *A – Z = 2d*, наприкінці модуля до сформованого вектора Rdec дописується ще один елемент, значення якого враховує верхній граничний індекс елементів вектора *Wvec (2d).*

Одержання масиву *Vrnd* індексів елементів вектора Wvec, розставлених у псевдовипадковому порядку, дозволяє надалі провести генерування пар координат (за рядками і стовпцями) кожного піксела шляхом перетворення послідовності ПВЧ у двовимірну послідовність. Це, у свою чергу, уможливлює після псевдовипадкового вибору елемента з вектора *Wvec* помістити його значення в обумовлений згенерованою парою координат елемент масиву, розмірність якого ідентична розмірності ЦВДЗ.

Вищенаведена процедура реалізована в програмному модулі (2.25), у якому для кожного елемента *M*-послідовності обчислюються індекси *a* й *z* елемента масиву *Wrnd*, у який заноситься поточний елемент вектора *Wvec*-функція trunc(jt) повертає цілу частину від аргументу £, відкидаючи його мантису;

функція mod(£, *m)* повертає залишок від розподілу *k* на *m.* Додатком одиниці врахована можливість повернення відзначеними функціями нульового результату.

$$
\begin{aligned}
W_{rnd} := \;& \zeta \leftarrow Vrnd(s) \\
& \text{for} \quad i \in 1..A \cdot Z \\
& \quad a \leftarrow \operatorname{trunc}\left(\frac{\zeta_i - 1}{A}\right) + 1 \\
& \quad z \leftarrow \operatorname{mod}(\zeta_i, Z) + 1 \\
& \quad W_{(rnd_{a,z})} \leftarrow W_{(vec_i)} \\
& W_{rnd}
\end{aligned}
\tag{2.24}
$$

Результат виконання модуля (2.24) при *s* := 12 наведений на рис. 2.20.

W_{rnd} 255

Рис. 2.20. **Результат псевдовипадкової перестановки елементів**

Крок 6.

Модуль розбиття масиву ЦВДЗ на *NW* блоків розмірністю *n*×*n* за своєю будовою аналогічний модулю (2.25). Різниця полягає у такому:

• змінна, котрій привласнюється результат виконання модуля (як, власне, і відповідна змінна в тілі модуля), позначається як *BW* (замість БС);

• виділення блоків проводиться з масиву *Wrncj* (замість ІЗ);

• замість розмірності масиву *N* використовується розмірність *n*;

• відповідно, змінюється й граничне значення індексу рядка (замість *X* – значення *A*). Загальна кількість блоків, на яку розбивався контейнер (*Кс*), з обліком такої ж їх кількості у ЦВДЗ, можна не змінювати.

Крок 7

За допомогою програмних модулів (2.25) і (2.26) формуємо таблиці результатів сортування блоків контейнера (за значеннями стандартного відхилення елементів блоків) і блоків ЦВДЗ (за кількістю значущих елементів).

$$T_C := \left| \begin{array}{l} \text{for } b \in 1..\zeta_C \\ \quad \left| \begin{array}{l} T_{b,1} \leftarrow b \\ \sigma \leftarrow \sqrt{\frac{1}{N^2}\cdot\sum_{i=1}^{N}\sum_{j=1}^{N}\left[B_{\left[(c_v)_{i,j}\right]}\right]^2 - \left[\frac{1}{N^2}\cdot\sum_{i=1}^{N}\sum_{j=1}^{N}B_{\left[(c_v)_{i,j}\right]}\right]^2} \\ T_{b,2} \leftarrow \sigma \end{array} \right. \\ \text{reverse}(\text{csort}(T,2)) \end{array} \right. \qquad (2.25)$$

У перший стовпець таблиць характеристик блоків контейнера (*Тс*) і ЦВДЗ вносяться порядкові індекси досліджуваних блоків. У другий – результат обчислення, відповідно, стандартного відхилення (*а*) і кількості значущих (одиничних) елементів (E_1). Після формування таблиць, останні сортуються за значеннями другого стовпця (функція csort(*Т*,2)).

```
T_W := | for b ∈ 1..ζ_W
       |   T_{b,1} ← b
       |   Σ1 ← 0
       |   for υ ∈ 1..n
       |     for ν ∈ 1..n
       |       Σ1 ← Σ1 + [B((w_b))]_{υ,ν}
       |   T_{b,2} ← Σ1
       | reverse(csort(T,2))
```

(2.26)

Фрагмент результату сортування для обраних контейнера й ЦВДЗ наведений у табл. 2.5.

Таблиця 2.5

Приклад сортування блоків контейнера та ЦВДЗ

№ п/п	Індекси блоків контейнера	Значення станд. відхилення	Індекси блоків ЦВДЗ	Кількість значущих елементів
1	232	66,676	84	15
2	54	65,551	183	15
3	55	65,236	185	15
…	…	…	…	…
64	45	44,565	152	12
65	190	44,394	153	12
66	98	44,318	154	12
…	…	…	…	…
128	81	33,652	169	11
129	79	33,462	117	11
130	209	33,287	245	11
…	…	…	…	…
192	175	22,564	48	10
193	29	22,396	193	10
194	239	21,151	194	10
…	…	…	…	…
254	3	0	138	7
255	1	0	50	7
256	241	0	46	6

Шляхом виділення перших стовпців з масивів T_C і T_W зіставляємо індекси блоків контейнера з індексами блоків ЦВДЗ:

$$T_{\sum} := augment\left(\left(T_C\right)^{\langle 1\rangle}, \left(T_W\right)^{\langle 1\rangle}\right)$$

Крок 8

Відповідно до отриманого масиву наведених у відповідність один з одним індексів, проводимо перестановку блоків ЦВДЗ у порядку, що відповідає даній відповідності. Реалізацію даного етапу здійснює модуль (2.27).

$$W_{sort} := \left| \begin{array}{l} \text{for } b \in 1..\zeta_C \\ \quad \left| \begin{array}{l} i1 \leftarrow T_{(\Sigma_{b,1})} \\ i2 \leftarrow T_{(\Sigma_{b,2})} \\ W_{(sort_{i1})} \leftarrow B_{(W_{i2})} \end{array} \right. \\ W_{sort} \end{array} \right. \tag{2.27}$$

Як приклад наведемо результат виконання програмного модуля (М.61) для першого рядка табл. 2.5:

$$B_{(W_{84})} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \qquad W_{(sort_{232})} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

Крок 9

Використовуючи програмні модулі (2.26, 2.27) виконуємо пряме ДКП блоків контейнера.

З отриманих масивів коефіцієнтів ДКП, які мають розмірність *Nx*, витягаємо тільки середньочастотні коефіцієнти (див. рис. 2.20), паралельно звертаючи їх у масив *n*×*n*.

Перед початком проведення операції добування формується масив координат виділюваних СЧ-коефіцієнтів (2.28):

$$\theta := \left| \begin{array}{l} \text{for } i \in 1..N \\ \quad \text{for } j \in 1..N \\ \quad\quad \text{if } \Theta_{i,j} > 0 \\ \quad\quad\quad \left| \begin{array}{l} \theta_{\left\lfloor (\Theta_{i,j})_1 \right\rfloor} \leftarrow i \\ \theta_{\left\lfloor (\Theta_{i,j})_2 \right\rfloor} \leftarrow j \end{array} \right. \\ \theta \end{array} \right. \tag{2.28}$$

У цьому випадку масив 6 буде містити 16 рядків, елементи кожного з яких несуть інформацію про індекси рядка й стовпця відповідного СЧ-коефіцієнта в масиві 0 (рис. 2.21).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	8	8	7	7	6	6	5	5	4	4	4	3	3	2	2	1
2	1	2	2	3	3	4	4	5	4	5	6	6	7	7	8	8

Рис. 2.21. **Таблиця координат СЧ-коефіцієнтів ДКП**

За допомогою програмного модуля (2.29), відповідно до таблиці (рис. 2.21), для кожного блоку b виконується формування матриці Ω_{mid} обраних для модифікації коефіцієнтів.

$$
\begin{aligned}
&\Omega_{mid} := \left|\begin{array}{l}
\text{for } b \in 1..\zeta_C \\
\quad \left|\begin{array}{l}
q \leftarrow 1 \\
\text{for } \upsilon \in 1..n \\
\quad \text{for } \nu \in 1..n \\
\qquad \left|\begin{array}{l}
\Omega\left(mid_{\upsilon,\nu}\right) \leftarrow \left(\Omega_b\right)_{\left(\theta_{q,1}\right),\left(\theta_{q,2}\right)} \\
q \leftarrow q + 1
\end{array}\right. \\
\Omega\left(mid_b\right) \leftarrow \Omega_{mid}
\end{array}\right. \\
\Omega_{mid}
\end{array}\right.
\end{aligned}
\tag{2.29}
$$

Крок 10

Проводимо обчислення масиву полярностей блоків контейнера. При цьому в основу програмного модуля (2.21) покладена реалізація виразу (2.29). Для створення більш стійкого вбудовування ЦВДЗ даний модуль можна модифікувати відповідно до (2.30) шляхом внесення відповідної заміни.

$$
\begin{aligned}
&P := \left|\begin{array}{l}
\text{for } b \in 1..\zeta \\
\quad \left|\begin{array}{l}
\Delta \leftarrow \Omega\left(mid_b\right) - \Omega\left[mid\left(\zeta_C\right)\right] \quad \text{if } b = 1 \\
\Delta \leftarrow \Omega\left(mid_b\right) - \Omega\left[mid_{(b-1)}\right] \quad \text{if } b > 1 \\
\Delta \leftarrow \overrightarrow{\left[\text{trunc}\left[\frac{\left|\Omega\left(mid_b\right)\right|}{Q_{mid}}\right] \cdot Q_{mid}\right]} - \text{trunc}\left[\frac{\left(\Omega_b\right)_{1,1}}{\psi \cdot Q_{1,1}}\right] \cdot Q_{1,1} \\
\text{for } \upsilon \in 1..n \\
\quad \text{for } \nu \in 1..n \\
\qquad \left|\begin{array}{l}
P'_{\upsilon,\nu} \leftarrow 1 \quad \text{if } \Delta_{\upsilon,\nu} > 0 \\
P'_{\upsilon,\nu} \leftarrow 0 \quad \text{if } \Delta_{\upsilon,\nu} \le 0
\end{array}\right. \\
P_b \leftarrow P'
\end{array}\right. \\
P
\end{array}\right.
\end{aligned}
\tag{2.30}
$$

Крок 11

Відповідно до виразу проводимо зміну поточного масиву полярності відповідно до значень елементів переставленого ЦВДЗ – модуль (2.31).

$$P' := \left| \begin{array}{l} \text{for} \quad b \in 1..\zeta_C \\ \quad P'_b \leftarrow \overrightarrow{\left[P_b \oplus W\left(sort_b\right) \right]} \\ P' \end{array} \right. \tag{2.31}$$

Крок 12

На підставі поточних матриць СЧ-коефіцієнтів (Ω_{mid}) формуємо нові Ω_{mid}, причому перетерплюють зміни ті елементи первинної матриці, по координатах яких виконується нерівність виду $P`_{\upsilon,\nu} \neq P_{\upsilon,\nu}$.

Крок 13

Модифіковані для кожного блоку матриці СЧ-коефіцієнтів (Ω_{mid}) відображаються в загальні матриці коефіцієнтів ДКП (Ω') – програмний модуль (2.32). При цьому також використовується таблиця координат середньочастотних коефіцієнтів ДКП.

$$\Omega' := \left| \begin{array}{l} \text{for} \quad b \in 1..\zeta_C \\ \quad \left| \begin{array}{l} \Omega'' \leftarrow \Omega_b \\ q \leftarrow 1 \\ \text{for} \quad \upsilon \in 1..n \\ \quad \text{for} \quad \nu \in 1..n \\ \quad \quad \left| \begin{array}{l} \Omega''\left(\theta_q,1\right),\left(\theta_q,2\right) \leftarrow \left(\Omega'_{mid}\right)_{\upsilon,\nu} \\ q \leftarrow q + 1 \end{array} \right. \\ \Omega'_b \leftarrow \Omega'' \end{array} \right. \\ \Omega' \end{array} \right. \tag{2.32}$$

Крок 14

Після об’єднання до модифікованих матриць необхідно застосувати операцію зворотного ДКП (модуль (2.34)) і сформувати на основі отриманих блоків загальний масив зображення-контейнера (модуль (2.35)).

$$
\begin{aligned}
B' := \;& \text{for } b \in 1..\zeta_C \\
& \quad \text{for } x \in 0..N-1 \\
& \quad\quad \text{for } y \in 0..N-1 \\
& \quad\quad\quad B''_{x+1,\,y+1} \leftarrow \frac{1}{\sqrt{2\cdot N}} \cdot \sum_{\upsilon=0}^{N-1} \sum_{\nu=0}^{N-1} \left[Z(\upsilon) \cdot Z(\nu) \cdot \left(\Omega'_b\right)_{\upsilon+1,\,\nu+1} \cdot \cos\left[\frac{\pi \cdot \upsilon \cdot (2\cdot x+1)}{2\cdot N}\right] \cdot \cos\left[\frac{\pi \cdot \nu \cdot (2\cdot y+1)}{2\cdot N}\right]\right] \\
& \quad B'_b \leftarrow B'' \\
& B'
\end{aligned}
\tag{2.33}
$$

Отримане при цьому зображення з убудованим ЦВДЗ при певних обставинах може істотно втратити яскравість, що викликано декількома причинами: по-перше, для спрощення програмних модулів не була проведена оптимізація вбудовування за формулою (2.31); по-друге, сусідні блоки контейнера можуть мати досить різні значення інтенсивностей і, відповідно, СЧ-коефіцієнтів ДКП, що викликає необхідність при побудові алгоритму (2.29) істотно змінювати значення цих коефіцієнтів для задоволення поставлених умов. У комплексі ці дві причини викликають появу пікселей контейнера, яскравість яких після проведення зворотного ДКП виходить за межі діапазону [0, 255]. Останнє усувається шляхом нормування значень елементів масиву CA наприкінці модуля (2.34), що і викликає зниження загальної яскравості зображення.

$$
\begin{aligned}
C' := \;& C' \leftarrow B'_1 \\
& \text{for } b \in 2..\frac{X}{N} \\
& \quad C' \leftarrow \text{stack}\left(C', B'_b\right) \\
& C'' \leftarrow 0 \\
& \text{for } b \in \frac{X}{N}+1..\zeta_C \\
& \quad C'' \leftarrow B'_b \;\text{ if }\; C''=0 \\
& \quad C'' \leftarrow \text{stack}\left(C'', B'_b\right) \;\text{ otherwise} \\
& \quad \text{if } \text{mod}\left(b, \frac{X}{N}\right) \\
& \quad\quad C' \leftarrow \text{augment}(C', C') \\
& \quad\quad C'' \leftarrow 0 \\
& C' \leftarrow \frac{C + \left|\min(C)\right|}{\max\left(C' + \left|\min(C)\right|\right)} \cdot 255
\end{aligned}
\tag{2.34}
$$

Результати вбудовування ЦВДЗ у контейнер шляхом модифікації відносин між значеннями коефіцієнтів сусідніх блоків і в межах одного блоку наведені, відповідно, на рис. 2.22а та б.

а) б)

Рис. 2.22. **Контейнери з ЦВДЗ**

Крок 15

Розглянемо процес добування ЦВДЗ із зображення-контейнера. Як було зазначено вище, для добування ЦВДЗ, крім контейнера *C*, можливо, убудованим водяним знаком (*C*`), необхідна наявність оригінального (незаповненого) контейнера (*C*) і зображення ЦВДЗ (*W*). Зображення *C* й *C\** розбиваємо на блоки, використовуючи програмний модуль (2.35). Помітимо, що при розбитті *C*` у модулі необхідно провести відповідну заміну змінних на такі, які характеризують саме це зображення.

До кожного блоку оригінального зображення й зображення, досліджуваного на наявність ЦВДЗ, застосовуємо операцію прямого ДКП (модулі (2.26, 2.27)). На основі отриманих матриць коефіцієнтів ДКП (Ω і Ω') формуємо матриці СЧ-коефіцієнтів Ω_{mid} і Ω'_{mid} (модуль (2.32)), які використовуємо при обчисленні шаблонів полярності P й P* (модуль (2.31)). Шляхом поблочного додавання за модулем 2 отриманих матриць полярностей одержуємо двійкові дані, які, у тому випадку, якщо контейнер дійсно містить ЦВДЗ, відповідають переставленим у просторі й псевдовипадково змішаним елементам ЦВДЗ. Операцію підсумовування виконуємо за допомогою програмного модуля (2.35).

$$W'_{sort} := \left| \begin{array}{l} \text{for} \quad b \in 1..\zeta_{C'} \\ \quad W'_{(sort_b)} \leftarrow \overrightarrow{\left(P_b \oplus P'_b\right)} \\ W'_{sort} \end{array} \right. \qquad (2.35)$$

Формуємо масив B^*_W індексів зіставлених пар блоків контейнера й оригінального ЦВДЗ, на підставі якого виконуємо зворотну просторову перестановку блоків масиву W*sort – програмний модуль (2.36).

$$
\begin{aligned}
&B'_W := \left|\begin{array}{l}
\text{for}\quad b \in 1..\zeta_{C'} \\
\quad \left|\begin{array}{l} s1 \leftarrow T\left(\Sigma_{b,1}\right) \\ s2 \leftarrow T\left(\Sigma_{b,2}\right) \\ B'\left(W_{s2}\right) \leftarrow W'\left(sort_{s1}\right) \end{array}\right. \\
B'_W
\end{array}\right.
\end{aligned}
\qquad (2.36)
$$

Просторово переставлені блоки B^*_W поєднуємо в загальний масив *W*sort* (2.37), елементи якого гіпотетично є псевдовипадково перемішаними елементами оригінального ЦВДЗ.

$$
\begin{aligned}
&W'_{rnd} := \left|\begin{array}{l}
W'_{rnd} \leftarrow B'_W \\
\text{for}\quad b \in 2..\dfrac{A}{n} \\
\quad W'_{rnd} \leftarrow \text{stack}\left[W'_{rnd}, B'\left(W_b\right)\right] \\
W'_{rnd} \leftarrow 0 \\
\text{for}\quad b \in \dfrac{A}{n} + 1..\zeta_W \\
\quad \left|\begin{array}{l}
W''_{rnd} \leftarrow B'\left(W_b\right) \quad \text{if}\ W''_{rnd} = 0 \\
W''_{rnd} \leftarrow \text{stack}\left[W''_{rnd}, B'\left(W_b\right)\right] \quad \text{otherwise} \\
\text{if}\ \text{mod}\left(b, \dfrac{A}{n}\right) = 0 \\
\quad \left|\begin{array}{l} W''_{rnd} \leftarrow \text{augment}\left(W'_{rnd}, W''_{rnd}\right) \\ W''_{rnd} \leftarrow 0 \end{array}\right.
\end{array}\right. \\
W'_{rnd}
\end{array}\right.
\end{aligned}
\qquad (2.37)
$$

Використовуючи програмний модуль (2.37), проводимо зворотну псевдовипадкову перестановку даних, використовуючи (2.38) для одержання ПСЧ, на основі яких генерується пара координат елемента в масиві W^*_{rnd}, значення якого привласнюється *i*-му елементу вектора W^*_{vec}.

$$
\begin{aligned}
W'_{vec} := \;& E \leftarrow Vrnd(s) \\
& \text{for} \quad i \in 1..A \cdot Z \\
& \quad a \leftarrow trunc\left(\frac{E_i - 1}{A}\right) + 1 \\
& \quad z \leftarrow mod(E_i, Z) + 1 \\
& \quad W'_{(vec_i)} \leftarrow W'_{(rnd_{a,z})} \\
& W'_{vec}
\end{aligned}
\tag{2.38}
$$

Отриманий у результаті виконання (2.38) вектор обертаємо в масив *W**, що має розмірність оригінального ЦВДЗ (модуль (2.39)).

$$
\begin{aligned}
W' := \;& \text{for} \quad z \in 1..Z \\
& W'^{\langle z \rangle} \leftarrow submatrix\left[W'_{vec}, (z-1) \cdot A + 1, z \cdot A, 1, 1\right]
\end{aligned}
\tag{2.39}
$$

Графічне подання витягнутих ЦВДЗ, вбудовування яких у контейнер було проведено шляхом зміни відносин між значеннями коефіцієнтів ДКП, відповідно, сусідніх блоків і в межах одного блоку, зображене на рис. 2.23а та б.

а) б)

Рис. 2.23. **ЦВДЗ вилучені з контейнера**

Відмітимо, що при порівнянні отриманих результатів з результатами інших методів, варто брати до уваги, що в контейнер вбудовувалася інформація, піксельний обсяг якої був усього лише в 4 рази менше обсягу контейнера.

2.9. Приховування даних у нерухомих зображеннях за допомогою методів розширення спектра

Метод прямого розширення спектра дискретних сигналів

Розглянемо основні математичні положення, що лежать в основі методів розширення спектрів і прихованої передачі інформації із цифрових каналів зв’язку, введемо деякі важливі визначення й математичні позначення з теорії дискретних сигналів [12; 17].

Дискретний сигнал – інформаційний сигнал, який представляється у вигляді окремих значень, взятих за часом. Далі будемо розглядати дискретний сигнал як двійкову псевдовипадкову послідовність (ПВП) $\Phi_i = (\varphi_{i_0}, \varphi_{i_1}, ..., \varphi_{i_{n-1}})$ довжини n з множини $\Phi = \{\Phi_0, \Phi_1, ..., \Phi_{M-1}\}$ потужності $|\Phi| = M$ [6].

Елементи двійкової ПВП ухвалюють одне зі значень:

$$\varphi_{i_z} = \begin{cases} +1 \\ -1 \end{cases}, \; z = 0, ..., n-1.$$

Кореляція дискретних сигналів – статистичний взаємозв'язок двох або декількох випадкових сигналів. Математичним заходом кореляції двох дискретних сигналів $\Phi_i, \Phi_j \in \Phi$ служить коефіцієнт кореляції ρ [5; 35]:

$$\rho = \frac{1}{\mathrm{n}} \sum_{z=0}^{n-1} \Phi_{iZ} \Phi_{jZ} \; . \qquad (2.40)$$

Два сигнали Φ_i, Φ_j називаються *ортогональними*, якщо коефіцієнт кореляції $\rho = 0$. Якщо $\rho \approx 0$ будемо називати сигнали Φ_i і Φ_j квазіортогональними [19; 36].

У роботах [12; 17] досліджені різні підходи до побудови дискретних сигналів з поліпшеними ансамблевими й кореляційними властивостями: похідні ортогональні системи сигналів (ПОСС); нелінійні похідні кодові послідовності (НПКП); повні кодові кільця (ПКК); послідовності Голда. У табл. 2.6 як приклад наведені результати досліджень ансамблевих і кореляційних властивостей похідних систем сигналів [14].

Таблиця 2.6

Ансамблеві й кореляційні властивості дискретних сигналів

n	64	128	256	512	1024	2048	4096
M	102	103	105	106	108	109	1010
ρ	$2.1/\sqrt{\mathrm{n}}$	$2.5/\sqrt{\mathrm{n}}$	$2.9/\sqrt{\mathrm{n}}$	$3.2/\sqrt{\mathrm{n}}$	$3.5/\sqrt{\mathrm{n}}$	$3.8/\sqrt{\mathrm{n}}$	$3.9/\sqrt{\mathrm{n}}$

Як випливає з наведених у табл. 2.6 даних, застосування похідних ортогональних дискретних сигналів дозволяє при збереженні низької корельорованості дискретних послідовностей ($\rho \approx 0$) суттєво підвищити потужність *M* ансамблів дискретних сигналів, зі зростанням довжини послідовностей ця тенденція посилюється.

У сучасній теорії цифрового зв'язку більші ансамблі слабко-корельорованих дискретних сигналів використовуються для побудови

широкосмугових вірогідних систем передачі даних. Передані повідомлення в таких каналах отримують від шумоподібних послідовностей, а за рахунок великої потужності ансамблів дискретних сигналів і прямого розширення частотного спектра забезпечується висока імітостійкість, вірогідність і скритність цифрових каналів зв'язку [12; 14;17].

Для передачі даних у широкосмуговій системі зв'язку інформаційний сигнал $x = \begin{cases} +1 \\ -1 \end{cases}$ модулюється за допомогою його множення на розширювальний кодовий сигнал $g = \Phi_i \in \Phi$ – псевдовипадкову послідовність із розглянутих вище ансамблів дискретних сигналів. Оскільки кодовий сигнал за своїми статистичними властивостями подібний шуму, то отриманий розширений сигнал слабко відрізняється від шумів у каналах зв'язку, що й дозволяє здійснити приховану передачу.

$$y = xg(t) = x(t)\Phi_i . \qquad (2.41)$$

При прийманні в демодуляторі отриманий сигнал $y' = y + e$ як суміш переданої послідовності у і як той, що відбувся в каналі зв'язку помилок *e*, множиться на синхронізовану копію розширювального сигналу *g*. Інакше кажучи, на прийомній стороні здійснюється обчислення коефіцієнта кореляції (2.40), значення якого визначає правило ухвалення рішення:

$$\rho = n\sum_{z=0}^{n-1} x(t)\Phi_{i_z}\Phi_{i_z} + \frac{1}{n}\sum_{z=0}^{n-1} e(t)\Phi_{i_z} . \qquad (2.42)$$

Враховуючи псевдовипадковість послідовностей Φ_i, використовуваних у якості *g*, другим доданком у правій частині рівності можна зневажити (кількість "+1" зразково дорівнює кількості "–1"), тобто

$$\rho \approx \rho x \frac{1}{n}\sum_{z=0}^{n-1}(\Phi_{i_z})^2 = x(t),$$

таким чином, значення інформаційного сигналу на прийомній стороні визначається за виразом

$$x = \begin{cases} +1, при \rho(y'(t), g(t)) \approx +1; \\ -1, при \rho(y'(t), g(t)) \approx -1; \end{cases} \qquad (2.43)$$

де знак "$\approx$" припускає наявність помилок *e*, викликаних природними або навмисними перешкодами в каналі зв'язку.

Структурна схема передачі інформації з використанням прямого розширення спектра наведена на рис. 2.24.

Часова тривалість немодульованого сигналу *x* дорівнює *T* а його частота відповідно – $F = \frac{1}{T}$. Передача модульованого сигналу *y* при тій же часовій тривалості *T* приведе до розширення частотного спектра переданого сигналу, пропорційно кількості елементів псевдовипадкової послідовності, тобто пропорційно довжині *n*:

$$F(y(t)) = n\frac{1}{T} = nF.$$

Проте використання прямого розширення спектра частот переданого сигналу забезпечує одночасну передачу багатьох інших інформаційних сигналів у тій же смузі частот. Це випливає із взаємної ортогональності (квазіортогональності) застосовуваних ансамблів дискретних сигналів.

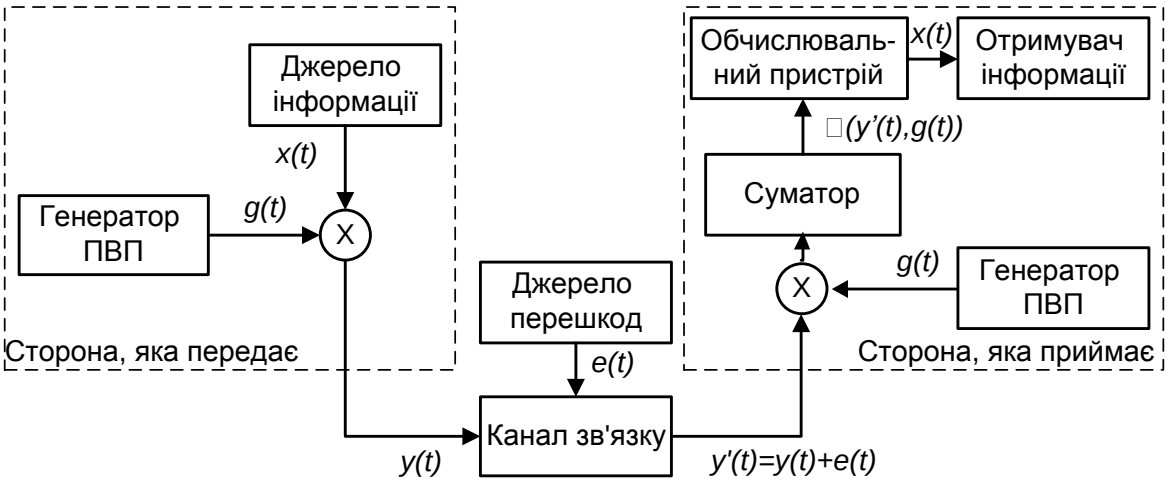


Рис. 2.24. **Структурна схема передачі інформації з використанням прямого розширення спектра**

Дійсно, якщо на прийомній стороні прийнята аддитивна суміш $\sum_l y_l(t)$ декількох модульованих сигналів, тоді обчислення коефіцієнта кореляції дасть таке:

$$\rho(\sum_l y_l(t), g(t)) = \frac{1}{n}\sum_l \sum_{z=0}^{n-1} x_l(t)\Phi_{l_z}\Phi_{i_z}. \qquad (2.44)$$

Але всі послідовності з множини *Φ* мають низьке значення взаємної кореляції, тобто при $l \neq i$ маємо $\rho \approx 0$ (для ортогональних сигналів маємо рівність $\rho \approx 0$). Отже, всіма доданками при $l \neq i$ правої частини

рівності (2.44) можна зневажити. Звідси, при наявності в аддитивній сумі $\sum_l y_l(t)$ дискретного сигналу $\Phi_{l=i}$ маємо вираз (2.42) і відповідне правило ухвалення рішення (2.43).

Метод прямого розширення спектра знайшов практичне використання в системах цифрового зв'язку з кодовим розподілом каналів (CDMA), де для кожного абонента інформаційного обміну використовуються унікальні розширювальні кодові сигнали з ансамблю ортогональних (квазіортогональних) дискретних послідовностей. Таким чином, для розрізнення кодових сигналів і розподілу відповідних абонентських каналів використовувані ПВП повинні бути слабкокорельованими один з одним, в ідеальному випадку – ортогональними.

Так, наприклад, у стандарті CDMA IS-95 для кодового розподілу каналів використовуються ортогональні дискретні сигнали Уолша – Адамара (2.44). Вони утворюються з рядків матриці Адамара H_i, сформованої за рекурентним правилом:

$$H_i = \begin{bmatrix} H_{i-1} & H_{i-1} \\ H_{i-1} & -H_{i-1} \end{bmatrix}, \; H_0 = [1]. \qquad (2.45)$$

Багаторазове повторення правила (2.45) дозволяє сформувати матрицю Адамара будь-якого розміру, кратного чотирьом. Рядки сформованих матриць взаємоортогональні, тобто їх скалярний добуток дорівнює нулю. Ці рядки й становлять ансамбль

$$\Phi = \{\Phi_0, \Phi_1, ..., \Phi_{M-1}\}$$

дискретних сигналів Уолша – Адамара

$$\Phi_i = (\varphi_{i_0}, \varphi_{i_1}, ..., \varphi_{i_{n-1}}),$$

де n – розмірність сформованої матриці H_i (в IS-95 використано H_i з $n = 64$).

Для передачі інформації один з рядків $\Phi_i \in \Phi$ матриці Адамара ставиться у відповідність абонентському каналу, наприклад, для зв'язку між базовою станцією й конкретним абонентом. Модуляція здійснюється за правилом (2.41), тобто для передачі інформаційної "1" посилає рядок Φ_i, для "0" – посилає послідовність, сформовану шляхом логічного заперечення Φ_i (її інверсна копія).

Для виділення сигналу на прийомній стороні використовується кореляційний приймач, тобто обчислюється коефіцієнт кореляції (2.45). При точному збігу початку послідовності, яка отримана, й наявної копії Φ_i спостерігаються піки кореляційної функції позитивної й негативної полярностей – залежно від переданого біта. Тобто детектування сигналу відбувається в такий спосіб:

$$x_i = polarity = \begin{cases} "1", при\ polarity > 0; \\ "0", при\ polarity < 0; \\ сторонній\ сигнал, при\ polarity = 0, \end{cases}$$

де *polarity – полярність* піка кореляційної функції.

Таким чином, застосування ортогональних систем дискретних сигналів Уолша – Адамара дозволяє забезпечити високоефективний широкосмуговий цифрової зв'язок. Кількість утворених абонентських каналів зв'язку не може перевищувати потужності *M* ансамблю сигналів, у цьому випадку вона не перевищує розмірності матриці H_i, $M = n$. Інакше кажучи, максимальна кількість можливих ортогональних кодів обмежена їх довжиною. Для розглянутого прикладу маємо $M = 64$ (по специфікації IS-95 утворюються 61 абонентський і 3 службових канали). У цьому змісті квазіортогональні дискретні сигнали (з $M > n$) мають незаперечну перевагу (див. табл. 2.6), їх застосування потенційно дозволить суттєво підвищити абонентську потужність системи зв'язку. Крім того, для розглянутих сигналів функція взаємної кореляції дорівнює нулю лише при відсутності часового зсуву між послідовностями. Як наслідок, такі сигнали використовуються лише в синхронних системах і переважно в прямих каналах (від базової станції до абонента).

Метод Сміта – Комісклі

У методі Сміта – Комісклі [64], як і в розглянутій вище системі зв'язку із прямим розширенням спектра, інформаційне повідомлення побітно модулюється шляхом множення на ансамбль ортогональних сигналів. Потім промодульоване повідомлення вбудовується в контейнер-зображення.

Введемо умовні позначки й математичні співвідношення, які за аналогією з розглянутими вище системами широкосмугового цифрового зв'язку дозволять досліджувати особливості побудови й інформаційного обміну даних у стеганосистемі.

Представимо, що інформаційне повідомлення *m* яке необхідно вбудувати в цифрової контейнер-зображення, у вигляді блоків m_i рівної довжини, тобто

$$m = (m_0, m_1, ..., m_{N-1}),$$

де кожний блок m_i – послідовність (вектор) з n бітів:

$$m_i = \left(m_{i_0}, m_{i_1}, ..., m_{i_{n-1}}\right).$$

Контейнер-зображення будемо розглядати як масив даних *C* розмірністю *K·L*, розбитий на підблоки розміром *k·ln*. У якості елементів масиву *C* можуть виступати, наприклад, растрові дані використовуваного зображення.

Секретними ключовими даними є набір базисних функцій

$$Key = \Phi = \{\Phi_0, \Phi_1, ..., \Phi_{M-1}\},$$

де всі базисні функції

$$\Phi_i = (\varphi_{i_0}, \varphi_{i_1}, ..., \varphi_{i_{n-1}})$$

– взаємоортогональні дискретні сигнали з довжиною, рівною розміру n блоку повідомлення m_i, таким чином, для будь-яких $i, j \in [0,...,M-1]$ виконується рівність

$$\rho(\Phi_i, \Phi_j) = \frac{1}{n}\sum_{z=0}^{n-1} \Phi_{l_z} \Phi_{i_z} = \begin{cases} 1, \text{при } i = j; \\ 0, \text{при } i \neq j. \end{cases}$$

Формальна графічна вистава інформаційного повідомлення, контейнера-зображення й ключових даних наведені на рис. 2.25.

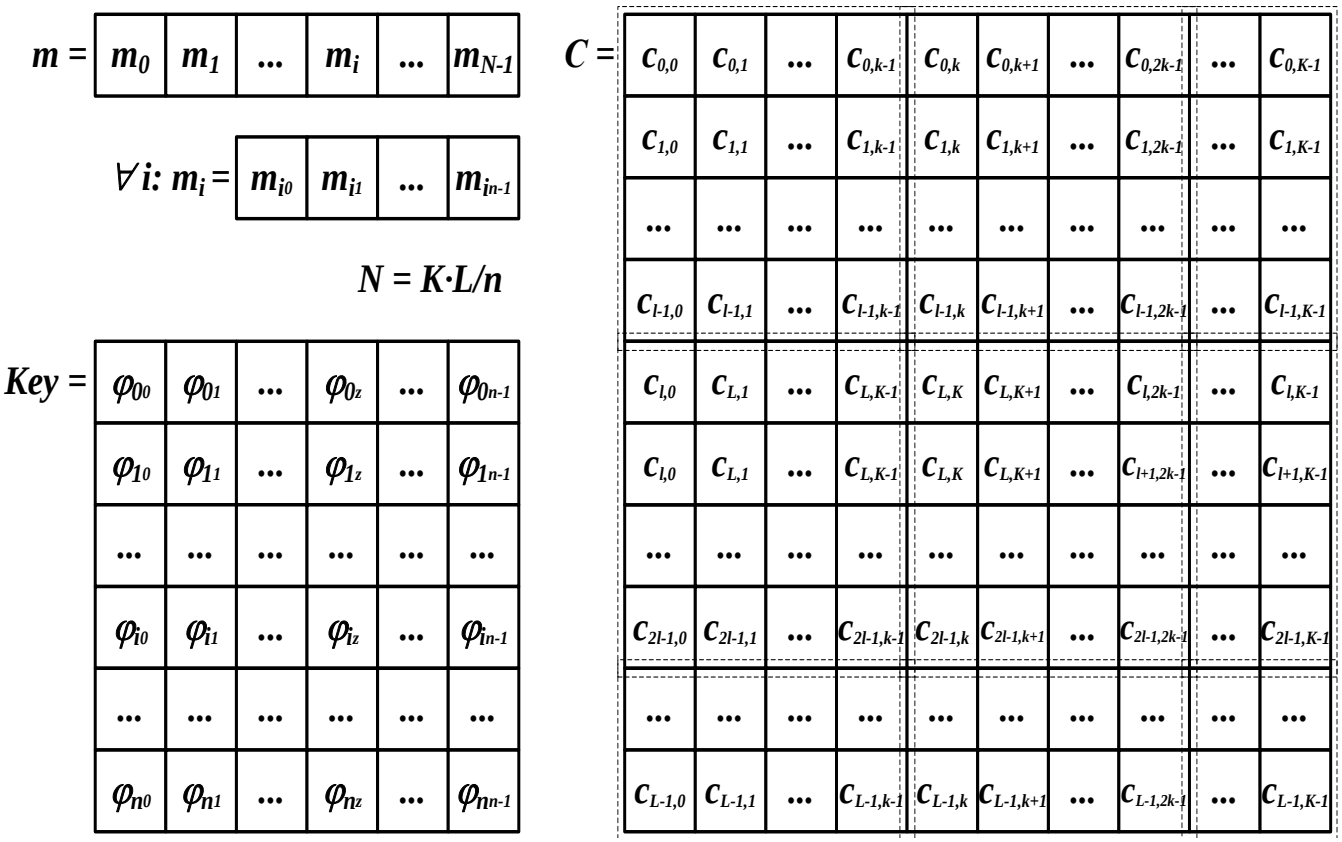


Рис. 2.25. **Формальна вистава інформаційного повідомлення, контейнера-зображення й ключових даних**

Метою стеганоперетворення інформації є вбудовування кожного окремого блоку повідомлення m_i у відповідний блок контейнера-зображення. Таким чином, у блок даних цифрового зображення розмірністю $K \cdot L$ елементів потенційно може бути вбудоване $K \cdot L$ блоків інформаційного повідомлення, тобто до $K \cdot L$ бітів.

Розбиття контейнера на блоки може бути довільним, однак, як показує практика, найбільш доцільним є двовимірна розбивка, яка наведена на рис. 2.25. В якості ключових даних (масиву базисних функцій $Key = \Phi$) будемо використовувати розглянуті вище ансамблі ортогональних дискретних сигналів Уолша – Адамара.

Вбудовування інформаційного повідомлення здійснюється в такий спосіб. Кожний блок повідомлення $m_i, i = 0,...,N-1$ зіставляється з окремим блоком контейнера-зображення. Кожний інформаційний бітів блоку $m_{i_J}, m_0,...,n$ представляється у вигляді інформаційного сигналу

$$m = \begin{cases} +1, m_{i_J} = 1 \\ -1, m_{i_J} = 0 \end{cases}$$

і за аналогією з (2.41) модулюється розширювальним кодвим сигналом (базисними функціями), тобто ПВП $\Phi_j \in \Phi$.

У результаті для кожного інформаційного блоку m_i формується модульований інформаційний сигнал

$$E_i(t) = \sum_{j=0}^{n-1} \sum_{z=0}^{n-1} m_{iJ}(t) \Phi_{jZ} \ . \qquad (2.46)$$

Отриманий блок повідомлення E_i попіксельно підсумується з підблоком контейнера. Позначимо блоки контейнера в такий спосіб:

$$\boldsymbol{C_0} = \begin{pmatrix} \boldsymbol{c_{0,0}} & \boldsymbol{c_{0,1}} & \cdots & \boldsymbol{c_{0,k-1}} \\ \boldsymbol{c_{1,0}} & \boldsymbol{c_{1,1}} & \cdots & \boldsymbol{c_{1,k-1}} \\ \cdots & \cdots & \cdots & \cdots \\ \boldsymbol{c_{l-1,0}} & \boldsymbol{c_{l-1,1}} & \cdots & \boldsymbol{c_{l-1,k-1}} \end{pmatrix}, \boldsymbol{C_1} = \begin{pmatrix} \boldsymbol{c_{0,k}} & \boldsymbol{c_{0,k+1}} & \cdots & \boldsymbol{c_{0,2k-1}} \\ \boldsymbol{c_{1,k}} & \boldsymbol{c_{1,k+1}} & \cdots & \boldsymbol{c_{1,2k-1}} \\ \cdots & \cdots & \cdots & \cdots \\ \boldsymbol{c_{l-1,k}} & \boldsymbol{c_{l-1,k+1}} & \cdots & \boldsymbol{c_{l-1,2k-1}} \end{pmatrix}, \ldots,$$

$$\boldsymbol{C_{N-1}} = \begin{pmatrix} \boldsymbol{c_{L-l-1,K-k-1}} & \boldsymbol{c_{L-l-1,K-k}} & \cdots & \boldsymbol{c_{L-l-1,K-1}} \\ \boldsymbol{c_{L-l,K-k-1}} & \boldsymbol{c_{L-l,K-k}} & \cdots & \boldsymbol{c_{L-l,K-1}} \\ \cdots & \cdots & \cdots & \cdots \\ \boldsymbol{c_{L-1,K-k-1}} & \boldsymbol{c_{l-1,k+1}} & \cdots & \boldsymbol{c_{L-1,K-1}} \end{pmatrix}.$$

Відповідні модульовані інформаційні сигнали E_i(t) представимо у вигляді двовимірного масиву даних:

$$E_i = \begin{pmatrix} E_{i_0} & E_{i_1} & \dots & E_{i_{k-1}} \\ E_{i_k} & E_{i_{k+1}} & \dots & E_{i_{2k-1}} \\ \dots & \dots & \dots & \dots \\ E_{i_{(l-1)(k-1)-k+1=n-k+1}} & E_{i_{(l-1)(k-1)-k+2=n-k+2}} & \dots & E_{i_{(l-1)(k-1)=n-1}} \end{pmatrix}, i = 0,\dots,N-1$$

Тоді стеганограма (заповнений контейнер) формується за допомогою об'єднання масивів даних $S_i, S_0, \dots, N-1$:

$$\mathrm{S_i} = \mathrm{C_i} + \mathrm{E_i} \times \mathrm{S} \qquad (2.47)$$

де G – коефіцієнт підсилення розширювального сигналу, що задає "енергію" інформаційної послідовності, що вбудовується.

Таким чином, заповнений контейнер S утворюється зі сформованих блоків $S_i, S_0, \dots, N$ – за допомогою їх об'єднання як це показане на рис. 2.24 для вихідного (порожнього) контейнера C.

На етапі витягнення даних немає необхідності володіти інформацією про первинний контейнер C. Операція декодування полягає у відновленні прихованого повідомлення шляхом проектування кожного блоку S_i, отриманого стегазображення S_i на всі базисні функції $\Phi_j \in \Phi, j = 0,\dots,n-1$. Для цього кожний блок S_i представляється у формі вектора $S_i = (S_{i_0}, S_{i_1}, \dots, S_{i_{n-1}}), i_{0, \dots, N-1}$.

Щоб витягти бітів повідомлення з i-го блоку стегазображення, необхідно обчислити коефіцієнт кореляції між Φ_j і прийнятим блоком S_i (представленого у вигляді вектора):

$$\rho(S_i, \Phi_j) = \frac{1}{n}\sum_{z=0}^{n-1} S_{i_Z} \Phi_{j_Z} = G\frac{1}{n}\sum_{z=0}^{n-1} E_{i_Z} \Phi_{j_Z} + \frac{1}{n}\sum_{z=0}^{n-1} C_{i_Z} \Phi_{j_Z}, \qquad (2.48)$$

де C_i – одномірний масив, тобто відповідний блок контейнера, представлений у формі вектора.

Припустимо, що масив C_i має випадкову статистичну структуру, тобто покладемо, що другий доданок у правій частині виразу (2.48) близький до нуля і їм можна зневажити.

Тоді маємо:

$$\rho(S_i, \Phi_j) \approx G \cdot E_i \cdot \Phi_j = G \cdot \sum_{l=0}^{n-1} \sum_{Z=0}^{n-1} m_{i_x}(t) \Phi_{l_Z} \Phi_{j_Z}. \qquad (2.49)$$

За аналогією з (2.44) відзначимо, що всі послідовності з множини Φ взаємортогональні, тобто при $l \neq j$ маємо $\rho = 0$. Отже, всіма доданками в правій частині рівності (2.49) при $l \neq j$ можна зневажити. Звідси маємо:

$$\rho(S_i, \Phi_j) \approx G \cdot m_{i_j}(t) \cdot \frac{1}{n} \sum_{Z=0}^{n-1} (\Phi_{j_Z})^2 = G \cdot m_{i_j}(t). \qquad (2.50)$$

За аналогією з виділенням корисного сигналу значення $m_{i_j}(t)$ можуть бути легко відновлені за допомогою знакової функції. Оскільки *G0in> 0* знак ρ в (2.50) залежить тільки від $m_{i_j}(t)$, звідки маємо:

$$m_{i_j}(t) = sign(\rho(S_i, \Phi_j)) = \begin{cases} -1, \textit{при } \rho(S_i, \Phi_j) < 0; \\ 1, \textit{при } \rho(S_i, \Phi_j) > 0; \\ ?, \textit{при } \rho(S_i, \Phi_j) = 0. \end{cases} \qquad (2.54)$$

Якщо $\rho = 0$ в (2.51), будемо вважати, що вбудована інформація була втрачена.

Таким чином, вискористання прямого розширення спектра дискретних сигналів дозволяє здійснити вбудовування інформаційних даних у нерухливі зображення для прихованої передачі й реалізувати таким чином стеганографічний захист інформації.

Контрольні запитання

1. Назвіть основні особливості зорової системи людини.
2. Визначте узагальнену схему впровадження даних у зображення.
3. Назвіть основні цифрові формати нерухомих зображень.
4. У чому полягають особливості комп'ютерної обробки зображень.
5. У чому полягає суть формату BMP?
6. Для чого призначений дескриптор екрана формату GIF?
7. У чому полягає сегментація зображення? Основні види.
8. Охарактеризуйте суть дискретно косинусного перетворення.
9. Назвіть основні етапи алгоритму стиску зображень JPEG.
10. У чому полягає стійкість стеганосистеми до активних атак.
11. Охарактеризуйте метод псевдовипадкового інтервалу.

12. У чому полягає метод псевдовипадкової перестановки?

13. Охарактеризуйте алгоритми Bruyndonckx, Rongen, Patchwork, Bender, Marvel.

14. У чому полягає приховування даних у просторі множини зображень? Блокове приховування, метод квантування, метод "хреста".

15. У чому полягає приховування даних у частотній множині зображень? Метод Коха – Жао та його модифікації. Приховування даних у частотній множини зображень. Метод Хсу – Ву та метод Фрідріх.

16. Охарактеризуйте приховування даних у нерухомих зображеннях за допомогою методів розширення спектра. Метод прямого розширення спектра дискретних сигналів.

17. У чому полягає суть формату GIF?

18. Охарактеризуйте суть формату TIFF.

19. У чому полягає суть формату JPEG. Назвіть основні його характеристики та властивості?

20. Охарактеризуйте атаки на стеганосистеми із застосуванням формату JPEG.

21. Назвіть основні властивості та вимоги до генераторів псевдовипадкових чисел.

22. У чому полягає суть алгоритму формування матриці Адамара?

23. Що таке ансамблі ортогональних дискретних сигналів Уолша – Адамара? Назвіть їх основні переваги.

Розділ 3. Приховування даних в аудіосигналах

3.1. Особливості слухової системи людини (ССЛ). Основні властивості ССЛ, що використовуються при приховуванні даних в аудіосигналах

Для того щоб перейти до обговорення питань впровадження інформації в аудіосигнали, необхідно визначити вимоги, які можуть бути висунуті до стеганосистем, що застосовуються для вбудовування інформації в аудіосигнали:

приховувана інформація повинна бути стійкою до наявності різних пофарбованих шумів, стиску із втратами, фільтрування, аналогово-цифрового й цифро-аналогового перетворень;

приховувана інформація не повинна вносити в сигнал перекручування, сприймані системою слуху людини;

спроба видалення приховуваної інформації повинна приводити до помітного ушкодження контейнера (для ЦВДЗ);

приховувана інформація не повинна вносити помітних змін у статистику контейнера.

Для впровадження приховуваної інформації в аудіосигнали можна використовувати методи, що застосовуються в інших видах стеганографії. Наприклад, можна впроваджувати інформацію, заміщаючи найменш значущі біти (всі або деякі). Або можна будувати стеганосистеми, ґрунтуючись на особливостях аудіосигналів і системи слуху людини.

Систему слуху людини можна представити як аналізатор частотного спектра, що може виявляти й розпізнавати сигнали в діапазоні 10 – 20000 Гц. Систему слуху людини можна змоделювати як 26 проникних фільтрів, смуга пропущення яких збільшується зі збільшенням частоти. Система слуху людини розрізняє зміни фази сигналу слабкіше, ніж зміни амплітуди або частоти.

Аудіосигнали можна розділити на три класи:

розмова телефонної якості, діапазон 300 – 3400 Гц;

широкосмугова мова 50 – 7000 Гц;

широкосмугові аудіосигнали 20 – 20000 Гц.

Практично всі аудіосигнали мають характерну рису. Кожний з них становить досить великий обсяг даних для того, щоб використовувати статистичні методи впровадження інформації. Перший з описуваних методів, розрахований на цю особливість аудіосигналів, працює в тимчасовій множині.

3.2. Цифрові формати аудіосигналів (формати WAV, WMA, MP3, AAC, OGG Vorbis). Особливості комп’ютерної обробки аудіосигналів

Опис формату MP3. Формат стиску аудіоданих MP3 (скорочення від MPEG Layer3) – один з перших популярних способів стиску звуку; розроблений німецькою компанією Fraunhofer IIS і пізніше, за підтримкою фірми THOMSON, впроваджений як частина відеоформатів MPEG1 і MPEG2; забезпечує високу якість звуку при порівняно невеликих розмірах файла.

Технічні відомості про формат MP3

Високий ступінь стиску в MP3 досягається за рахунок досить складного алгоритму кодування. Використовуються як математичні

методи компресії, так і особливості людського слуху (психоакустична модель): ефект маскування слабкого звуку однієї частоти більш голосним звуком такої ж або сусідньої частоти, зниження чутливості вуха до тихого звуку відразу після голосного, несприйнятливість до звуків нижче визначеного рівня гучності.

Потік звуку при кодуванні розбивається на рівні ділянки (фрейми). Кожний із фреймів кодується окремо зі своїми параметрами й містить заголовок, у якому ці параметри зазначені.

Стиск може бути виконано з різною якістю й відповідно розміром кінцевого файла. Ступінь стиску характеризується бітрейтом (bitrate) – кількість переданої за одиницю часу інформації. Файли MP3 звичайно закодовані з бітрейтом від 64 до 320 кілобіт у секунду (kbps або kb/s), а також зі змінним бітрейтом (VBR) – коли для кожного фрейму використовується свій, оптимальний для даної ділянки, бітрейт.

Вихідний сигнал за допомогою фільтрів розділяється на кілька частотних діапазонів, для кожного діапазону визначається величина ефекту, що маскує, від сусідніх діапазонів і попереднього фрейму, несуттєві сигнали ігноруються. Для даних, що залишилися, для кожного діапазону визначається, скількома бітами можна пожертвувати, щоб втрати були нижче величини ефекту, що маскує. На цьому робота психоакустичної моделі завершується, а підсумковий потік додатково стискується по алгоритму Хаффмана (аналогічно RAR – архіватору). При бітрейті 320 kbps застосовується тільки підсумковий стиск, без психоакустичного моделювання.

Варто мати на увазі, що різні кодеки можуть кодувати аудіосигнал по-різному, розходження особливо проявляються на високих частотах і низьких бітрейтах.

У форматі MP3 кодується стереосигнал, при цьому можливі кілька варіантів перетворення:

Dual Channel – кожний канал одержує половину потоку й кодується окремо – можливий запис двох абсолютно різних сигналів.

Stereo – кожний канал кодується окремо, але програма-кодер може використовувати вільне місце одного каналу для розміщення інформації іншого. Режим stereo виставлений за замовчуванням у більшості кодерів.

Joint Stereo (MS Stereo) – стереосигнал розкладається на загальний для обох каналів і відмінний. Має варіант MS/IS Stereo зі спрощеним відмінним сигналом.

Сильні сторони формату MP3:

високий ступінь стиску при прийнятній якості звуку;

ступінь стиску і якість може регулюватися користувачем;

фреймова структура зручна для передачі по мережі, дозволяє перехід до будь-якого місця файла;

широке поширення апаратури й програм;

особливості застосування MP3.

Незважаючи на те, що кодування в MP3 здійснюється із втратою частини вихідної інформації, при бітрейтах 256 і 320 kbps практично неможливо відрізнити на слух стислий сигнал від оригіналу, особливо при прослуховуванні на розповсюдженій аудіоапаратурі. При цьому розмір файла в найгіршому випадку буде в 4 рази менше, ніж у форматі CD-audio. Для використання в компактних плеєрах і інших пристроях з невисокою якістю акустики можна цілком використовувати бітрейт не нижче 192 kbps. Бітрейти нижче 192 kbps рекомендується використовувати для стиску сигналу з обмеженим частотним діапазоном або невисокими вимогами до вірогідності (наприклад, розмова або телепередача).

AAC (Advanced Audio Coding) – формат аудіофайла з меншою втратою якості при кодуванні, ніж MP3 при однакових розмірах.

AAC споконвічно створювався як спадкоємець MP3 з поліпшеною якістю кодування. Формат AAC, офіційно відомий як ISO/IEC 13818-7, вийшов у світ у 1997 р. як нова, сьома частина родини MPEG-2. Хоча формат для аудіофайлів AAC був зареєстрований як 7-ма частина стандарту MPEG-2, існує також формат AAC, відомий як MPEG-4 частина 3.

Перевага AAC над MP3:

частоти з 8 Гц до 96 кГц (mp3:16 Гц – 48 кГц);

до 48 звукових каналів;

більша ефективність кодування при постійному звуковому потоці;

більша ефективність кодування при звуковому потоці, що змінюється;

поліпшена обробка частот вище 16 Кгц;

більш гнучкий joint stereo.

Ogg Vorbis – це відносно новий універсальний формат аудіокомпресії, що офіційно вийшов улітку 2002 р. Він належить до того ж типу форматів, що й MP3, AAC, VQF і WMA, тобто до форматів

компресії із втратами. Психоакустична модель, використовувана в Ogg Vorbis, за принципами дій близька до MP3, але й тільки – математична обробка й практична реалізація цієї моделі докорінно відрізняються, що дозволяє авторам оголосити свій формат зовсім незалежним від всіх попередників.

Головна незаперечна перевага формату Ogg Vorbis – це його повна відкритість і вільність. Більше того, у ньому використана новітня й найбільш якісна психоакустична модель, через що співвідношення бітрейт/якість значно нижче, ніж в інших форматів. Як результат – якість звуку краще, але розмір файла менше.

У форматі є велика кількість достоїнств. Наприклад, формат Ogg Vorbis не обмежує користувача тільки двома аудіоканалами (стерео – лівий і правий). Він підтримує до 255 окремих каналів із частотою дискретизації до 192kHz і розрядністю до 32bit (чого не дозволяє жоден формат стиску із втратами), тому Ogg Vorbis чудово підходить для кодування 6-канального звуку DVD-audio. До того ж формат OGG Vorbis – sample accurate. Це гарантує, що звукові дані перед кодуванням і після декодування не будуть мати зсувів або додаткових/загублених симплів відносно один одного. Це легко оцінити, коли ви кодуєте non-stop музику (коли один трек поступово входить в інший) – у підсумку збережеться цілісність звуку.

Можливістю потокового віщання зараз нікого не здивуєш, але в цьому форматі вона закладена із самих основ. Це дає формату досить корисний побічний ефект – в одному файлі можна зберігати кілька композицій із власними тегами. При завантаженні такого файла в плеєр повинні відобразитися всі композиції, начебто їх завантажили з декількох різних файлів.

Окремо варто згадати досить гнучку систему тегів. Заголовок тегів легко розширюється й дозволяє включати тексти будь-якої довжини й складності (наприклад, текст пісні), що перемежовуються зображеннями (наприклад, фотографія обкладинки альбому). Текстові теги зберігаються в UTF-8, що дозволяє писати хоч на всіх мовах одночасно й виключає можливі проблеми з кодуваннями. Це значно зручніше різних хитрувань типу id3 тегів.

Ogg Vorbis за замовчуванням використовує змінний бітрейт, при цьому значення останнього не обмежені якимись твердими значеннями, і він може варіюватися навіть на 1 kbps. При цьому варто помітити, що

форматом жорстко не обмежений максимальний бітрейт, і при максимальних настроюваннях кодування він може варіюватися від 400 kbps до 700 kbps. Такою ж гнучкістю володіє частота дискретизації – користувачам надається будь-який вибір у межах від 2 000 Hz до 192 000 Hz.

Ogg Vorbis був розроблений співтовариством Xiphophorus для того, щоб замінити всі платні запатентовані аудіоформати. Незважаючи на те, що це наймолодший формат із всіх конкурентів MP3, Ogg Vorbis має повну підтримку на всіх відомих платформах (Windows, PocketPC, Symbian, DOS, Linux, MacOS, FreeBSD, BeOS та ін.), а також велику кількість апаратних реалізацій. Популярність на сьогоднішній день значно перевершує всі альтернативні рішення.

Варто відмітити, що Ogg Vorbis є всього лише невеликою частиною мультимедіа-проекту Ogg Squish, у який також входять вільні кодувальники: Speex – для стиску голосу; FLAC – для стиску звуку без втрат; Theora – для стиску відео.

Чому саме Ogg Vorbis?

На сьогоднішній день основними гравцями на арені аудіоформатів, крім MP3 і Ogg Vorbis, виступають також WMA і AAC. Чому ж саме Ogg Vorbis ми вважаємо найбільш оптимальним вибором? Для початку автори пропонують глянути на результати найбільш свіжих тестів на слух, проведених учасниками найбільш авторитетного в цій множині ресурсу Hydrogen Audio.

Результати тестів на бітрейтах 80 і 180 kbps

Беззастережна перемога на найбільш затребуваних швидкостях потоку 80 і 180 kbps – досить вагомий показник. Висока якість – не єдина перевага формату. Ogg Vorbis значно більш продуманий з боку технічної реалізації. Більше того, серед всіх розглянутих це єдина безкоштовна й вільна альтернатива.

Як видно, формат Ogg Vorbis – сучасне й найбільш оптимальне рішення на ринку форматів аудіокомпресії. Якщо стежити за динамікою розвитку формату, можна легко помітити, що в нього є майбутнє. Велика кількість розроблювачів із усього світу постійно розвивають формат, доводячи його до досконалості. Здавалося б, усе, що можна було зробити, уже зроблено. Однак це не так. Усе ще не розкритий весь потенціал формату. Залишається маса ще зовсім незачеплених проблем.

Просування формату

Ogg Vorbis – формат вільний. Розроблювачі не можуть займатися просуванням цього продукту. Справа залишається за користувачами. Тут кожна крапля важлива. Адже ціль велика... Як могло б бути? Кожний новий користувач формату залучає ще двох. Ті – ще чотирьох і т. д. З такими темпами формат би давно домінував над MP3. А як насправді? Усі довідаються про цей формат зовсім випадково. Хтось побачив його у своїй іграшці, хтось помітив у друга. А самі користувачі формату мовчать. Просто користуються форматом і все. А не заважало б посприяти хоч співтовариству вільних розроблювачів. Навіть 2 нових притягнутих користувача – це було б непоганим подарунком для співтовариства. І для кожного користувача окремо. Більше користувачів формату – більше попит – більше підтримка виробників апаратних плеєрів. Це дуже важливо для кожного користувача формату. Користувачі не будуть слухати Ogg Vorbis тільки на комп'ютері. Його зручно слухати й на Flash/CD/HDD/DVD плеєрі.

3.3. Приховування даних у просторій множині аудіосигналу (приховування в найменш значущому біті даних та за допомогою ехосигналів)

Тими ж авторами був запропонований метод впровадження інформації з використанням ехосигналу.

Цей метод дозволяє впроваджувати дані в сигнал прикриття, змінюючи параметри ехосигналу. До параметрів ехосигналу, що несе впроваджувану інформацію (рис. 3.1), відносяться: початкова амплітуда, час спаду й зсуву (час затримки між вихідним сигналом і його ехо). При зменшенні зсуву два сигнали змішуються. У певній точці людське вухо перестає розрізняти два сигнали, і ехо сприймається як додатковий резонанс. Цю точку важко визначити точно, тому що вона залежить від вихідного запису, типу звуку й слухача. У загальному випадку, за дослідженнями В. Бендера й Н. Моримото, для більшості типів сигналів і для більшості слухачів злиття двох сигналів відбувається при відстані між ними близько 0,001 секунди.

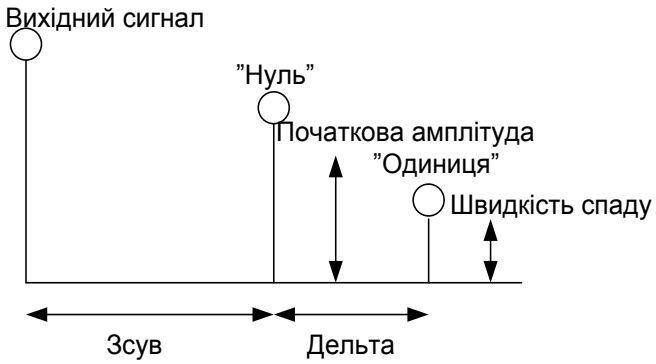


Рис. 3.1. **Параметри ехосигналу**

Кодер використовує два часи затримки: один для кодування нуля, інший для кодування одиниці. І той, і інший час затримки менше того, на якому людське вухо може розпізнати ехо. Крім зменшення часу затримки, необхідно домогтися встановлення початкової амплітуди й часу спаду для того, щоб впроваджена інформація не могла бути сприйнята системою слуху людини.

Кодування. Для простоти був обраний приклад тільки двох імпульсів (один для копіювання вихідного сигналу, інший для формування ехосигналу). Збільшення кількості імпульсів приведе до збільшення кількості відліків ехосигналів.

Нехай на рис. 3.2а показаний спосіб кодування "одиниці", а на рис. 3.2б – спосіб кодування "нуля". Упровадження даних показане на рис. 3.3.

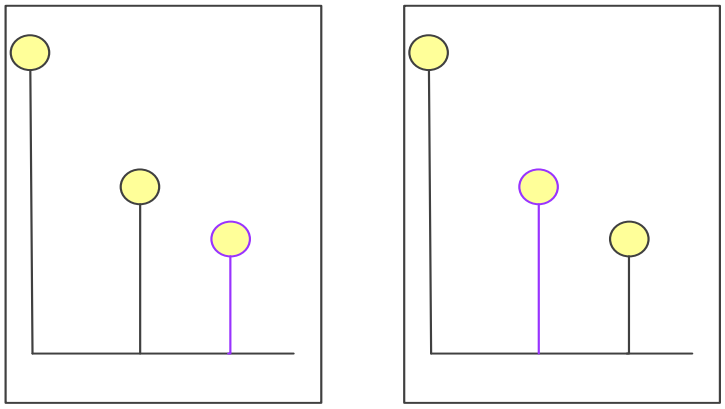


Рис. 3.2. **Кодування одного біта інформації**

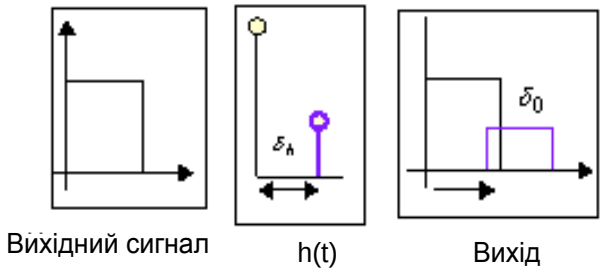


Рис. 3.3. **Упровадження одного біта інформації**

Затримка (δ_h) між вихідним сигналом і його ехо залежить від впроваджуваних у цей момент даних. Одиниці відповідає затримка (δ_1), а нулю – затримка ехосигналу (δ_0).

Для того щоб закодувати більше одного біта, вихідний сигнал розділяється на маленькі ділянки. Кожна ділянка розглядається як окремий сигнал, і в нього впроваджується один бітів інформації. Результуючий закодований сигнал (що містить трохи бітів упровадженої інформації) становить комбінацію окремих ділянок. На рис. 3.4 показаний приклад, у якому сигнал розділяється на сім ділянок – a, b, c, d, e, f, g.

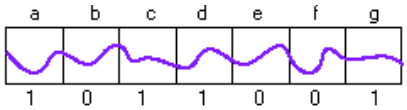


Рис. 3.4. **Поділ сигналу на ділянки**

У ділянки a, з, d, g буде впроваджена одиниця. Отже, на цих ділянках система буде функціонувати так, як показано на рис. 3.2а. Нулі будуть упроваджені в ділянки b, e, f, на цих ділянках система буде функціонувати так, як показано на рис. 3.2б.

Для досягнення мінімуму помітності спочатку створюються два сигнали: один, що містить тільки "одиниці", і інший – тільки нулі. Отримані в результаті сигнали показані на рис. 3.5.

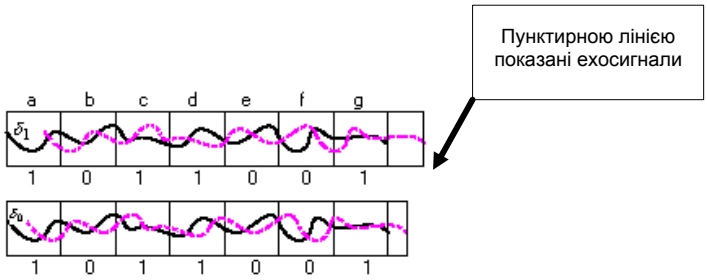


Рис. 3.5. **Сигнали, що містять тільки одне бінарне значення**

Потім створюються два перемикаючі сигнали – нульовий і одиничний (рис. 3.6). Кожний з них становить бінарну послідовність, стан якої залежить від того, який бітів повинен бути впроваджений у дану ділянку звукового сигналу.

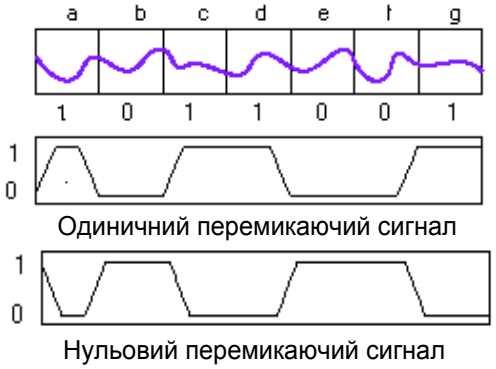


Рис. 3.6. **Перемикаючі сигнали**

Далі обчислюється сума добутків нульового сигналу, що змішує, і аудіосигналу із затримкою "нуль", а також одиничного сигналу, що змішує, і аудіосигналу із затримкою "одиниця". Інакше кажучи, коли в аудіосигнал необхідно впровадити "одиницю", на вихід подається сигнал із затримкою "одиниця", у противному випадку – сигнал із затримкою "нуль". Оскільки сума двох сигналів, що змішують, завжди дорівнює одиниці, то забезпечується гладкий перехід між ділянками аудіосигналу, у які впроваджені різні біти. Блок-схема стеганокодера показана на рис. 3.7.

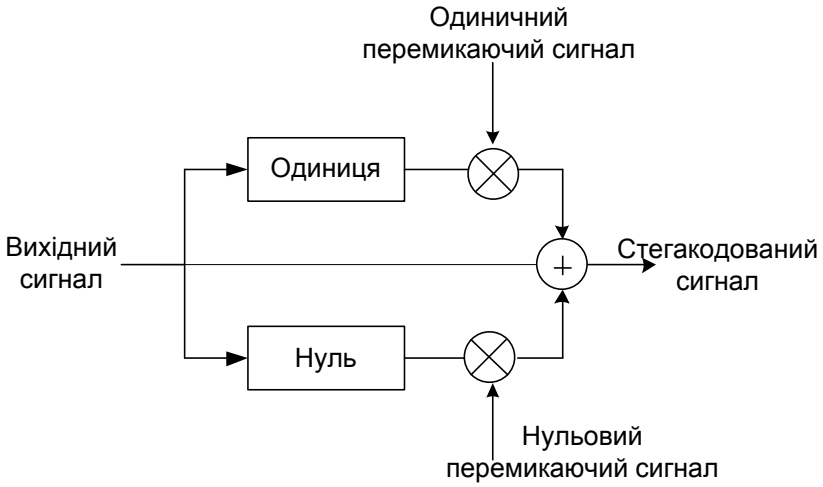


Рис. 3.7. **Блок-схема стегакодера**

Декодування. Декодування впровадженої інформації є визначенням проміжку часу між сигналом і ехо. Для цього необхідно розглянути амплітуду (у двох точках) автокореляційної функції дискретного косинусного перетворення логарифму спектра потужності (кепстра). У результаті обчислення кепстра вийде послідовність імпульсів (ехо, дубльована кожні δ секунд) (рис. 3.8).

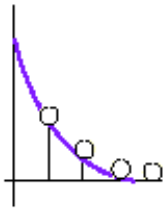
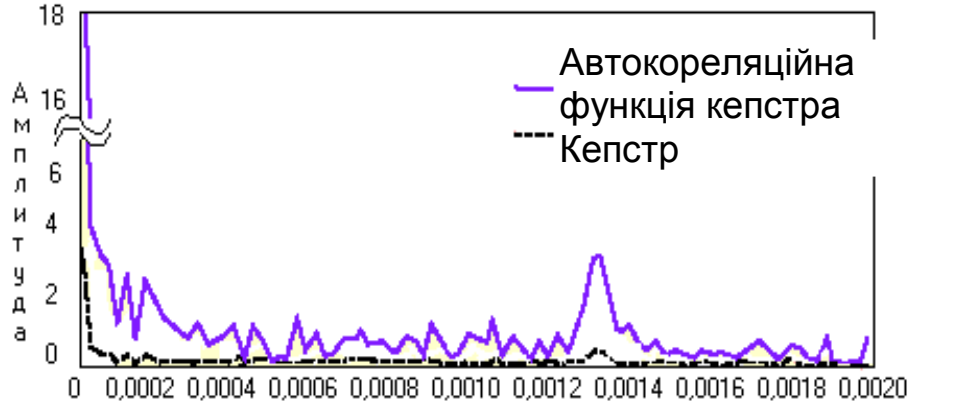


Рис. 3.8. **Результат обчислення кепстра**

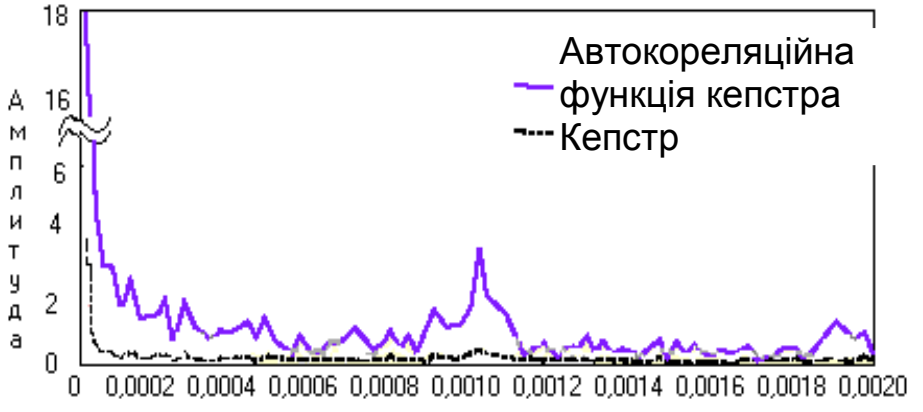
Для визначення проміжку часу між сигналом і його ехо необхідно розрахувати автокореляційну функцію кепстра.

Сплеск автокореляційної функції буде мати місце через δ_1 або δ_0 секунд після вихідного сигналу (рис. 3.9). Правило декодування засноване на визначенні проміжку часу між вихідним сигналом і сплеском

автокореляції. При декодуванні "одиниця" приймається, якщо значення автокореляційної функці через δ_1 секунд більше, ніж через δ_0 секунд, у противному випадку – "нуль".



а) Нуль



б) Одиниця

Рис. 3.9. **Поводження автокореляційної функції при різній впровадженій інформації**

За дослідженнями В. Бендера й Н. Моримото дана схема дозволяє впроваджувати 16 бітів в одну секунду аудіозапису непомітно, без втрати його якості.

3.4. Приховування даних у частотній множині аудіосигналу (фазове кодування)

Метод, що пропонує використовувати слабку чутливість системи слуху людини до незначних змін фази сигналу, був запропонований В. Бендером, Н. Моримото та ін.

Упровадження інформації модифікацією фази аудіосигналу – це метод, при якому фаза початкового сегмента аудіосигналу модифікується

залежно від упроваджуваних даних. Фаза наступних сегментів узгоджується з ним для збереження різниці фаз. Це необхідно тому, що до різниці фаз людське вухо більш відчутне. Фазове кодування, коли воно може бути застосовано, є одним з найбільш ефективних способів кодування за критерієм відношення сигнал – шум.

Процедура фазового кодування полягає в такому:

1. Звуковий сигнал $s[i], (0 \le i \le I-1)$ розбивається на серію N коротких сегментів $s_n[i]$ $(0 \le n \le N-1)$.

2. До n-го сегмента сигнала $s_n[i]$ застосовується *k*-точкове дискретне перетворення Фур'є, де ϕ = *I*/*N*, і створюються матриці фаз $\phi_n(w_k)$ і амплітуд $A_n(w_k)$ для $(0 \le k \le K-1)$.

3. Запам'ятовується різниця фаз між кожними двома сусідніми сегментами $(0 \le n \le N-1)$.

$$\Delta\phi_{n+1}(w_k) = \phi_{n+1}(w_k) - \phi_n(w_k). \qquad (3.1)$$

4. Бінарна послідовність даних представляється як $\pi/2$ і $-\pi/2$, $\phi_0^I = \phi_{data}^I$.

5. З урахуванням різниці фаз створюється нова матриця фаз для *n*>0:

$$\begin{bmatrix} (\phi_1^I(w_k) = \phi_0^I(w_k) + \Delta\phi_1(w_k)) \\ \dots \\ (\phi_n^I(w_k) = \phi_{n-1}^I(w_k) + \Delta\phi_n(w_k)) \\ \dots \\ (\phi_0^I(w_k) = \phi_{N-1}^I(w_k) + \Delta\phi_N(w_k)) \end{bmatrix} \qquad (3.2)$$

6. Стеганокодований сигнал виходить шляхом застосування зворотного дискретного перетворення Фур'є до вихідної матриці амлітуд і модифікованої матриці фаз.

Одержувачі повинні бути відомі: довжина сегмента і точки ДПФ. Перед декодуванням послідовність повинна бути синхронізована.

Недоліком цієї схеми є її низька пропускна здатність. В експериментах В. Бендера й Н. Моримото пропускна здатність каналу варіювалася від 8 до 32 бітів у секунду.

3.5. Приховування даних в аудіосигналах за допомогою методів розширення спектра

Запропонований алгоритм задовольняє більшості із пропонованих вимог, викладених вище [22]. ЦВДЗ впроваджується в аудіосигнали (послідовність 8×8-бітних або 16-бітних відліків) шляхом незначної зміни амплітуди кожного відліку. Для виявлення ЦВДЗ не потрібно вихідного аудіосигналу.

Нехай аудіосигнал складається з N відліків $x(i)$, $i = 1, ..., N$, де значення N не менше 88200 (відповідно 1 секунда для стерео-аудіосигналу, дискретизованого на частоті 44,1 кГц). Для того щоб вмонтувати ЦВДЗ, використовується функція $f(x(i), w(i))$, де $w(i)$ – відлік ЦВДЗ. Функція f повинна брати до уваги особливості системи слуху людини, щоб уникнути відчутних перекручувань вихідного сигналу. Відлік результуючого сигналу виходить у такий спосіб:

$$y(i) = x(i) + f(x(i),w(i)). \tag{3.3}$$

Відношення сигнал – шум у цьому випадку обчислюється як:

$$SNR = 10 \log_{10} \frac{\sum_n x^2(n)}{\sum_n \left[x(n) - y(n)\right]^2}. \tag{3.4}$$

Важливо відзначити, що застосовуваний у схемі генератор випадкових чисел повинен мати рівномірний розподіл. Стійкість ЦВДЗ, у загальному випадку, підвищується зі збільшенням енергії ЦВДЗ, але це збільшення обмежується зверху припустимим відношенням сигнал – шум.

Виявлення ЦВДЗ відбувається в такий спосіб. Позначимо через S таку суму:

$$S = \sum_{i=1}^{N} y(i)w(i). \tag{3.5}$$

Комбінуючи (3.3) і (3.5), одержуємо:

$$S = \sum_{i=1}^{N} \left[x(i)w(i) + f(x(i),w(i))w(i)\right]. \tag{3.6}$$

Перша сума в (3.6) дорівнює нулю, якщо числа на виході ГВЧ розподілене рівномірно й математичне очікування значення сигналу дорівнює нулю. У більшості ж випадків спостерігається деяка відмінність, позначувана Δw, котру необхідно також ураховувати.

Отже, (3.6) приймає вигляд:

$$S = \sum_{i=1}^{N-\Delta w} x(i)w(i) + \sum_{i=1}^{\Delta w} x(i)w(i) + \sum_{i=1}^{N} f(x(i),w(i))w(i). \qquad (3.7)$$

Сума $\sum_{i=1}^{N-\Delta w} x(i)w(i)$, як показано вище, приблизно дорівнює нулю. Якщо в аудіосигнал не був впроваджений ЦВДЗ, то *S* буде приблизно дорівнювати $\frac{\Delta w}{N}\sum_{i=1}^{N} x(i)w(i)$. З іншого боку, якщо в аудіосигнал був впроваджений ЦВДЗ, то *S* буде приблизно дорівнювати:

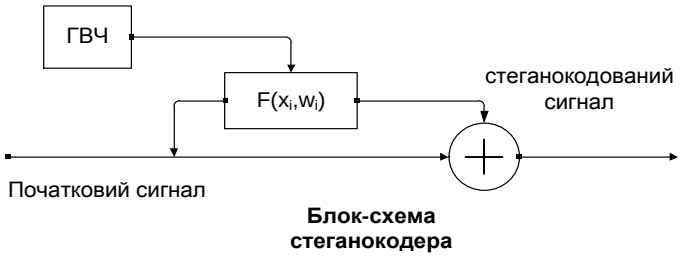
$$\frac{\Delta w}{N}\sum_{i=1}^{N} x(i)w(i) + \sum_{i=1}^{N} f(x(i),w(i))w(i).$$

Однак $x(i)$ – це вихідний сигнал, що за умовою не може бути використаний у процесі виявлення ЦВДЗ. Сигнал $x(i)$ можна замінити на $y(i)$, це приведе до заміни $\sum_{i=1}^{\Delta w} x(i)w(i)$ на $\frac{\Delta w}{N}S$, помилка при цьому буде незначною.

Отже, віднімаючи величину $\frac{\Delta w}{N}S$ з *S* і ділячи результат на $\sum_{i=1}^{N} f(y(i),w(i))w(i)$, одержимо результат *r*, нормований до 1. Детектор ЦВДЗ, використовуваний у цьому методі, обчислює величину r, що задається формулою:

$$r \hat{=} \frac{S - \frac{\Delta w}{N}|S|}{\sum_{i=1}^{N} f(y(i),w(i))w(i)}. \qquad (3.8)$$

Гранична величина виявлення теоретично лежить між 0 і 1. Досвідченим шляхом установлено, що для того, щоб визначити, чи дійсно певний ЦВДЗ перебуває в сигналі, граничне значення ЦВДЗ повинне бути вище 0,7. Якщо потрібна більша вірогідність у визначенні наявності ЦВДЗ у сигналі, граничне значення необхідно збільшити. Робота кодера й декодера наведені на рис. 3.10 [12].

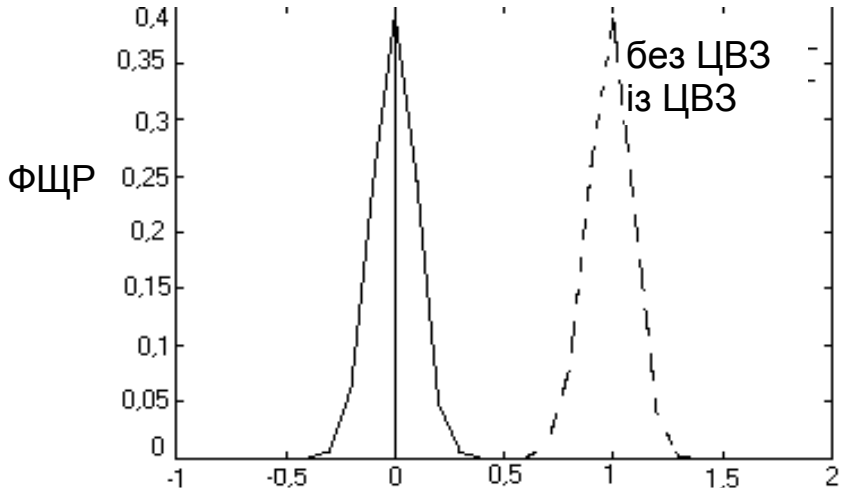


Блок-схема стеганокодера

Блок-схема стеганодекодера

Рис. 3.10. **Блок-схеми стегакодера й стеганодекодера**

На рис. 3.11 показана емпірична функція щільності ймовірності для аудіосигналу з ЦВДЗ і без ЦВДЗ. Емпірична функція щільності ймовірності аудіосигналу без ЦВДЗ показана безперервною кривою, пунктирна крива описує емпіричну функцію щільності ймовірності аудіосигналу з убудованим ЦВДЗ. Обидва розподіли були обчислені з використанням 1 000 різних значень ЦВДЗ при відношенні сигнал – шум 26 дб.



Величина знаходження (відношення сигнал – шум – 26 дб)

Рис. 3.11. **Функція щільності розподілу величини виявлення для сигналів зі ЦВДЗ і без ЦВДЗ**

Упровадження в один аудіосигнал великої кількості різних ЦВДЗ приводить до збільшення чутності перекручувань. Максимальна кількість ЦВДЗ обмежена енергією кожного з них. Декодер здатний правильно відновити кожний ЦВДЗ за умови використання кодером унікальних ключів. На рис. 3.12 показаний приклад виявлення ЦВДЗ із використанням 1 000 різних ключів, з яких тільки один – правильний [22].

При стиску аудіосигналу до 48 кб/с з'являються звукові ефекти, відчутно знижується якість сигналів зі ЦВДЗ. Стійкість алгоритму вбудовування ЦВДЗ до фільтрації перевірена застосуванням до нього ковзного фільтра середніх частот і фільтра нижніх частот. У роботі [22] перевірялася стійкість розглянутого методу впровадження інформації до стиску MPEG до швидкостей 80 кб/с і до 48 кб/с.

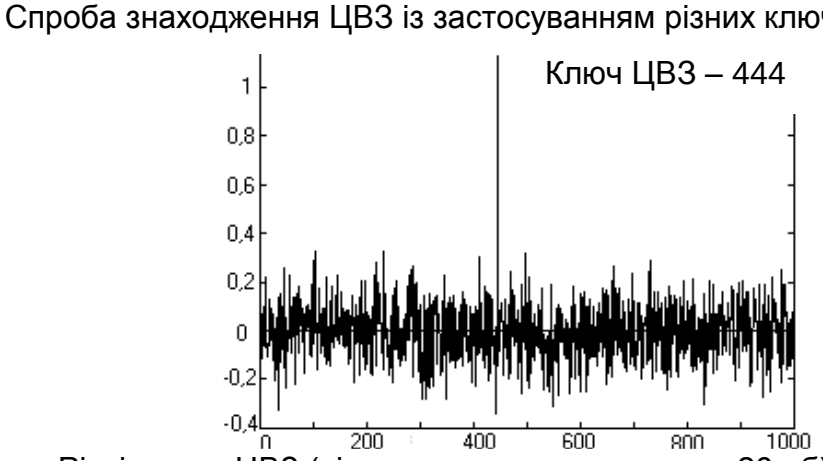


Рис. 3.12. **Розпізнавання заданого ключа збудованих ЦВДЗ**

Після відновлення при стиску до швидкості 80 кб/с можна спостерігати незначне зменшення граничної величини виявлення в аудіосигналах зі ЦВДЗ (рис. 3.13).

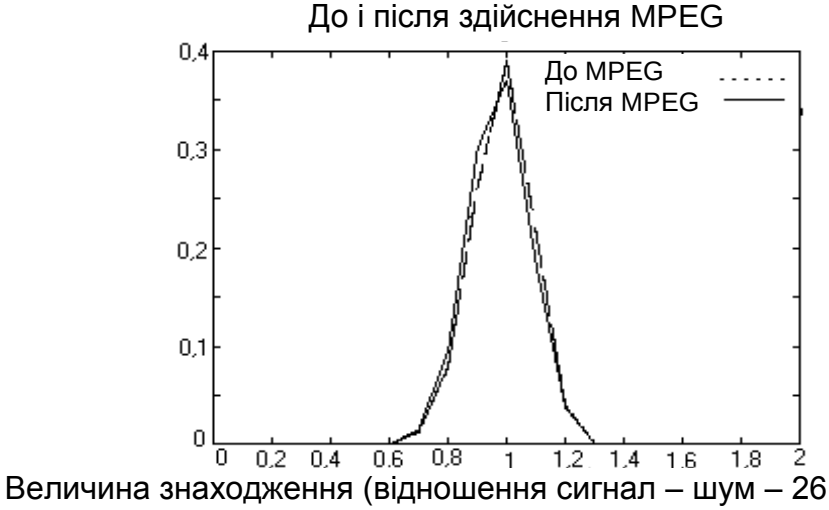


Рис. 3.13. **Вплив стиску даних на ЦВДЗ**

Аудіофайли із впровадженим ЦВДЗ профільтровані ковзним фільтром середніх частот довжини 20, що вносить в аудіоінформацію значні перекручування.

На рис. 3.14 показано, як змінюється гранична величина виявлення при застосуванні описаного фільтра. Загалом, поріг виявлення збільшується у відфільтрованих сигналах. Це відбувається через те, що функція щільності розподілу сигналів після фільтрації зрушується вправо порівняно з відносною функцією розподілу сигналів, що не піддавалися фільтрації.

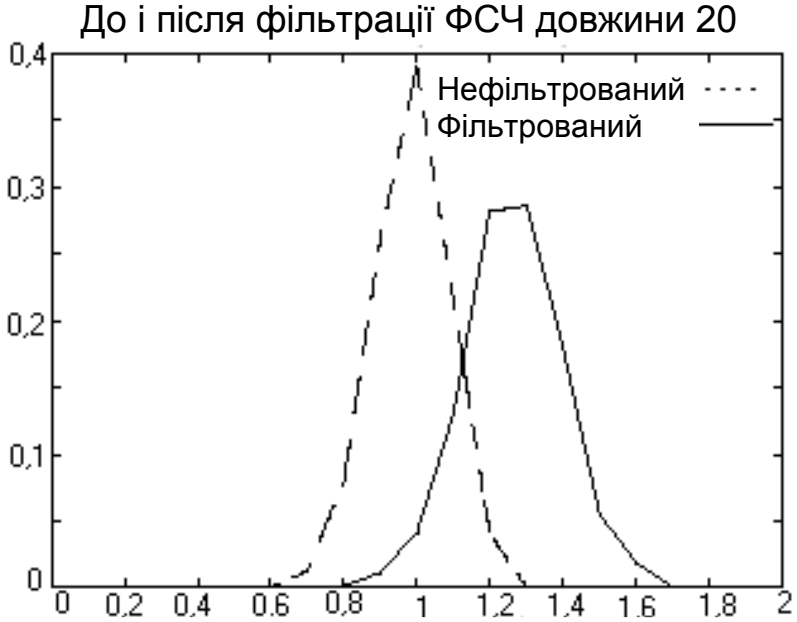


Рис. 3.14. **Вплив на ЦВДЗ застосування до аудіосигналу ковзного фільтра середніх частот**

ЦВДЗ зберігається й при застосуванні до аудіосигналу фільтра нижніх частот. Однак при фільтрації аудіосигналів зі ЦВДЗ фільтром нижніх частот Хемінга 25-го порядку із частотою зрізу 2205 Гц мало місце зменшення ймовірності виявлення наявності ЦВДЗ.

Для перевірки стійкості ЦВДЗ до передискретизації Р. Бассіа й І. Пітасом аудіосигнали були передискретизовані на частоти 22050 Гц і 11025 Гц і назад на початкову частоту. ЦВДЗ зберігався.

При переквантуванні аудиосигнала з 16-бітного в 8-бітний і назад впроваджений ЦВДЗ зберігається, незважаючи на часткову втрату інформації. На рис. 3.15 показано наскільки добре ЦВДЗ зберігається в 1000 аудиосигналах при їх переквантуванні в 8-бітні й назад в 16-бітні.

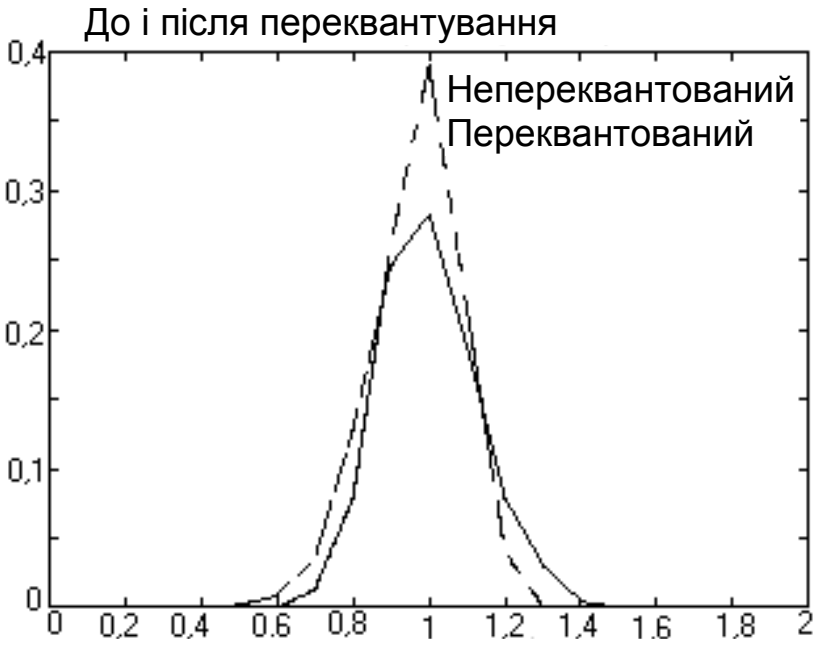


Рис. 3.15. **Вплив переквантування сигналу на ЦВДЗ**

Девіація функції щільності розподілу переквантованого сигналу збільшується, як і у випадку застосування фільтра нижніх частот, отже, має місце зменшення ефективності виявлення.

Контрольні запитання

1. Назвіть особливості слухової системи людини (ССЛ). Назвіть основні властивості ССЛ, що використовуються при приховуванні даних в аудіосигналах.

2. У чому полягає суть цифрового формату аудіосигналів WAV?

3. Охарактеризуйте сутність приховування даних у просторій множині аудіосигналу.

4. Охарактеризуйте сутність приховування в найменш значущому біті даних та за допомогою ехосигналів.

5. Охарактеризуйте сутність приховування даних у частотній множині аудіосигналу (фазове кодування).

6. Охарактеризуйте сутність приховування даних в аудіосигналах за допомогою методів розширення спектра.

7. У чому полягають особливості комп'ютерної обробки аудіо-сигналів?

8. Побудуйте блок-схему стеганокодера.

9. У чому полягає суть цифрового формату аудіосигналів OGG Vorbis?

10. Поводження автокореляційної функції при різній впровадженій інформації.

11. Побудуйте блок-схему стеганодекодера.

12. Охарактеризуйте функцію щільності розподілу величини виявлення для сигналів зі ЦВДЗ і без ЦВДЗ.

13. У чому полягає суть цифрового формату аудіосигналів MP3?

14. Охарактеризуйте суть впливу стиску даних на ЦВДЗ.

15. Охарактеризуйте суть впливу на ЦВДЗ застосування до аудіосигналу ковзного фільтра середніх частот.

16. У чому полягає суть цифрового формату аудіосигналів AAC?

17. Охарактеризуйте суть впливу переквантування сигналу на ЦВДЗ.

18. У чому полягає суть цифрового формату аудіосигналів WMA?

19. Охарактеризуйте емпіричну функціу щільності ймовірності аудіосигналу без ЦВДЗ.

20. У чому полягає суть впливу переквантування сигналу на ЦВДЗ?

21. Охарактеризуйте девіацію функції щільності розподілу перекванто-ваного сигналу.

Розділ 4. Приховування даних у текстових файлах

4.1. Методи текстової стеганографії

Стеганографія, що використовує текстові контейнери, називається текстовою (text steganography). Далі буде розглянуто, яким чином можна застосовувати текстові контейнери для зберігання стега. Досить повна класифікація подібних методів дана в роботах [5; 7; 20]. З автоматичних методів текстової стеганографії в цій роботі згадується тільки один – *форматування*, тобто вирівнювання, *тексту за допомогою пробілів*.

Суть даного методу [20] полягає в розсуненні рядка шляхом збільшення пробілів між словами, коли один пробіл відповідає, наприклад, біту 0, два пробіли – біту 1. Однак пряме його застосування хоча й

можливе, але на практиці породжує масу незручностей, зокрема, оформлення тексту стає неохайним, що дозволяє легко запідозрити в ньому наявність стега.

У додатку Д наведена програма, де подібні проблеми вже вирішені. Програма перерозподіляє пробіли в межах поточної довжини рядка, переносячи по можливості довгі пробіли в її кінець. У результаті рядки вихідного тексту мають акуратний вигляд, що утрудняє виявлення стега.

Одиночні пробіли й пробіли перед останнім словом рядка не несуть інформаційного навантаження. В інших випадках парне число пробілів кодує 0, непарне – 1. Стег при записі попередньо шифрується, а при читанні розшифровується з використанням операції виключно "АБО" і убудованого в систему програмування датчика випадкових чисел, керованого константами Key1 і Key2.

Програма працює у двох режимах, обумовлених кількістю параметрів виклику. Першим параметром завжди вказується текстовий файл контейнера. Якщо таких параметрів два, програма витягає стег з контейнера й поміщає його у файл, зазначений другим параметром. Паралельно стег роздруковується на екрані. При трьох параметрах відбувається створення стега. Джерелом стегаповідомлення є файл, зазначений другим параметром, а результат роботи програми міститься у файл, заданий третім.

Необхідність обліку множини нюансів робить дану програму досить складною. Тому становлять інтерес способи вбудовування стега, віднесені в роботі [39] до категорії "багато інших". Із всіх таких способів були обрані найпростіші. Якщо перераховувати їх з підвищенням рівня складності, то це буде метод зміни порядку проходження маркерів кінця рядка, метод хвостових пробілів, метод знаків однакового накреслення і метод двійкових нулів (додаток Д). Тепер коротко розглянемо їх.

Метод зміни порядку проходження маркерів кінця рядка CR/LF використовує індиферентність гнітючого числа засобів відображення текстової інформації до порядку проходження символів перекладу рядка (CR) і повернення каретки (LF), що обмежують рядок тексту. Традиційний порядок проходження CR/LF відповідає 0, а інвертований LF/CR означає 1.

Метод хвостових пробілів припускає дописування наприкінці коротких рядків (менше 225 символів; значення 225 обране досить довільно) від 0 до 15 пробілів, що кодують значення напівбайта.

Метод знаків однакового накреслення припускає підміну (бітів 1) або відмову від такої підміни (бітів 0) російського символу латинським того ж накреслення. Ідея цього методу запозичена із роботи [20].

Метод двійкових нулів є різновидом методу знаків однакового накреслення й припускає або заміну першого в групі із двох або більше внутрішніх пробілів двійковим нулем (бітів 1), або відмову від неї (бітів 0).

Інтерфейс програм розглянутих методів однаковий. Усього можливі три режими їх роботи, обумовлені числом параметрів виклику, перший з яких завжди представляє файл стегаконтейнера. Якщо цей параметр єдиний, то кожна із програм реалізує функції стегадетектора: виробляється сканування контейнера з виведенням результату на екран. Якщо таких параметрів два, то стеганограма витягає з контейнера й розшифровується. Результат виводиться у файл, зазначений другим параметром, а також на екран. При трьох параметрах стег вибирається з файла, зазначеного другим параметром, шифрується виключно "АБО" і міститься у файл контейнера, заданого першим. Модифікований стеганограмний контейнер записується у файла, заданий третім параметром. Для шифрування у всіх випадках використовується програмний датчик випадкових чисел, початковий стан якого визначається константами Key1 і Key2. Завжди виробляється ехопечатка стега. Програма лістингу є допоміжною й призначена для твердого форматування таблиць, необхідного для роботи із програмою лістингу 1. Суть твердого форматування полягає в заміні всіх внутрішніх пробілів рядка двійковими нулями. При роботі із цією програмою в рядку виклику необхідно вказати імена вхідного й вихідного файлів. Для наочності всі програми написані вхідною мовою компактного компілятора Turbo Pascal 3.x компанії Borland, майже ідеально, якби не відсутність убудованого асемблера, прийнятного для дослідницької роботи.

4.2. Аналіз реалізації методів

Ефективність описаних методів упакування стега в контейнері була досліджена на переведеному в ASCII-вигляд тексті глави VI тому I книги "Мертва вода" обсягом 126 729 байтів, що нараховує 2 143 рядка з рядками, вирівняними на 65-символьну границю при абзацному відступі в чотири символи. Отримана щільність упакування (у порядку зростання) наведена в табл. 4.1.

Таблиця 4.1

Порівняння методів текстової стеганографії

Метод	Знаків стега	Щільність, %
Чергування маркерів кінця	267	0,21
Вирівнювання пробілами	411	0,32
Двійкові нулі	740	0,58
Хвостові пробіли	1071	0,85
Знаки однакового накреслення	4065	3,21

Звертає на себе увагу незвичайно висока ефективність упакування стега з використанням підміни символів. Отримані дані є лише оцінними й залежать не тільки від властивостей контейнера, але й від властивостей, що поміщені в його стег, хоча й у меншому ступені. Кількість автоматичних методів текстової стеганографії, природно, не обмежується розглянутими прикладами. Поповнити запас прикладів можна, зокрема, розумною комбінацією вже наведених.

Варто згадати про одне несподіване спостереження, що свідчить про те, що текстових файлів, придатних для використання в якості стегаконтейнера, набагато більше, ніж це може здатися з першого погляду. Дійсно, такими є й файли баз даних, символьні поля записів яких фактично становлять рядки фіксованої довжини (природно, без завершальних символів CR/LF).

Контрольні запитання

1. Назвіть методи текстової стеганографії.
2. Охарактеризуйте метод зміни порядку проходження маркерів кінця рядка CR/LF.
3. У чому полягає метод форматування тексту за допомогою пробілів?
4. Охарактеризуйте метод хвостових пробілів.
5. Охарактеризуйте метод двійкових нулів.
6. Охарактеризуйте метод чергування маркерів кінця.
7. Охарактеризуйте метод знаків однакового накреслення.
8. Назвіть основні порівняння методів текстової стеганографії.

Розділ 5. Атаки на стеганосистеми та протидія їм

5.1. Атаки проти систем прихованої передачі повідомлень. Атаки на системи цифрових водяних знаків (ЦВЗ)

Повернемося до розглянутої в першому розділі стеганосистемі, призначеної для прихованої передачі повідомлень. Досліджуємо докладніше можливості зловмисника Віллі за протидією Алісі й Бобу. Як відзначалося, зловмисник може бути пасивним, активним і злочинним. Залежно від цього він може створювати різні загрози.

Пасивний зловмисник може лише виявити факт наявності стега-каналу й, можливо, читати повідомлення. Чи зможе він прочитати повідомлення після його виявлення залежить від стійкості системи шифрування, і це питання, як правило, не розглядається в стеганографії. Якщо у Віллі є можливість виявити факт наявності прихованого каналу передачі повідомлень, то стеганосистема звичайно вважається нестійкою. Хоча існують і інші точки зору на стійкість стеганосистем . Здійснення виявлення стегаканалу є найбільш трудомістким завданням, а захист від виявлення вважається основним завданням стеганографії, за визначенням. Діапазон дій активного зловмисника значно ширше. Сховане повідомлення може бути їм вилучено або зруйновано. У цьому випадку Боб і, можливо, Аліса довідаються про факт втручання. У більшості випадків це суперечить інтересам Віллі (наприклад, за юридичними мотивами). Інша справа – видалення або руйнування цифрового водяного знака, які можуть розглядатися як основні загрози в цій множині.

Дії злочинного зловмисника найнебезпечніші. Він здатний не тільки руйнувати, але й створювати помилкові стеги. Історія протистояння розвідки й контррозвідки знає чимало прикладів, коли реалізація цієї загрози приводила до катастрофічних наслідків. Ця загроза актуальна й стосовно систем ЦВЗ. Маючи здатність створювати водяні знаки, зловмисник може створювати копії контента, що захищається, створювати помилкові оригінали й т. д. У багатьох випадках зловмисник може створювати помилкові стеги без знання ключа.

Для здійснення тієї або іншої загрози зловмисник застосовує атаки.

Найбільш проста атака – суб’єктивна. Віллі уважно розглядає зображення (слухає аудіозапис), намагаючись визначити "на око", чи є в ньому приховане повідомлення. Ясно, що подібна атака може бути

проведена лише проти зовсім незахищених стеганосистем . Проте вона, напевно, найпоширеніша на практиці, принаймні, на початковому етапі розкриття стеганосистеми. Первинний аналіз також може містити в собі такі заходи [5; 11]:

Первинне сортування стега за зовнішніми ознаками.

Виділення стега з відомим алгоритмом вбудовування.

Визначення використаних стегаалгоритмів.

Перевірка достатності обсягу матеріалу для стеганоаналізу.

Перевірка можливості проведення аналізу по окремих випадках.

Аналітична розробка стегаматеріалів. Розробка методів розкриття стеганосистеми.

Виділення стега з відомими алгоритмами вбудовування, але невідомими ключами та ін.

Із криптоаналізу нам відомі такі різновиди атак на шифровані повідомлення [5; 7; 11]:

атака з використанням тільки шифртексту;

атака з використанням відкритого тексту;

атака з використанням обраного відкритого тексту;

адаптивна атака з використанням відкритого тексту;

атака з використанням обраного шифртексту.

За аналогією із криптоаналізом у стегааналізі можна виділити такі типи атак [23; 27]:

Атака на основі відомого заповненого контейнера. У цьому випадку у зловмисника є один або трохи стегів. В останньому випадку передбачається, що вбудовування прихованої інформації здійснювалося Алісою тим самим способом. Завдання Віллі може полягати у виявленні факту наявності стегаканалу (основною), а також у його витягненні або визначенні ключа. Знаючи ключ, зловмисник одержить можливість аналізу інших стегаповідомлень.

Атака на основі відомого збудованого повідомлення. Цей тип атаки більшою мірою характерний для систем захисту інтелектуальної власності, коли як водяний знак використовується відомий логотип фірми. Завданням аналізу є одержання ключа. Якщо відповідному схованому повідомленню заповнений контейнер невідомий, то завдання вкрай важко розв’язуване.

Атака на основі обраного прихованого повідомлення. У цьому випадку Віллі має можливість пропонувати Алісі проаналізувати повідомлення яке пропонується для передачі у стеганосистемі.

Адаптивна атака на основі обраного прихованого повідомлення. Ця атака є частковим випадком попередньої атаки. У цьому випадку Віллі має можливість вибирати повідомлення для нав'язування Алісі адаптивно, залежно від результатів аналізу попередніх стегів.

Атака на основі обраного заповненого контейнера. Цей тип атаки більш характерний для систем ЦВЗ. Стегааналітик має детектор стега у вигляді "чорного ящика" і трохи стегів. Аналізуючи детектировані приховані повідомлення, зловмисник намагається розкрити ключ.

У Віллі може бути можливість застосувати ще три атаки, що не мають прямих аналогій у криптоаналізі.

*Атака на основі відомого порожнього контейнера. Я*кщо він відомий Віллі, то шляхом порівняння його з передбачуваним стегом він завжди може встановити факт наявності стегаканалу. Незважаючи на тривіальність цього випадку, у ряді робіт приводиться його інформаційно-теоретичне обґрунтування. Набагато цікавіше сценарій, коли контейнер відомий приблизно, з деякою погрішністю (як це може мати місце при додаванні до нього шуму). У розділі 4 показано, що в цьому випадку є можливість побудови стійкі стеганосистеми .

Атака на основі обраного порожнього контейнера. У цьому випадку Віллі здатний змусити Алісу користуватися запропонованим їй контейнером. Наприклад, запропонований контейнер може мати більші однорідні множини (однотонні зображення), і тоді буде важко забезпечити таємність впровадження.

Атака на основі відомої математичної моделі контейнера або його частини. При цьому атакуючий намагається визначити відмінність підозрілого повідомлення від відомої йому моделі. Наприклад, допустимо, що біти усередині відліку зображення корельовані. Тоді відсутність такої кореляції може служити сигналом про наявне приховане повідомлення. Завдання повідомлення полягає у тому, щоб не порушити статистики контейнера. Впроваджуючий й атакуючий можуть мати у своєму розпорядженні різні моделі сигналів, тоді в протиборстві, що інформаційно приховує, переможе той, що має кращу модель.

Розглянуті вище атаки мають одну особливість: вони не змінюють стегаповідомлення, що посилаються Алісою, а також не спрямовані на протидію роботи декодера Боба. У цьому полягає їхня позитивна сторона: дії Віллі навряд чи здатні насторожити Алісу й Боба. Порівняння працездатності стеганосистем виробляється звичайно стосовно деяких

стандартних тестів. У якості одного з них є атака, заснована на застосуванні алгоритму стиску JPEG (досить неефективна атака). Набагато більше подання про достоїнства того або іншого стегаалгоритму можна одержати, комплексно використовуючи різні атаки. Загальнодоступна в Internet програма Stirmark дозволяє більш повно аналізувати працездатність стегаалгоритмів. За твердженням творців програми на сьогоднішній день не існує загальновідомого стегаалгоритму, стійкого до їх комплексних атак.

Тому розроблювачами надається велике значення забезпеченню завадостійкості впровадження ЦВЗ. Це досягається, як правило, розширенням спектра прихованого повідомлення або застосуванням завадостійких кодів. Системи з розширенням спектра широко застосовуються у зв'язку для завадостійкої передачі сигналів. Але чи є вони досить завадостійкими для застосування у ЦВЗ? Виявляється, далеко не завжди. Розглянемо пропоновані дослідниками методи атак і протидії їм.

5.2. Класифікація атак на стеганосистеми цифрових відеознаків (ЦВДЗ)

Як відзначалося в першому розділі, ЦВЗ повинні задовольняти суперечливим вимогам візуальної (аудіо) непомітності й працездатності до основних операцій обробки сигналів. Надалі без втрати спільності будемо припускати, що як контейнер використовується зображення.

Звернемося знову до системи вбудовування повідомлень шляхом модифікації молодшого значущого біта (LSB) пікселів, розглянутої в першому розділі. Практично будь-який спосіб обробки зображень може привести до руйнування значної частини убудованого повідомлення. Наприклад, розглянемо операцію обчислення ковзного середнього по двох сусіднім пікселам $(a+b)/2$, що є найпростішим прикладом низькочастотної фільтрації. Нехай значення пікселів a і b можуть бути парними або непарними з імовірністю $p=1/2$. Тоді й значення молодшого значущого біта зміниться після усереднення в половині випадків. До того ж ефекту може привести й зміна шкали квантування, скажемо, з 8 до 7 бітів. Аналогічний вплив робить і стиск зображень із втратами. Більше того, застосування методів очищення сигналів від шумів, що використовують оцінювання й вирахування шуму, приведе до перекручування переважної більшості бітів прихованого повідомлення.

Існують також і набагато більш згубні для ЦВДЗ операції обробки зображень, наприклад, масштабування, повороти, усікання, перестановка пікселів. Ситуація збільшується ще й тим, що перетворення стегаповідомлення можуть здійснюватися не тільки зловмисником, але й законним користувачем, або бути наслідком помилок при передачі по каналу зв'язку.

Зрушення на трохи пікселів може привести до невиявлення ЦВДЗ у детекторі. Розглянемо це на прикладі наведеного в першому розділі стегаалгоритму. У детекторі маємо

$$S_{WS} \cdot W = (S_{0S} + W_S) \cdot W = S_{0S} \cdot W + W_S \cdot W,$$

де індексом S позначені зміщені версії відповідних сигналів.

Добуток $S_{0S} \cdot W$, як і колись, близький до нуля. Однак, якщо знаки $\pm$ у W вибиралися випадково й незалежно, то й $W_S \cdot W$ буде близьким до нуля, і стегаповідомлення не буде виявлено. Аналогові відеомагнітофони, як правило, трохи зрушують зображення через нерівномірність обертання двигуна стрічкопротягувального механізму або зношування стрічки. Зрушення може бути непомітний для ока, але привести до руйнування ЦВДЗ.

Можлива різна класифікація атак на стеганосистеми. Розглянемо атаки, специфічні для систем ЦВДЗ. Можна виділити наступні категорії атак проти таких стеганосистем [28; 29; 44;56]:

1. *Атаки проти збудованого повідомлення* – спрямовані на видалення або псування ЦВДЗ шляхом маніпулювання стега. Вхідні в цю категорію методи атак не намагаються оцінити й виділити водяний знак. Прикладами таких атак можуть бути лінійна фільтрація, стиск зображень, додавання шуму, вирівнювання гістограми, зміна контрастності й т. д.

2. *Атаки проти стегадетектора* – спрямовані на те, щоб утруднити або унеможливити правильну роботу детектора. При цьому водяний знак у зображенні залишається, але губиться можливість його прийому. У цю категорію входять такі атаки, як афінні перетворення (тобто масштабування, зсуви, повороти), усікання зображення, перестановка пікселів і т. д.

3. *Атаки проти протоколу використання ЦВДЗ* – в основному пов'язані зі створенням помилкових ЦВДЗ, помилкових стегів, інверсією ЦВДЗ, додаванням декількох ЦВДЗ.

4. *Атаки проти самого ЦВДЗ* – спрямовані на оцінювання й витягнення ЦВДЗ зі стегаповідомлення, по можливості без перекручування контейнера. У цю групу входять такі атаки, як атаки змови, статистичного усереднення, методи очищення сигналів від шумів, деякі види нелінійної фільтрації [56] та ін.

Треба відмітити, що розглянута класифікація атак не є єдино можливою й повною. Крім того, деякі атаки (наприклад, видалення шуму) можуть бути віднесені до декількох категорій. У роботі [44] була запропонована інша класифікація атак, що також має свої достоїнства й недоліки.

Відповідно до цієї класифікації всі атаки на системи вбудовування ЦВДЗ можуть бути розділені на чотири групи:

1) атаки, спрямовані на видалення ЦВДЗ;
2) геометричні атаки, спрямовані на перекручування контейнера;
3) криптографічні атаки;
4) атаки проти використовуваного протоколу вбудовування й перевірки ЦВДЗ.

5.3. Атаки, спрямовані на видалення ЦВДЗ

До цієї групи відносяться такі атаки, як очищення сигналів-контейнерів від шумів, перемодуляція, стиснення з втратами (квантування), усереднення і колізії. Ці атаки засновані на припущенні про те, що ЦВДЗ є статистично описуваним шумом. Очищення від шуму полягає в фільтрації сигналу з використанням критеріїв максимальної правдоподібності або максимуму апостеріорної імовірності. В якості фільтра, що реалізує критерій максимальної правдоподібності, може використовуватися медіа (для ЦВДЗ, що має розподіл Лапласа) або усереднюючий (для гаусівського розподілу) фільтр, який застосовують у програмному пакеті StirMark. За критерієм максимуму апостеріорної ймовірності найкращим буде адаптивний фільтр Вінера (у випадку, якщо в якості моделі контейнера використовується нестаціонарний гаусівський процес), а також порогові методи очищення від шуму (м’які і жорсткі пороги) (модель – узагальнений гаусівський процес), які мають багато спільного з методами стиснення з втратами.

Стиснення з втратами та очищення сигналів від шумів значно зменшують пропускну здатність стегаканалу, особливо для гладких

галузей зображення, коефіцієнти перетворення яких можуть бути "обнулені" без помітного зниження якості відновленого зображення.

Перемодуляція – порівняно новий метод, який є специфічним саме для атак на ЦВДЗ. Атака перемодуляції була вперше запропонована в роботі [56]. У даний час відомі її різні варіанти, залежно від використовуваного в стеганосистемі декодера. У побудові атаки є свої нюанси для стеганосистеми *M*-ї модуляції, стеганосистеми, що використовують коди завадостійкості, які використовують кореляційний декодер. У будь-якому випадку вважається, що ЦВДЗ впроваджений у зображення з застосуванням широкосмугових сигналів і розмножений на всі зображення. Оскільки оцінюваний декодером ЦВДЗ корельований з істинним, з'являється можливість обману декодера. Атака будується таким чином. Спочатку ЦВДЗ "пророкує" шляхом вирахування фільтрованої версії зображення з захищеного зображення (застосовується медіанний фільтр). "Пророкований" ЦВДЗ піддається ВЧ-фільтрації, урізається, множиться на два і віднімається з вихідного зображення. Крім того, якщо відомо, що при впровадженні ЦВДЗ множився на деяку маску для підвищення непомітності вбудованості, то атакуючий оцінює цю маску і домножує на неї ЦВДЗ. В якості додаткової засоби в "обмані" декодера вважається ефективним вбудовування в високочастотні множини зображення (де спотворення непомітні) шаблонів, які мають не гаусівський розподіл. Таким чином буде порушено оптимальність лінійного кореляційного детектора.

Така атака буде ефективною лише проти високочастотного ЦВДЗ, тому реальні ЦВДЗ будуються так, щоб їх спектр відповідав спектру вихідного зображення. Справа в тому, що достовірна оцінка виходить лише для високочастотних компонент ЦВДЗ. Після її вирахування низькочастотна компонента ЦВДЗ залишається незмінною і дає в детекторі позитивний кореляційний відгук. Високочастотна ж складова дасть негативний відгук, що в сумі дасть нуль, і ЦВДЗ не буде виявлено. В якості другої протидії цій атаці було запропоновано виконання попередньої низькочастотної фільтрації.

У роботі [37] наведена модифікація цього алгоритму, що полягає в застосуванні фільтра Вінера замість медіанного і більш інтелектуального способу знаходження коефіцієнта множення. Він вибирається так, щоб мінімізувати коефіцієнт взаємної кореляції між ЦВДЗ і стегом. Крім того, додається ще один крок: накладення випадкового шуму. Ця атака не

працює проти адаптивно вбудованого ЦВДЗ, оскільки у ній передбачається, що ЦВДЗ і стег є стаціонарним гаусівським процесом з нульовим середнім. Ясно, що це припущення не виконується також і для реальних зображень. Тому С. Волошиновським та іншими запропонована атака, в якій сигнали моделюються як не стаціонарний гаусівський або узагальнений стаціонарний гаусівський процес [38]. Коефіцієнт множення ЦВДЗ вибирається виходячи з локальних властивостей зображення. Замість накладення випадкового шуму запропоновано додавати відліки зі знаком, протилежним знаку відліку ЦВДЗ (в припущенні, що ЦВДЗ є послідовністю біполярних символів). Це ще більше ускладнює роботу кореляційного детектора. Звичайно, знаки потрібно змінювати не у всіх, а тільки у частині відліків оцінюваного ЦВДЗ, наприклад, випадково.

До інших атак цієї групи відносяться атаки усереднення і змови. У разі наявності великої кількості копій стега з різними ЦВДЗ або з різними ключами впровадження можна виконати їх усереднення. Наприклад, кадри відеосигналу можуть мати різні ЦВДЗ. Якщо ЦВДЗ мав нульове середнє, то після усереднення він буде відсутній у зображенні.

Атака шляхом статистичного усереднення наведена в [56]. Зловмисник може спробувати оцінити ЦВДЗ і вирахувати його з зображення. Такий вид атак особливо небезпечний у випадку, коли атакуючий може отримати певний узагальнений ЦВДЗ, наприклад, деякий $W = f(S_0, W)$, не залежних сильно від вихідного зображення S_0.

Атакуючий може виявити ЦВДЗ шляхом усереднення декількох зображень. Наприклад, у нього є $S_0 + W$, $S_1 + W$, ..., $S_N + W$. Тоді їх сума $NW + \sum_i S_i$ буде досить близька до NW, якщо N велике, а зображення статистично незалежні.

Протиотрутою подібної атаки може бути випадкове використання одного із двох ЦВДЗ з ймовірностями p_1 й $p_2 = 1 - p_1$. Тоді вищенаведена атака дасть лише $p_1 W_1 + (1 - p_1) W_2$. Однак атака може бути поліпшена в тому випадку, якщо в атакуючого є якісь припущення про те, який ЦВДЗ із двох убудований у дане зображення. Тоді всі зображення можуть бути розподілені на два класи: 1 й 2. Нехай $P_л$ – ймовірність того, що зображення віднесене до невірного класу. Тоді усереднення по великій кількості N_1 зображень класу 1 дає $x_1 = N_1 p_1 (1 - P_л) W_1 + N_1 (1 - p_1) P_л W_2$. Аналогічне усереднення по N_2 зображень класу 2 дає $x_2 = N_2 p_1 P_л W_1 + N_2 (1 - p_1)(1 - P_л) W_2$.

Обчислення зваженої різниці дає $\frac{x_1}{N_1} - \frac{x_1}{N_2} = p_1(1-2P_л)W_1 - (1-p_1)(1-2P_л)W_2$.

Отже, для будь-якого $P_л \neq 1/2$, що атакує, може оцінити суму й різницю p_1W_1 й $(1-p_1)W_2$, звідки він може одержати W_1 й W_2.

При *атаці змови* є кілька однакових копій, що містять різні ЦВДЗ, а для атаки з кожної копії вибираються якісь частини, які в сукупності й утворять атаковану множину. Атаки на основі "змови" описані, наприклад, у роботах [28; 29]. Чим більше утримуючих стеги копій є в зловмисника, тим вище ймовірність того, що близьке до вихідного реконструйоване зображення не буде містити стеги. У стегасистемах із закритим ключем така атака не настільки ефективна в силу того, що атакуючий не може перевірити, чи містять апроксимації ЦВДЗ, що в нього виходять. Це підвищує безпеку стеганосистем із закритим ключем. Захищеність від цієї атаки можна також підвищити за рахунок спеціальної побудови стега.

Ще одна ефективна атака на ЦВДЗ називається *мозаїчною* [29]. Ця атака спрямована на пошукові системи, що відслідковують незаконно розповсюджувані зображення. Зображення розбивається на кілька частин, так що пошукова система ЦВДЗ не виявляє. Internet-браузер демонструє фактично кілька шматочків зображення, близько розташованих один від одного, так що в цілому зображення виглядає неспотвореним. Для протидії такій атаці ЦВДЗ повинен виявлятися навіть у малих частинах зображення. Це дуже складна вимога, навіть більш важка, ніж робастність до обрізання країв зображення, тому що в останньому випадку атакуючий обмежений необхідністю збереження якості зображення. Напевно, більш здійсненним було б створення інтелектуальних пошукових систем, здатних "зібрати" зображення зі шматочків і перевірити наявність у ньому ЦВДЗ.

Цікава і практично значуща атака запропонована в роботі [57]. Вона заснована на оцінюванні ЦВДЗ, але не в множині вихідного зображення, а у його гістограмі. Атака особливо ефективна проти систем неадаптивних систем ЦВДЗ, але може бути використана й для оцінювання адаптивно впровадженого ЦВДЗ.

Пояснити атаку можна на такому прикладі. Нехай ЦВДЗ $w \in \{-1,1\}$, а у вихідному зображенні є ізольоване значення піксела. Наприклад, значення 200 зустрічається 300 разів, а значення 199 та 201 – жодного

разу. Тоді після впровадження ЦВДЗ значення 199 і 201 зустрінуться приблизно 150 разів, а значення 200 – жодного разу. Це і є демаскуюча ознака. Як показано на прикладі в роботі [57], цей метод може бути застосований і у випадку наявності на гістограмі зображення декількох ненульових значень, розділених трьома і більше нулями.

Для успішного використання гістограмної атаки запропоновано виконувати попереднє згладжування зображення-контейнера. Тоді зменшується діапазон значень кольорів й з'являється багато нульових ланцюжків. Утім, ефективність атаки підвищується в результаті згладжування не для всіх зображень.

У роботі [57] показано також, як гістограмна атака підсилюється при наявності декількох зображень, тобто у випадку її комбінування з атакою змови.

5.4. Геометричні атаки

На відміну від атак видалення, геометричні атаки прагнуть не видалити ЦВДЗ, а змінити його шляхом внесення просторових або тимчасових перекручувань. Геометричні атаки математично моделюються як афінні перетворення з невідомим декодеру параметром. Усього є шість афінних перетворень: масштабування, зміна пропорцій, повороти, зсуви і усікання. Ці атаки приводять до втрати синхронізації в детекторі ЦВДЗ і можуть бути локальними або глобальними, тобто застосованими до всього сигналу. При цьому можливо вирізання окремих пікселів або рядків, перестановка їх місцями, застосування різних перетворень і т. д. Подібні атаки реалізовані в програмах Unsign (локальні атаки) і Stirmark (локальні та глобальні атаки).

Існують і більш "інтелектуальні" атаки на застосовуваний метод синхронізації ЦВДЗ. Основна ідея цих атак полягає в розпізнаванні методу синхронізації і руйнуванні його шляхом згладжування піків в амплітудному спектрі ЦВДЗ. Атаки ефективні в припущенні про те, що у якості механізмів синхронізації використовуються періодичні шаблони. При цьому для забезпечення синхронізації можуть використовуватися два підходи: вбудовування піків у спектральній множині, або періодичне впровадження послідовності ЦВДЗ. В обох випадках у спектрі утворяться піки, які руйнуються в розглянутій атаці. Після руйнування можна застосовувати інші геометричні атаки: синхронізації вже немає.

Сучасні методи вбудовування ЦВДЗ робастні до глобальних атак. У них застосовуються спеціальні методи відновлення синхронізації, що мають багато загального із застосовуваними в техніці зв'язку. Робастність досягається за рахунок використання інваріантних до зсуву галузей [40], застосування опорного ЦВДЗ [6], обчислення авто-кореляційної функції ЦВДЗ.

Якщо забезпечення робастності до глобальних геометричних атак є більш-менш вирішеним завданням, то забезпечення стійкості до локальних змін зображення є відкритим питанням. Ці атаки засновані на тому, що людське око мало чутливе до невеликих локальних змін картинки.

5.5. Криптографічні атаки

Криптографічні атаки названі так тому, що вони мають аналоги у криптографії. До них відносяться атаки з використанням оракула, а також злом за допомогою "грубої сили".

Атака з використанням оракула дозволяє створити незахищене ЦВДЗ зображення при наявності в зловмисника детектора. У роботі [49] досліджується стійкість ЦВДЗ на основі розширення спектра до атаки при наявності детектора у вигляді "чорного ящика". Метод полягає в експериментальному вивченні поводження детектора для з'ясування того, на які зображення він реагує, а на які – ні. Наприклад, якщо детектор виносить "м'які" рішення, тобто показує ймовірність наявності стега в сигналі, то атакуючий може з'ясувати, які невеликі зміни в зображенні впливають на поводження детектора. Модифікуючи зображення піксел за пікселем, він може взагалі з'ясувати, який алгоритм використовує детектор. У випадку детектора з "жорстким" рішенням атака здійснюється біля границі, де детектор міняє своє рішення з "є присутнім" на "відсутній".

Приклад атаки на детектор з жорстким рішенням:

1. На основі наявного зображення, що містить стегано-повідомлення, створюється тестове зображення. Тестове зображення може бути створено різними шляхами, модифікуючи вихідне зображення доти, доки детектор не покаже відсутності ЦВДЗ. Наприклад, можна

поступово зменшувати контрастність зображення, або піксел за пікселем заміняти дійсні значення якимись іншими.

2. Атакуючий збільшує або зменшує значення якого-небудь піксела доти, доки детектор не виявить ЦВДЗ знову. У такий спосіб з'ясовується, збільшив або зменшив значення даного піксела ЦВДЗ.

3. Крок 2 повторюється для кожного піксела в зображенні.

4. Знаючи, наскільки чутливий детектор до модифікації кожного піксела, атакуючий визначає пікселі, модифікація яких не приведе до істотного погіршення зображення, але порушить роботу детектора.

5. Дані пікселі віднімаються з вихідного зображення.

Чи можлива побудова стеганоалгоритму, стійкого до подібної атаки, поки невідомо.

Відомо різновид вищенаведеної атаки для ймовірнісного детектора. Також, як і раніше, атака починається з побудови тестового зображення на границі ухвалення рішення детектором. Потім вибирається випадкова двійкова послідовність, і її елементи додаються до пікселей тестового зображення. Якщо детектор виносить рішення про наявність, то ця послідовність вважається ЦВДЗ. В іншому випадку – ЦВДЗ вважається протилежним цій послідовності. Далі виконується випадкова перестановка елементів у послідовності, і процес повторюється. Повторивши цю процедуру кілька разів і додавши всі проміжні результати, одержимо досить гарну оцінку ЦВДЗ. Можна показати, що точність оцінювання $O\left(\sqrt{J/N}\right)$, де J – кількість спроб, N – кількість пікселів у вихідному зображенні. Звідси виходить, що при фіксованій точності оцінювання кількість спроб лінійно залежить від кількості пікселів у зображенні. Також може бути показано, що кількість спроб пропорційно квадрату ширини зони ухвалення рішення. Таким чином, розроблювач ймовірнісного детектора повинен компромісно вибрати між такими параметрами: великою величиною зони ухвалення рішення, тобто безпекою, малим значенням верхнього порога зони, тобто малою ймовірністю помилкового виявлення стега, і більшим значенням нижнього порога зони, тобто малою ймовірністю помилкового невиявлення стега. У цілому з роботи [2] та інших слідує, що система ЦВДЗ на основі розширення спектра не повинна мати загальнодоступного детектора.

5.6. Атаки проти протоколу, що використовується

У роботах [8; 10; 26] показано, що багато стеганосистем ЦВДЗ чутливі до так званої інверсної атаки. Ця атака полягає в наступному. Зловмисник заявляє, що в захищеному зображенні частина даних є його водяним знаком. Після цього він створює помилковий оригінал, віднімаючи цю частину даних. У помилковому оригіналі присутній дійсний ЦВДЗ. З іншого боку, у захищеному зображенні присутній проголошений зловмисником помилковий ЦВДЗ. Настає нерозв'язна ситуація. Звичайно, якщо в детектора є вихідне зображення, то власник може бути виявлений. Але, як показано у роботі [8], далеко не завжди. У роботах [8; 10; 26] наведені методи захисту від подібної атаки. У них показано, що стійкий до подібної атаки ЦВДЗ повинен бути незворотним. Для цього він робиться залежним від зображення за допомогою односпрямованої функції.

Нехай V – вихідне зображення, W – водяний знак законного власника. Тоді захищене зображення $V_W = V + W$. Зловмисник повідомляє довільну послідовність бітів W_F своїм водяним знаком і віднімає її із захищеного зображення, у результаті чого одержує помилковий оригінал $V_F = V_W - W_F$. Тепер якщо виконується рівність $V_F + W_F = V_W$, то ціль зловмисника досягнута. ЦВДЗ називається в цьому випадку зворотним. Неможливо визначити, що є оригіналом: V або V_F тобто, хто є власником контента.

У роботі [8] дано два визначення незворотності: ослаблене і сильне. При цьому використовуються такі позначення:

$E(V,W) = V_W$ – процедура вбудовування ЦВДЗ;

$D(V,V_W) = W'$ (або $D(V_W) = W'$) – процедура витягнення ЦВДЗ;

α – маштабуючий коефіцієнт;

C – бінарна ознака подоби двох сигналів: дорівнює 1, якщо коефіцієнт взаємної кореляції більше деякого порога δ; в іншому випадку – дорівнює 0.

Перше визначення незворотності таке.

Стеганоалгоритм (E,D,C) є (строго) зворотним, якщо для кожного V_W існує відображення E^{-1} таке, що $E^{-1}(V_W) = (V_F, W_F)$ і $E(V_F, W_F) = V_W$. При цьому E^{-1} є розрахунково здійсненним, W_F належить до класу

припустимих ЦВДЗ, істинне та помилкове зображення візуально подібні і $C(D(V_W,V_F),W_F,\delta)=1$. Інакше (E,D,C) (слабо) незворотний.

У цьому визначенні вимога, щоб $E(V_F,W_F)=V_W$, накладає занадто сильне обмеження. Справді, навіть $E(V,W)=V_W$ може не виконуватися в силу різного роду перекручувань V_W. З іншого боку, ця вимога занадто слабка для визначення зворотності. Тому у роботі [8] воно замінено на вимогу, щоб $E(V_F,W_F)=V_{W'}$, де $C(V_{W'},V_W,\delta)=1$.

Друге визначення незворотності таке.

Стеганоалгоритм (E,D,C) є (слабко) зворотним, якщо для будь-якого V_W існує відображення E^{-1} таке, що $E^{-1}(V_W)=(V_F,W_F)$ і $E(V_F,W_F)=V_{W'}$. При цьому E^{-1} розрахунково здійснено, W_F належить до класу припустимих ЦВДЗ, $C(V_{W'},V_F,\delta)=1$, $C(V_{W'},V_W,\delta)=1$ і $C(D(V_{W'},V_F),W_F,\delta)=1$. Інакше (E,D,C) (строго) незворотний.

У роботі [6] описані атаки, що використовують наявність стеганокодера. Подібна атака є однією з найнебезпечніших. Одним з можливих сценаріїв, коли її небезпека існує, є наступний. Нехай користувачеві дозволено зробити одну копію з оригіналу, але не дозволено робити копії з копій. Записуючий пристрій повинен змінити ЦВДЗ з "дозволена копія" на "копіювання не дозволено". У цьому випадку атакуючий має доступ до повідомлення до і після вкладення ЦВДЗ. Виходить, він може обчислити різницю між вихідним і модифікованим повідомленням. Ця різниця дорівнює $f(S_0,W)$. Далі вихідне зображення попередньо деформується: з нього віднімається $f(S_0,W)$. Після здійснення копіювання буде записано $S_0-f(S_0,W)+f(S_0-f(S_0,W),W)$, що дуже близько до вихідного зображення S_0. Ця близькість обумовлена тим, що ЦВДЗ повинні бути робастні до додавання адитивного шуму. Отже, $f(S_0+\varepsilon,W)\approx f(S_0,W)$. У випадку даної атаки стеганоповідомлення виступає як шум та $f(S_0-f(S_0,W),W)\approx f(S_0,W)$.

У роботі [23] та інших досліджуються атаки на системи захисту від копіювання. У ряді випадків набагато простіше не видаляти ЦВДЗ, а перешкодити його використанню за призначенням. Наприклад, можливе впровадження додаткових ЦВДЗ так, що стає не зрозуміло, який з них ідентифікує справжнього власника контента.

Іншою відомою атакою на протокол використання ЦВДЗ є *атака копіювання*. Ця атака полягає в оцінюванні ЦВДЗ у захищеному

зображенні і впровадженні оціненого ЦВДЗ в інші зображення. Метою може бути, наприклад, протидія системі імітозахисту або автентифікації.

Одним з недоліків стеганосистеми, застосовуваної для захисту від копіювання, є те, що детектор здатний виявити ЦВДЗ тільки коли відеосигнал візуально прийнятний. Однак можна піддати сигнал скреблюванню, одержати шумоподібний сигнал, потім без перешкод незаконно скопіювати його. У відеоплеєр у цьому випадку вбудовується дескремблер, що і відновлює незаконно зроблену копію. Апаратна реалізація скремблера і дескремблера досить проста і іноді використовується для захисту, наприклад, програм кабельного телебачення. Можливим захистом проти такого підходу є дозвіл копіювання тільки визначеного формату даних.

5.7. Методи протидії атакам на системи ЦВДЗ. Статистичний стеганоаналіз та протидії

У найпростіших стеганосистемах ЦВДЗ при вбудовуванні використовується псевдовипадкова послідовність, що є реалізацією білого гаусівського шуму і невраховуючої властивості контейнера. Такі системи практично нестійкі до більшості розглянутих вище атак. Для підвищення робастності стеганосистем можна запропонувати ряд поліпшень.

У робастній стеганосистемі необхідний правильний вибір параметрів псевдовипадкової послідовності. Відомо, що при цьому системи з розширенням спектра можуть бути досить робастними стосовно атак типу додавання шуму, стиску й т. п. Вважається, що ЦВДЗ повинен виявлятися при досить сильній низькочастотній фільтрації (7×7 фільтр із прямокутною характеристикою). Отже, база сигналу повинна бути велика, що знижує пропускну здатність стеганоканалу. Крім того, використовувана як ключ ПВП повинна бути криптографічно безпечною.

Причина нестійкості систем ЦВДЗ з розширенням спектра до подібних атак пояснюється тим, що послідовність, яка звичайно використовується для вкладання, має нульове середнє.

Після усереднення великої кількості реалізацій ЦВДЗ видаляється. Є відомим спеціальний метод побудови водяного знака, націлений проти подібної атаки. При цьому коди розробляються таким чином, щоб при будь-якому усередненні завжди залишалася не дорівнююча нулю частина послідовності (статична компонента). Більш того, по ній можливе відновлення решти послідовності (динамічна компонента). Недоліком запропонованих кодів є те, що їх довжина збільшується експонентно зі

зростанням кількості розповсюджуваних захищених копій. Можливим виходом із цього становища є застосування ієрархічного кодування, тобто призначення кодів для групи користувачів. Деякі аналогії тут є з системами стільникового зв'язку з кодовим поділом користувачів (CDMA).

Різні методи протидії пропонувалися для *вирішення проблеми прав власності*. *Перший спосіб* полягає в побудові незворотного алгоритму ЦВДЗ. ЦВДЗ повинен бути адаптивним до сигналу і вбудовуватися за допомогою односпрямованої функції, наприклад, геш-функції. Геш-функція перетворює 1 000 бітів вихідного зображення V в бітову послідовність b_i, $i = \overline{1...1000}$ [35]. Далі, залежно від значення використовується дві функції убудовування ЦВДЗ. Якщо $b_i = 0$, то використовується функція $v_i(1+\alpha w_i)$, якщо $b_i = 1$, то функція $v_i(1-\alpha w_i)$, де v_i – i -й коефіцієнт зображення; w_i – i -й бітів вбудованого повідомлення. Передбачається, що такий алгоритм формування ЦВДЗ запобіжить фальсифікації. У роботі [26] на прикладі показано, що для того, щоб цей алгоритм був незворотним, всі елементи w_i повинні бути позитивними.

Другий спосіб вирішення проблеми прав власності полягає у вбудовуванні в ЦВДЗ деякої часової позначки, що надається третьою, довіреною стороною. У разі виникнення конфлікту особа, яка має на зображенні більш ранню часову позначку, вважається справжнім власником.

Один із принципів побудови робастної ЦВДЗ полягає в адаптації його спектра. У ряді робіт показано, що огинаюча спектра ідеального ЦВДЗ повинна повторювати огинаючу спектра контейнера. Спектральна щільність потужності ЦВДЗ, звичайно ж, набагато менше. При такій огинаючій спектра вінеровський фільтр дає найгіршу оцінку ЦВДЗ з можливих: дисперсія значень помилки досягає дисперсії значень заповненого контейнера. На практиці адаптація спектра ЦВДЗ можлива шляхом локального оцінювання спектра контейнера. З іншого боку, методи вбудовування ЦВДЗ в множини перетворення досягають цієї мети за рахунок адаптації в множини трансформанти.

Для захисту від *атак типу афінного перетворення* можна використовувати додатковий (опорний) ЦВДЗ. Цей ЦВДЗ не несе в собі інформації, але використовується для "реєстрації" виконуваних зловмисником перетворень. У детекторі ЦВДЗ є схема попереднього деформування, що виконує зворотне перетворення. Тут є аналогія з використовуваними у зв'язку тестовими послідовностями. Однак у цьому випадку атака може бути спрямована саме проти опорного ЦВДЗ. Іншою альтернативою є вкладення ЦВДЗ у візуально значущі множини зображення, які не можуть

бути вилучені з нього без суттєвої його деградації. Нарешті, можна розмістити стеги в інваріантних до перетворення коефіцієнтах. Наприклад, амплітуда перетворення Фур'є інваріантна до зсуву зображення (при цьому змінюється тільки фаза).

Іншим методом захисту від подібних атак є блочний детектор. Модифіковане зображення розбивається на блоки розміром 12×12 або 16×16 пікселів, і для кожного блоку аналізуються всі можливі спотворення. Тобто пікселі в блоці піддаються повороту, перестановці і т. д. Для кожної зміни визначається коефіцієнт кореляції ЦВДЗ. Перетворення, після якого коефіцієнт кореляції виявився найбільшим, вважається реально виконаним зловмисником. Таким чином з'являється можливість як би звернути внесені зловмисником спотворення. Можливість такого підходу ґрунтується на припущенні про те, що зловмисник не буде значно спотворювати контейнер (це не в його інтересах).

Основним завданням стеганоаналізу є визначення факту наявності прихованого повідомлення у ймовірному контейнері (мовлення, відео, зображення). Вирішити це завдання можливо шляхом вивчення статистичних властивостей сигналу. Наприклад, розподіл молодших бітів сигналів має, як правило, шумовий характер (помилки квантування). Вони несуть найменшу кількість інформації про сигнал і можуть використовуватися для впровадження прихованого повідомлення. При цьому, можливо, зміниться їх статистика, що й послужить для атак ознакою наявності прихованого каналу.

Для непомітного вбудовування даних стеганокодер повинен вирішити три завдання: виділити підмножину бітів, модифікація яких мало впливає на якість, вибрати із цієї підмножини потрібну кількість бітів відповідно до розміру прихованого повідомлення і виконати їх редагування. Якщо статистичні властивості контейнера не змінилися, то впровадження інформації можна вважати успішним. Оскільки розподіл незначущих бітів найчастіше близький до білого шуму, вбудовані дані повинні мати той самий характер. Це досягається за рахунок попереднього шифрування повідомлення або його стиснення.

Стеганоаналітік на основі вивчення сигналу завжди може виділити підмножину незначущих бітів, роблячи ті ж припущення, що і стеганограф. Далі він повинен перевірити відповідність їх статистики передбачуваній. При цьому, якщо аналітик має у своєму розпорядженні кращу модель даних, ніж стеганограф, вкладення буде виявлено. Тому по-справжньому гарні моделі сигналів різного характеру, ймовірно, тримаються в секреті,

і ви не зустрінете їх у відкритих публікаціях. Можна лише дати рекомендації загального характеру. При побудові моделі треба враховувати:

неоднорідність послідовностей відліків;

залежність між бітами у відліку (кореляцію);

залежність між відліками;

нерівномірність умовних розподілів у послідовності відліків;

статистику довжин серій (послідовностей з однакових бітів).

Відповідність статистики, що реально спостерігається очікуваній зазвичай перевіряється за допомогою критерію Хі-квадрат. Перевірка може здійснюватися на рівні монобітів, дібітів і т. д. Можливі й більш складні тести, аналогічні застосовуються при тестуванні криптографічних безпечних програмних датчиків випадкових чисел. Як показано в одній з робіт на прикладі звукових файлів, критерій Хі-квадрат дозволяє виявити модифікацію всього лише 10 % незначущих бітів. Там же показана ефективність для стеганоаналізу і ще більш простого критерію $\theta = \frac{m_{00} - m_{01}}{2} - \frac{m_{11} - m_{10}}{2}$, де – кількість переходів зі значення біта i в значення j. Застосування тесту довжин серій базується на такому факті: у випадковій послідовності серій великої довжини (>15) зустрічаються значно рідше, ніж у незначущих бітах реальних сигналів. Тому вбудовування випадкового сигналу може бути помічено після застосування цього тесту.

Таким чином, протидія статистичного стеганоаналізу повинна полягати в побудові математичних моделей сигналів-контейнерів, пошуці на їх основі "дозволених" для модифікації галузей та впровадження у них прихованої інформації, чия статистика невідрізняється від статистики контейнера.

5.8. Практична оцінка стійкості стеганосистем. Теоретико-складнісний підхід до оцінки стійкості стеганосистем. Імітостійкість систем передачі прихованих повідомлень

Порівняно з досить добре дослідженими криптографічними системами поняття та оцінка безпеки стеганографічних систем більш складні і допускають більшу кількість їх тлумачень [48; 30]. Зокрема, це пояснюється як недостатньою теоретичним та практичним опрацюванням питань безпеки стеганосистем, так і великою різноманітністю

завдань стеганографічного захисту інформації. Стеганосистеми водяних знаків, зокрема, повинні виконувати завдання захисту авторських і майнових прав на електронні повідомлення при різних спробах активного зловмисника перекручування або стирання вбудованої в них автентифікуючої інформації. Формально кажучи, системи ЦВДЗ повинні забезпечити автентифікацію відправників електронних повідомлень. Таке завдання може бути покладене на криптографічні системи електронного цифрового підпису (ЕЦП) даних, але, на відміну від стеганосистем водяних знаків, відомі системи ЕЦП не забезпечують захист авторства не тільки цифрових, але й аналогових повідомлень, і в умовах, коли активний зловмисник вносить спотворення в повідомлення, що захищаються, та автентифікацію інформації. Інші вимоги з безпеки висуваються до стеганосистем, призначених для приховування факту передачі конфіденційних повідомлень від пасивного зловмисника. Також має свої особливості забезпечення імітостійкості стеганосистем до введення в прихований канал передачі неправдивої інформації [15; 18].

Як і для криптографічних систем захисту інформації, безпека стеганосистем описується та оцінюється їх стійкістю (стеганографічною стійкістю, або стеганостійкістю). Під *стійкістю різних стеганосистем* розуміється їх здатність приховувати від кваліфікованого зловмисника факт прихованої передачі повідомлень, здатність протистояти спробам зловмисника зруйнувати, спотворити, видалити повідомлення, що таємно передаються, а також здатність підтвердити або спростувати справжність приховано переданої інформації.

Розглянемо визначення стеганостійкості, опишемо класифікацію атак на стеганосистеми і спробуємо визначити умови, в яких стеганосистеми можуть бути стійкими.

Досліджуємо стеганосистеми, завданням яких є прихована передача інформації. У криптографічних системах ховається вміст конфіденційного повідомлення від зловмисника, у той час як у стеганографії додатково ховається факт існування такого повідомлення. Тому визначення стійкості і зламу цих систем різні. У криптографії система захисту інформації є стійкою, якщо, маючи перехвачену криптограму, зловмисник не здатний читати, що міститься в ній. Неформально визначимо, що стеганосистема є стійкою, якщо зловмисник, спостерігаючи інформаційний обмін між відправником і одержувачем, не здатний виявити, що під прикриттям контейнерів передаються засекречені повідомлення, і тим більше читати ці повідомлення.

Назвемо в загальному випадку стеганосистему нестійкою, якщо протиборчі сторони здатні виявляти факт її використання. Розглянемо базову модель стеганосистеми (рис. 5.1), в якій в стеганокодері використовується стеганографічна функція *f* вбудовування за секретним ключем секретного повідомлення *M* в контейнер *C*, а в стеганодекодері стеганографічна функція *φ* його добування по тому ж ключу.

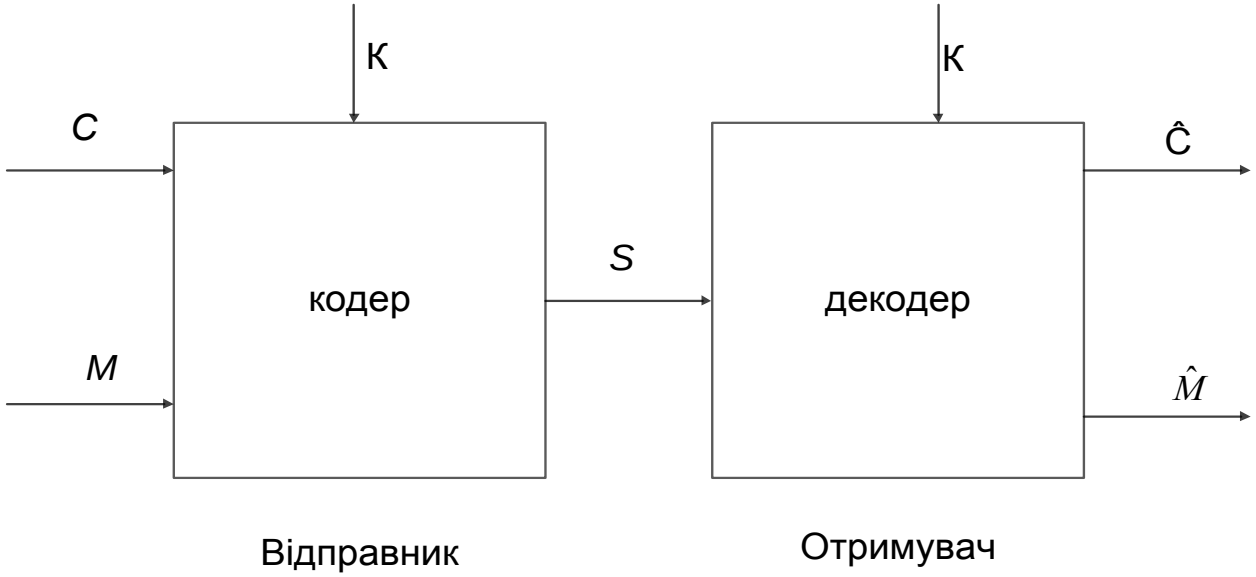


Рис. 5.1. **Базова модель стеганосистеми**

Зі стега по функції *φ* витягується убудоване повідомлення і при необхідності контейнер.

У результаті спотворення при вбудовуванні вплив випадкових і навмисних перешкод передачі, а також похибок при отриманні відновленого одержувачем повідомлення $\hat{M}$ може відрізнятися від оригіналу *M*. Аналогічно, отриманий контейнер $\hat{C}$ буде відрізнятися від вихідного *C*. Контейнер обов'язково буде спотворюватися при вбудовуванні секретного повідомлення. У ряді стеганосистем необхідно відновлювати контейнер, оскільки він фізично становить звичайні повідомлення (зображення, мовні сигнали тощо) кореспондентів відкритого зв'язку, під прикриттям яких здійснюється прихований зв'язок.

Ці повідомлення відкритого зв'язку повинні доставлятися їх одержувачам з якістю, що надається встановленими вимогами до достовірності відкритого зв'язку. Однак навіть якщо використовуваний контейнер є тільки переносником секретного повідомлення, ступінь допустимої похибки контейнера також повинен бути обмеженим, бо інакше зловмисник легко виявить факт використання стеганосистеми.

За ознакою використання ключа дана стеганосистема класифікується як симетрична. Логічно припустити, що стійкість стеганосистеми повинна забезпечуватися при використанні несекретних (загальновідомих) функцій вбудовування *f* і витягнення *φ*. Безпека стеганосистем має спиратися на такі принципи їх побудови, при яких якщо зловмисник не знає секретної ключової інформації, то навіть при повному знанні функцій вбудовування і витягнення прихованої інформації, законів розподілу прихованих повідомлень, контейнерів і стега він не здатний встановити факт прихованої передачі інформації.

Розглянемо класифікацію атак зловмисника, який намагається визначити факт прихованої передачі повідомлення та при встановленні цього факту намагався переглядати їх.

Атака тільки з стеганограмою. Зловмиснику відома одна або певна кількість стеганограм і він намагається визначити, чи не містять вони прихованих повідомлень, і якщо так, то намагається читати їх.

Зловмиснику дуже важко зламати стеганосистему в цій атаці. Це пояснюється тим, що при невідомості ні вихідного контейнера, ні якої-небудь частини прихованого повідомлення можна отримати дуже велику кількість помилкових розшифровок, серед яких жодній не можна віддати перевагу. Девід Кан у своїй знаменитій книзі описував, що якщо цензор при перегляді поштових відправлень у роки Другої світової війни не міг відразу знайти слідів прихованих повідомлень, то, швидше за все, ця задача не має однозначного рішення.

Атака з відомим контейнером. Зловмиснику доступні один або множина пар контейнерів і відповідних їм стеганограм. Зауважимо, що в цій атаці зловмисник знає вихідний вигляд контейнера, що дає йому істотні переваги порівняно з першою атакою. Наприклад, в якості відомого зловмиснику контейнера може служити студійний запис музичного твору, що передається радіомовним каналом з вбудованою інформацією. Або в якості контейнера використовується зображення якої-небудь відомої картини, що демонструється в Ермітажі, високоякісна цифрова копія якої вільно продається на CD-дисках.

Атака з обраним контейнером. Зловмисник здатний нав’язати для використання в стеганосистемі конкретний контейнер, що володіє якимись перевагами для проведення стеганоаналізу порівняно з усією безліччю контейнерів. Удосконалена версія цієї атаки: атака з адаптивно обраними контейнерами. Зловмисник нав’язує контейнер, аналізує отриманий стег

для формування оцінок ймовірності факту прихованої передачі, або прихованого повідомлення, або використовуваного стеганоключа. На основі отриманих оцінок зловмисник формує черговий контейнер, з урахуванням чергового стега уточнює оцінки і так далі до однозначного встановлення факту наявності прихованого зв'язку або його відсутності, а при виявленні каналу прихованого зв'язку до обчислення використовуваного стеганоключа і читання прихованого листування. Наприклад, така атака може мати місце при несанкціонованому використанні відправником прихованих повідомлень чужого каналу передачі інформації, коли законний власник інформаційних ресурсів проводить розслідування з метою позбутися від непроханих користувачів. Зокрема, в сучасних телекомунікаційних системах відомі спроби безкоштовно скористатися послугами дорогого супутникового та наземного мобільного зв'язку.

Атака з відомим повідомленням. Зловмиснику відомо вміст одного або декількох прихованих повідомлень і він намагається встановити факт їх передачі та/або використовування стеганоключа. Наприклад, така атака виконується тюремником Віллі у класичній завдання про ув'язнених [58]. Віллі, знаючи вигляд повідомлення про втечу, аналізує листування між ув'язненими, щоб виявити момент втечі, що готується. Очевидно, що відшукати сліди конкретного повідомлення в деякій множині переданих стегів суттєво простіше, ніж виявити в цій же множині факт прихованої передачі апріорі невідомого повідомлення.

Якщо зловмиснику відомі деякі приховані повідомлення та відповідні їм стеганограми, то його завданням є визначення ключа стеганосистеми для виявлення та читання інших приховано переданих повідомлень, або при неможливості (високій складності) визначення ключа завданням зловмисника є побудова методів безключового читання або визначення факту передачі прихованої інформації.

Атака з обраним повідомленням. Зловмисник здатний нав'язати для передачі по стеганосистемі конкретне повідомлення і він намагається встановити факт його прихованої передачі, при цьому використовується секретний ключ. Також можлива атака з адаптивно обраним повідомленням, в яке зловмисник послідовно підкидає приховану інформацію підбираємого повідомлення та ітеративно зменшує свою невизначеність про використання стеганосистеми та її параметри.

Наприклад, така атака може виконуватися, коли виникає підозра, що з якого-то автоматизованого робочого місця (АРМ) локальної мережі

установи відбувається витік конфіденційної інформації, яка потім таємно передається за межі цієї мережі. Для виявлення каналу витоку адміністратор безпеки формує повідомлення, які могли б зацікавити недобросовісного користувача та вводить їх в інформаційні масиви мережі. Потім адміністратор намагається виявити сліди цих повідомлень в інформаційних потоках, які передаються з АРМ користувачів через сервер у зовнішні мережі. Для однозначного встановлення факту наявності або відсутності каналу прихованою зв'язку адміністратор вибирає такі повідомлення, які легше інших виявити при їх передачі по стеганоканалу.

Крім того, можливі різні поєднання перерахованих атак, у яких зловмисник здатний знати або вибирати контейнери в яких таємно передаються повідомлення. Ступінь ефективності атак на стеганосистему зростає у міру збільшення знань зловмисника про використовувані контейнери, приховані повідомлення, об'єм перехвачених стеганограм і його можливостей з нав'язування обраних контейнерів та повідомлень.

Введемо моделі зловмисника, який намагається протидіяти прихованій інформації. Дотримуючись К. Шеннон, назвемо першою з цих моделей теоретико-інформаційну [4]. Нехай, як це прийнято для систем захисту інформації, для стеганосистем виконується принцип Кергоффа: зловмисник знає повний опис стеганосистеми, йому відомі ймовірнісні характеристики прихованих повідомлень, контейнерів, ключів, формуються стеганограми. Зловмисник має необмежені обчислювальні ресурси, запам'ятовуючі пристрої довільно великої ємності, має у своєму розпорядженні нескінченно багато часу для стеганоаналізу і йому відомо довільну множину перехвачених стеганограм [61]. Єдине, що невідомо зловмиснику, – використовуваний ключ стеганосистеми. Якщо у даній моделі зловмисник не в змозі встановити, міститься чи ні приховане повідомлення в контрольованому стегі, то назвемо таку стеганосистему *теоретико-інформаційно стійкою* до атак пасивного зловмисника або *досконалою*.

Стійкість різних стеганосистем може бути розділена на стійкість до виявлення факту передачі (існування) приховуваної інформації, стійкість до витягання приховуваної інформації, стійкість до нав'язування помилкових повідомлень за допомогою прихованого зв'язку (імітостійкість), стійкість до відновлення секретного ключа стеганосистеми.

Очевидно, що якщо стеганосистема є стійкою до виявлення факту передачі (існування) приховуваної інформації, то логічно припустити, що вона при цьому є стійкою і до читання приховуваної інформації. Зворотне в загальному випадку неправильне. Стеганосистема може бути стійкою до читання приховуваної інформації, але факт передачі певної інформації під прикриттям контейнера може виявлятися зловмисником. Перефразовуючи відомий вислів Ш. Гольдвассера про несиметричні системи шифрування [61], можна сказати, що якщо накрити верблюда ковдрою, то можна приховати число горбів у верблюда (назвемо це прихованим повідомленням), але важко приховати, що під ковдрою-контейнером щось заховано.

Стійкість стеганосистеми до нав'язування помилкових повідомлень за допомогою прихованого зв'язку характеризує її здатність виявляти і відкидати сформовані зловмисником повідомлення, що вводяться ним в канал передачі прихованих повідомлень з метою видачі їх за істинні, які виходять від законного відправника. Наприклад, якщо у класичній завдання Сіммонс тюремник Віллі виявиться здатним сфабрикувати неправдиве повідомлення про скасування втечі і одержувач Боб повірить, що її автором є законний відправник Аліса, то це означає суттєву слабкість використовуваної стеганосистеми. Якщо в системі ЦВДЗн зловмисник здатний ввести в контейнер, завірений законним відправником, свій водяний знак і детектор буде виявляти водяний знак зловмисника і не виявляти ЦВДЗн істинного відправника, то це означає дискредитацію (злам) системи ЦВДЗн.

Стійкість до відновлення секретного ключа стеганосистеми характеризує її здатність протистояти спробам зловмисника обчислити секретну ключову інформацію даної стеганосистеми. Якщо зловмисник здатен визначити ключ симетричної стеганосистеми, то він може однозначно виявляти факти передачі прихованих повідомлень і читати їх чи нав'язувати помилкові повідомлення без будь-яких обмежень. Таку подію можна назвати повною компрометації стеганосистеми. Очевидно, що атаки зловмисника на ключ стеганосистеми можуть бути побудовані аналогічно атакам на ключ систем шифрування інформації та систем автентифікації повідомлень.

Якщо зловмисник здатний вирахувати ключ вбудованого водяного знака будь-якого автора (власника) інформаційних ресурсів, то він може поставити цей водяний знак на будь-який контейнер. Тим самим

зловмисник дискредитує або водяний знак даного автора (власника), або цілком всю систему ЦВДЗн. В обох випадках ставиться під сумнів законність прав одного або всіх власників інформаційних ресурсів на те, що дійсно їм належить. Ця проблема має велике практичне значення для захисту авторських і майнових прав виробників різного роду інформаційних продуктів, таких, як ліцензійне програмне забезпечення, CD- і DVD-дисків, відео- та аудіокасет і т. д. Світовий ринок інформаційної індустрії оцінюється багатьма мільярдами доларів на рік і тому не дивно, що захист інформації як товару від різних посягань зловмисників швидко набуває конкретної практичної спрямованості.

Якщо система ЦВДЗн побудована як симетрична, то декодер повинен використовувати конфіденційний ключ виявлення водяного знака. Отже, такий детектор проблематично вбудовувати в пристрої, що експлуатуються, до яких доступ зловмисника технічно складно обмежити, наприклад, в персональні програвачі DVD-дисків. *Несиметрична система ЦВДЗн* використовує секретний ключ вбудовування водяного знака у контейнери і відкритий ключ перевірки ЦВДЗн. Очевидно, що з відкритого ключа перевірки повинно бути неможливо обчислення секретного ключа вбудовування водяного знака. Зловмисник не повинен бути здатний у контейнер вмонтувати водяний знак довільного автора (виробника), а сам водяний знак повинен однозначно ідентифікувати цього автора. Вимоги до ключової інформації несиметричних систем ЦВДЗн дуже нагадують вимоги до ключів відомих з криптографії систем цифрового підпису даних. При використанні несиметричних систем ЦВДЗн можна вбудовувати декодери в будь-яке обладнання, не боючись компромета-ції ключа вбудовування водяного знака. Зрозуміло, при цьому треба виключити можливість обходу зловмисником системи захисту. Якщо зловмисник може відключити детектор ЦВДЗн, то він зможе не санкціоновано скористатися платними інформаційними ресурсами. Наприклад, в сучасні DVD-пристрої записується інформація про географічні регіони їх виробництва та продажу, в межах якого дозволяється або обмежується програвання DVD-дисків з відповідними мітками доступу. Україна відповідно до цього розмежування доступу відноситься до регіону, в якому ймовірність електронної крадіжки значно вища, ніж, наприклад, у Західній Європі.

Зауважимо, що побудова несиметричних систем ЦВДЗн та інших стеганосистем викликає суттєві практичні проблеми. По-перше,

несиметричні системи, як відомо з криптографії, в реалізації виявляються обчислювально складніше симетричних систем. По-друге, крім вимог до стійкості ключа стеганосистеми, висуваються жорсткі вимоги до стійкості системи ЦВДЗн до різноманітних спроб зловмисника перекручування водяного знака. Несиметричні системи побудовані на основі односпрямованих функції з потайним ходом, ідея яких запропонована У. Діффі та М. Хелманом [9]. Принципи побудови переважної більшості відомих односпрямованих функцій з потайним ходом такі, що будь-яке завгодно мале спотворення вихідного значення цих функцій при використанні законним одержувачем потайного ходу призводить до істотного розмноженню помилок у повідомленні, що приймається. Цей недолік односпрямованих функцій характерний і для нині використовуваних несиметричних криптографічних систем. Однак там його можна компенсувати використанням додаткових заходів підвищення вірогідності переданих криптограм або цифрових підписів повідомлень. Але в стеганосистемах використання цих же способів підвищення достовірності ускладнене. По-перше, їх застосування демаскує прихований канал. По-друге, активний зловмисник в атаках на стеганосистему ЦВДЗн має великі можливості підібрати такий руйнівний вплив, при якому доступні методи підвищення вірогідності інформації можуть виявитися неефективними. Наприклад, якщо відправник використовує алгоритми завадостійкого кодування, які забезпечують захист приховуваного повідомлення від рівновірогідних розподілених помилок, то зловмисник підбирає закон розподілу пакетувальних помилок, при якому канальний декодер отримувача не здатний їх виправити і розмножує помилки при декодуванні.

Для аналізу стійкості стеганографічних систем до виявлення факту передачі секретних повідомлень розглянемо теоретико-інформаційну модель стеганосистеми з пасивним зловмисником, запропоновану в роботі [30].

Зловмисник Єва спостерігає повідомлення, що передаються відправником Алісою одержувачу Бобу. Єва не знає, містять ці повідомлення нешкідливий контейнер C або стег S з приховуваною інформацією. Будемо вважати, що Аліса може знаходиться в одному з двох режимів: вона або активна (і тоді через контрольований канал передається стег S) або пасивна (передається порожній контейнер C). Коли Аліса активна, вона перетворює контейнер з вкладенням в нього секретного

повідомлення *M*, використовуючи секретний ключ *K*. Можлива побудова стеганосистеми, в якій Аліса може сама генерувати для приховування повідомлення *M* прийнятний контейнер. Отримавши стег *S*, Боб повинен бути здатний витягти з нього повідомлення *M*, використовуючи ключ *K*. У термінах теорії інформації стеганосистема повинна задовольняти співвідношенням:

1. $H(S / CMK) = 0$. Сформоване відправником стега *S* однозначно визначається значеннями контейнера *C*, ключа *K* і повідомлення *M*.

2. $H(M) > 0$. Невизначеність до моменту передачі прихованого повідомлення *M* і для одержувача і для зловмисника строго більше нуля, інакше через прихований канал не передається невідома для одержувача інформація та втрачається сенс використання стеганосистеми. Зауважимо, що це може не виконуватися для систем ЦВДЗн, метою яких є не передача невідомої для одержувача або зловмисника інформації, а встановлення авторства переданого контейнера.

3. $H(M / SK) = 0$. Одержувач Боб має однозначно відновити приховане повідомлення *M* з прийнятого стега *S*, користуючись ключем *K*.

Будемо вважати, що імовірнісні розподілу множин приховані повідомлення, контейнери, стеги і ключі відомі для всіх учасників інформаційного протиборства. Додатково одержувач Боб знає, активна чи ні відправник Аліса. Єва, спостерігаючи повідомлення, що передаються Алісою повідомлення, повинна встановити, передається каналом зв'язку приховувана інформація чи ні. Для виявлення факту використання стеганосистеми Єва намагається визначити, чи відповідає інформаційний потік, що передається, розподілу контейнерів або розподілу стега. Якщо Єва здатна встановити, що в контрольованому каналі передаються повідомлення з розподілом стеганограм, то факт прихованої передачі інформації від Аліси до Боба доведено, а використовувана ними стеганосистема є нестійкою. Сформулюємо завдання доказу стійкості стеганосистеми як задачу розрізнення гіпотез.

У розглянутій моделі стеганосистеми відомо ймовірнісний розподіл порожніх контейнерів, які позначаються P_C, і ймовірнісний розподіл стеганограм, які позначаються P_S. Зловмисник у контрольованому каналі зв'язку може спостерігати множину можливих порожніх контейнерів і стеганограм. Позначимо цю множину можливих спостережень *Q*. Зловмисник, спостерігаючи передане повідомлення $q \in Q$, висуває дві гіпотези H_C і H_S. Якщо справедлива гіпотеза H_C, то повідомлення *q*

породжено згідно з розподілом P_C, а якщо справедлива H_S, то q відповідає розподілу P_C. Правило рішення полягає в розбитті множини Q на дві частини так, щоб призначити одну з двох гіпотез кожному можливому повідомленню $q \in Q$. У цій завдання розрізнення можливі два типи помилок: помилка першого типу, яка полягає у встановленні гіпотези H_S, коли вірною є H_C, і помилка другого типу, коли прийнято рішення H_C при вірній гіпотезі H_S. Ймовірність помилки першого типу позначається α, імовірність помилки другого типу – β.

Метод знаходження оптимального рішення задається теоремою Неймана – Пірсона. Правило рішення залежить від порога T. Змінні α і β залежать від T. Теорема встановлює, що для деякого заданого порога T і допустимої максимальної ймовірності α, імовірність β може бути мінімізована призначенням такої гіпотези H_C для спостереження $q \in Q$, якщо і тільки якщо виконуються

$$\log \frac{P_C(q)}{P_S(q)} \geq T. \qquad (5.1)$$

Основним інструментом для розрізнення гіпотез є відносна ентропія (ВЕ) або відношення між двома ймовірностями P_C і P_S, що визначається у вигляді

$$D(P_C \parallel P_S) = \sum_{q \in Q} P_C(q) \log \frac{P_C(q)}{P_S(q)}. \qquad (5.2)$$

Відносна ентропія між двома ймовірностями завжди невід'ємна і дорівнює 0, якщо і тільки якщо вони не відрізняються (збігаються). Хоча в математичному сенсі ВЕ не є метрикою, оскільки вона не має властивостей симетричності і трикутника, корисно її використовувати в якості відстані між двома порівнюваними розподілами. Двійкова відносна ентропія $d(\alpha,\beta)$ визначається як

$$d(\alpha,\beta) = \alpha \log \frac{\alpha}{1-\beta} + (1-\alpha) \log \frac{1-\alpha}{\beta}.$$

Використовуємо відносну ентропію $D\ (P_c \parallel P_s)$ між відділами P_c і P_s для оцінки стійкості стеганосистеми при пасивному зловмиснику. В роботі [30] дано таке визначення: стеганосистема називається ε-стійкою проти пасивного зловмисника, якщо

$$D\ (P_C \parallel P_S) \leq \varepsilon.$$

Якщо $\varepsilon = 0$, то стеганосистема є досконалою.

Якщо розподіли контейнера і стега однакові, то $D(P_c \| P_s) = 0$, і така стеганосистема є досконалою. Це означає, що ймовірність виявлення факту передачі приховуваної інформації не змінюється від того, спостерігає зловмисник інформаційний обмін від Аліси до Боба чи ні. Пасивний зловмисник, що володіє довільно великими ресурсами та будь-якими методами стеганоаналізу, не здатний виявити факт використання досконалої стеганосистеми.

Розглянемо умови забезпечення стійкості стеганосистем . Відомо співвідношення між ентропією, відносною ентропією і розміром алфавіту $|\mathbf{X}|$ для довільних випадкових змінних S і C. Зазначимо, що контейнери C і стега S належать одному і тому ж алфавіту X. Якщо мінлива S рівновірогідна і незалежно розподілена, то

$$H(C) + D(P_C \| P_S) = \log|\mathrm{X}|. \qquad (5.3)$$

Якщо мінлива C є рівновірогідною і незалежно розподіленою, то, як відомо з теорії інформації [10], виконується рівність $H(C) = \log|\mathrm{X}|$ і тоді $D(P_C \| P_S) = 0$. Отже, якщо в якості контейнерів C використовувати випадкові послідовності і приховуване повідомлення буде описуватися також випадковими послідовностями, то сформовані стеги S не будуть мати жодних статистичних відмінностей від порожніх контейнерів, і така стеганосистема буде досконалою. Якщо приховувана інформація становить осмислені повідомлення, які описуються послідовностями з нерівномірними і залежними між собою символами, то до необхідного вигляду їх легко привести шляхом шифрування будь-яким стійким шифром.

Опишемо приклад *формально досконалої стеганосистеми*, в якій контейнери становлять послідовності незалежних та рівновірогідних випадкових бітів і в якості функції вбудовування приховуваних повідомлень використовується відома криптографічна функція типу "однократної підстановки". Нехай контейнер C є рівновірогідно розподіленою випадковою послідовністю довжиною n бітів. Формувач ключа генерує випадкову рівновірогідно розподілену послідовність ключа k довжиною n бітів і передає її Алісі і Бобу. Якщо Аліса активна, то функція вбудовування становить побітне сумування за модулем 2 для приховування n-бітового повідомлення m, де стег формується за правилом $s = m \oplus k$. Одержувач Боб витягує приховане повідомлення обчислення $m = s \oplus k$. Сформований стег S рівновірогідно розподілений

для послідовності n бітів і тому $D\ (P_C \,||\, P_S) = 0$. Таким чином, побудова функції вбудовування як однократної підстановки забезпечує абсолютні стеганосистеми, якщо контейнер формується рівновірогідним випадковим джерелом.

Однак реальні передані каналами зв'язку повідомлення, що використовуються в стеганосистемах як порожні контейнери, далекі від моделі беззбиткових і рівновірогідних джерел. Тому передача зашифрованих описаним способом повідомлень на тлі повідомлень природних джерел відразу ж демаскує канал прихованого зв'язку. Для стеганографії характерний випадок не рівновірогідного розподілу змінної C, описує вихід природного джерела з деякою суттєвою пам'яттю. Повідомлення таких джерел зазвичай використовуються в якості контейнерів (зображення, мова тощо) та їх ентропія $H(S)$ зазвичай значно менше величини $\log|X|$. Для вбудовування приховуваних повідомлень з таких контейнерів видаляється частина надмірності і в стислі таким чином контейнери вкладаються приховувані повідомлення. В результаті цього імовірнісні характеристики формованих стеганограм відрізняються від характеристик порожніх контейнерів, наближаючись до характеристик випадкового незалежного джерела. У граничному випадку дискретні стеганограми описуються бернулієвським розподілом. У цьому випадку вся надмірність контейнера вилучена та вбудоване повідомлення породжено рівновірогідним випадковим джерелом.

Розглянемо такий приклад. Нехай в якості контейнерів використовуються повідомлення типу "ділова проза" українською мовою, для яких відома оцінка ентропії $H(C) = 0{,}83$ біти / буква [67]. Величина $\log|X|$ для української мови з алфавітом з 32 літер становить $\log 32 = 5$. Отже, в граничному випадку відносна ентропія між звичайними повідомленнями з розподілом P_C і стеганограмами з розподілом P_S дорівнює:

$$\varepsilon \geq D\ (P_C \,||\, P_S) = \log|X| - H(C) = 5 - 0{,}83 = 4{,}17 \text{ [біти/буква]}.$$

Очевидно, що в цьому випадку беззбиткові стеги, виглядають як випадковий набір букв української мови і відразу ж виділяються на тлі надмірних контейнерів, які становлять осмислені повідомлення. Таким чином, факт використання такої стеганосистеми легко виявляється при візуальному перегляді переданих від Аліси до Боба повідомлень.

При використанні такої стеганосистеми також легко автоматизувати процес пошуку слідів прихованого каналу. Для цього достатньо підраховувати приблизні оцінки ентропії переданих повідомлень. Оскільки ентропія стега приблизно в 5 разів більше ентропії звичайних повідомлень, то достатньо просто виявити факти наявності прихованого зв'язку.

У роботі [30] доводиться, що довільні детерміновані перетворення не збільшують ВЕ між двома розподілами.

Теорема 5.1. Нехай P_{Qc} і P_{Qs} описують ймовірнісні розподіли контейнерів і стега, відповідно, над множиною спостережень Q. Детерміновані відображення f перетворюють множину спостережень Q у множину спостережень T виду

$$f:Q \to T, \quad t_c = f(q_c),\ t_s = f(q_s),$$

де $q_c,\ q_s \in Q,\ t_c,\ t_s \in T$. Тоді справедливим є вираз:

$$D\ (P_{Tc} \,||\, P_{Ts}) \le D\ (P_{Qc} \,||\, P_{Qs}).$$

Оскільки розрізнення між гіпотезами H_C і H_S є приватною формою перетворення, ймовірності помилок α і β задовольняють нерівності:

$$d\ (\alpha,\beta) \le D\ (P_{Qc} \,||\, P_{Qs}). \qquad (5.4)$$

Це співвідношення може використовуватися в такому вигляді: нехай δ є верхня межа $D\ (P_{Qc} \,||\, P_{Qs})$ і задана верхня межа ймовірності α. Тоді вираз (5.5) дає нижню межу ймовірності β. Наприклад, при α = 0 значення помилки $\beta \ge 2^{-\delta}$.

Використовуючи цю теорему, в роботі [5] доводиться така теорема.

Теорема 5.2. Якщо стеганосистема є ε-стійкою проти пасивного зловмисника, то ймовірність β невиявлення факту прихованого зв'язку та ймовірність α помилкового встановлення факту прихованого зв'язку задовольняють нерівності $d\ (\alpha,\beta) \le \varepsilon$. В окремому випадку, якщо α = 0, то $\beta \ge 2-\varepsilon$.

Нехай Алісі дозволяється передати Бобу цифрове зображення C. Використовуючи модель чутливості зору, вона може сформувати множину C еквівалентних зображень, які візуально невідрізняються від вихідного C. Незалежно від того, активна Аліса чи ні, вона передає

вибране зображення з множини *C*. Нехай Аліса і Боб заздалегідь домовилися, якій модифікації зображення відповідає кожне з прихованого повідомлення. Формально це означає, що в стеганосистемі кожному із зображень C_j, де j = 1,2, ..., з секретного ключа ставиться у відповідність або одне з прихованих повідомлень M_j, де j = 1,2, ..., N, і N <, або відсутність прихованого повідомлення для $|C|-N$ випадків. Якщо ця відповідність побудована рівновірогідною і незалежно для множини контейнерів і приховуваних повідомлень, то при розрізнюваності розподілів контейнерів і стега зловмисник Єва, спостерігаючи за інформаційним обміном між Алісою та Бобом, потенційно не здатна отримати більше тієї інформації, яку мала апріорі. Оскільки за визначенням Еві відомі статистичні характеристики всіх множин, що входять у стеганосистему, то вона апріорі знає, що ймовірність активного стану Аліси дорівнює $N/|C|$, а ймовірність відсутності передачі приховуваної інформації дорівнює $(|C|-N)/|C|$. Активний і пасивний стан Аліси складає повну групу подій, отже,

$$N/|C|+(|C|-N)/|C|=1.$$

Таким чином, якщо Аліса збирається передавати N приховуваних повідомлень під прикриттям контейнерів, то ймовірність того, що Єва вгадає, що довільний контейнер містить вкладену інформацію, не може бути менше величини $N/|C|$. Якщо стеганосистема досконала, то ймовірність підбору зловмисником факту передачі приховуваного повідомлення суворо дорівнює цій величині.

З цього випливає, що ймовірність пасивного стану Аліси повинна бути в багато разів більше ймовірності її активного стану, і що використання контейнерів з урахуванням їх модифікацій повинно бути в багато разів більше приховуваних повідомлень. Перефразовуючи відому приказку, можна сказати, що голку більш надійно можна приховати від чужих очей у великому стозі сіна, ніж у маленькому.

Розглянемо вплив деякої додаткової інформації на розподіл контейнерів і стега і, відповідно, на стійкість стеганосистеми. Нехай деякі зовнішні події впливають на розподіл контейнерів, наприклад, випуски новин або погоди у відомій "завдання ув’язнених". Ця додаткова інформація позначається *Y* і відома всім учасникам. Відповідно змінимо нашу модель і визначення стійкості. Визначимо середню ймовірність

виду $\overline{\alpha} = \sum_{y \in Y} P_Y(y)\alpha(y)$ для помилок 1 роду і $\overline{\beta} = \sum_{y \in Y} P_Y(y)\beta(y)$ для помилки 2 роду, де α *(y)* і β *(y)* означають, відповідно, величину ймовірностей помилок 1 та 2 роду для *Y = y*.

Умовна відносна ентропія (УВЕ) між P_C і P_S, що належать одній абетці *X*, залежно від змінної *Y*, визначається у вигляді:

$$D(P_{C/Y} \parallel P_{S/Y}) = \sum_{y \in Y} P_Y(q) \sum_{c,s \in X} P_{C/Y=y}(q) \log \frac{P_{C/Y=y}(q)}{P_{S/Y=y}(q)}. \qquad (5.5)$$

З нерівності Іенсена [10] і з виразу (5.4) випливає, що:

$$d(\overline{\alpha}, \overline{\beta}) \le D(P_{C/Y} \parallel P_{S/Y}). \qquad (5.6)$$

Стеганосистема з додатковою інформацією *Y*, контейнерами *C* і стегами *S* називається *ε-стійкою проти пасивного противника*, якщо умовна відносна ентропія $D\ (P_{C/Y} \parallel P_{S/Y}) \le \varepsilon$. В якості прикладу використання в стеганосистемі зовнішньої інформації вкажемо "класичне" завдання Г. Сіммонса, в якому ув'язнені приховано обмінюються інформацією про втечу. Ймовірність передачі повідомлення про втечу в темну ніч вище порівняно зі світлою ніччю. Це загальновідомий факт не тільки для осіб, що здійснюють втечу, але й для їх тюремників, що посилює контроль за можливими каналами прихованої передачі інформації. Тому використання загальновідомої додаткової інформації в стеганосистемі полегшує завдання зловмисника. Можна сказати, що ε-стійка стеганосистема з додатковою інформацією *Y* забезпечує більш високу скритність зв'язку порівняно з аналогічною ε-стійкою стеганосистемою без цієї інформації.

У простіших стеганосистемах ЦВДЗ при вбудовуванні використовується псевдовипадкова послідовність, що є реалізацією гаусівського білого шуму і не враховує властивості контейнера. Такі системи практично нестійкі до більшості розглянутих вище атак. Для підвищення робастності стеганосистем можна запропонувати ряд поліпшень.

У попередньому розділі було показано, що на основі аналізу розподілів контейнерів і розподілів стегів виявляється факт використання стеганосистеми. Для цього в розглянутій теоретико-інформаційній моделі передбачається, що зловмисник знає точні імовірнісні характеристики контейнерів, стегів, приховуваних повідомлень та ключів. Також в моделі

передбачається, що передаються стеганограми та порожні контейнери, які не зазнають жодних викривлень у процесі їх доставки по каналу зв'язку, а відправник приховуваних повідомлень вибирає тільки такі контейнери, характеристики яких збігаються з характеристиками всієї множини контейнерів. У результаті будь-яке відхилення статистики контрольованого зловмисником у каналі зв'язку повідомлення від середньостатистичних характеристик порожніх контейнерів повинно кваліфікуватися як факт виявлення стеганоканалу. Очевидно, що така ідеальна модель не цілком адекватна реаліям інформаційно-приховуючих систем. По-перше, зловмисник знає характеристики не дійсно використаного відправником контейнера, а усереднені характеристики множини повідомлень деяких джерел, які потенційно можуть бути використані в якості контейнера. По-друге, всі відомі джерела можливих контейнерів у силу їх природи є нестаціонарними, тобто їх точних оцінок не існує. По-третє, приховувати інформацію для вбудовування приховуваної інформації вільні вибирати з усієї множини такі контейнери, характеристики яких відрізняються від відомих зловмиснику характеристик цієї множини. Більш того, відправник може підбирати такі контейнери або спеціально їх генерувати, щоб при вбудовуванні в них приховуваних повідомлень характеристики сформованого стега були б невідмінні від середньостатистичних характеристик порожніх контейнерів. По-четверте, у сучасних комуніка-ційних системах передаються надлишкові повідомлення, як правило, стискуються з внесенням деяких допустимих для їх одержувачів спотворень, що змінюють їх характеристики. Наприклад, мовленнєвий сигнал кодується методами лінійного передбачення мови, зображення стискуються алгоритмами JPEG, MPEG або H.263. І, по-п'яте, канал зв'язку може вносити перешкоди в інформаційні потоки, що передаються. А якщо канал досконалий, то відправник для маскування може сам зашумляти передавані стеги та порожні контейнери такими перешкодами, які в допустимих межах передачі повідомлення, в достатній для приховування мірі модифікують статистику стегів та контейнерів.

Перераховані причини приводять до моделі стеганосистеми, в якій зловмисник може бути здатним визначити, що статистика спостережуваних їм у каналі послідовностей відрізняється від відомої йому статистики контейнерів, але він не здатний встановити причину цих відмінностей. Таким чином, зловмисник хоча і підозрює про існування прихованого каналу, але не може довести або спростувати цього.

Необхідні докази можуть бути отримані, якщо зловмисник зуміє прочитати приховуване повідомлення. Методами теорії інформації опишемо стійкість стеганосистеми до читання приховуваних повідомлень.

У роботі [48] дещо з інших позицій, ніж у підході роботи [30], визначається стійкість стеганосистеми. *Стеганосистема* називається *теоретико-інформаційно стійкою*, якщо зловмисник не здатний отримати ніякої інформації про вбудоване повідомлення, аналізуючи перехвачені стеги за умови знання статистичних характеристик порожніх контейнерів. У рамках цього визначення підраховується взаємна інформація між приховуваними повідомленнями M і множинами стегів S і відповідних їм контейнерів C. У теоретико-інформаційно стійкій, або, інакше кажучи, досконалій стеганосистемі, повинна виконуватися рівність. Як відомо з теорії інформації [52], взаємна інформація може бути визначена через безумовну і умовну ентропію:

$$I(M;(S,C)) = H(M) - H(M/(S,C)) = 0. \qquad (5.7)$$

Це дає фундаментальні умови стійкості стеганосистеми виду

$$H(M/(S,C)) = H(M). \qquad (5.8)$$

Таке визначення теоретико-інформаційної стійкості стеганосистеми дуже нагадує відповідне визначення теоретико-інформаційної стійкості системи шифрування інформації. Якщо невизначеність зловмисника щодо повідомлення M не зменшується при перехопленні криптограми E, то за визначенням К. Шеннона дана система шифрування є досконалою [4]:

$$H(M/E) = H(M). \qquad (5.9)$$

Зауважимо, що вирази (5.8) та (5.9) вказують, що зловмисник не здатний визначити ні одного біта захищеного повідомлення. При цьому для системи шифрування точно відомо, що в криптограмі це повідомлення міститься. Для стеганосистеми вираз (5.8) може виконуватися в таких випадках:

1. Стеганосистема не використовується.

2. Здійснюється прихована передача інформації, яка використовується довершена до встановлення факту наявності прихованого зв’язку стеганосистеми. Якщо зловмисник не здатний визначити факт

наявності прихованого повідомлення, то тим більше він не здатний прочитати жодного біта цього повідомлення.

3. Здійснюється прихована передача інформації, зловмисник здатний визначити факт наявності прихованого зв'язку. Однак він не здатний прочитати жодного біта прихованого повідомлення.

Наприклад, третій випадок був описаний у попередньому розділі при вкладенні беззбиткових прихованих повідомлень у рівновірогідні випадкові контейнерні послідовності за функцією вбудовування однократної підстановки. Сформовані таким чином стеги легко виявляються зловмисником на тлі звичайних надлишкових повідомлень. Однак прочитати ці повідомлення принципово неможливо, якщо при вбудовуванні використовується випадкова рівновірогідна розподілена ключова послідовність.

Вираз (5.8) означає, що невизначеність зловмисника щодо повідомлення *M* не повинна зменшуватися при знанні їм стега *S* і контейнера *C*, тобто *M* повинно бути незалежно від *S* та *C*. Дослідимо умови стійкості стеганосистем. Вважаємо, що не тільки алфавіти *S* і *C*, але й їх ентропії *H (S)* і *H (C)* рівні. Розглянемо два випадки.

1. Нехай ніяке повідомлення *M* не вбудовується в контейнер *C*. Очевидно, що в цьому випадку, коли *S* і *C* збігаються, то виконується *H (S / C) = H (C / S)* = 0.

2. У стега S є повідомлення *M* з ентропією *H (M)*> 0. Очевидно, що при наявності цієї вбудованої інформації у зловмисника з'являється відмінна від нуля невизначеність щодо *S*, якщо відомо *C* і невизначеність щодо *C*, якщо відомо *S*: *H (S / C))*> 0, *H (C / S)*> 0. Отже, взаємна інформація між приховуваними повідомленнями та відповідними контейнерами і стегами вже не може бути рівною нулю:

$$I(M;(S,C)) = H(M) - H(M/(S,C)) > 0.$$

Тому,

$$H(M/(S,C)) < H(M). \qquad (5.10)$$

Це означає, що умова стійкості стеганосистеми не забезпечується. Можна показати, що необхідною і достатньою умовою стійкості є:

$$H(S/C) = H(C/S) = 0. \qquad (5.11)$$

Тому у роботі [48] робиться висновок, що якщо зловмиснику відомі стеганограми і відповідні їм контейнери, то стеганосистема не може бути досконалою. У рамках теоретико-інформаційної моделі розглянута стеганосистема в атаці зловмисника з відомим контейнером не може приховати факти передачі приховуваного повідомлення. А з виразу (5.10) випливає, що зловмисник також здатен дізнатися, якщо не повністю, то частково, зміст цього повідомлення: якщо $I(M;(S,C))>0$, то при відомих *S* і *C* невизначеність зловмисника про цей допис менше його ентропії.

Забезпечення необхідної стійкості може бути отримано при переході від детермінованих стеганосистем до недетермінованих (імовірнісних). Розглянемо один з можливих варіантів побудови ймовірнісної стеганосистеми, запропонованої в роботі [48]. У розглянутій ймовірнісній стеганосистемі для виконання необхідної і достатньої умови стійкості виду *H (S / C) = H (C / S)* = 0 забезпечується невідомість для зловмисника використовуваного контейнера. Для цього в модель стеганосистеми вводиться джерело контейнерів *CS*, характеристики якого відомі зловмиснику. Для вбудовування приховуваних повідомлень із множині *CS* випадково і рівновірогідно виберемо підмножину контейнерів *C*, яку назвемо підмножиною дійсних контейнерів: $C \subseteq CS$. Нехай виконується умова $H(CS) \geq H(C)$ і імовірнісні характеристики підмножини *C* відрізняються від відповідних характеристик множини *CS*. Зажадаємо, щоб невизначеність зловмисника щодо дійсних контейнерів при відомій множині *CS* була б суворо більше нуля: $H(C / CS) > 0$. Фізично це може бути забезпечено, якщо вибір дійсних контейнерів здійснюється за допомогою випадкового і рівновірогідного значення *R*, отриманого з виходу генератора випадкових чисел, як показано на рис. 5.2.

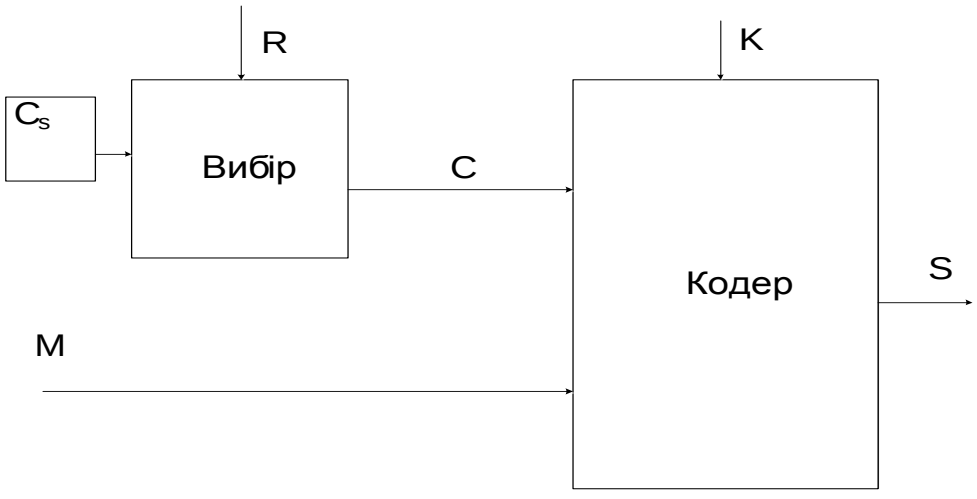


Рис. 5.2. **Стеганосистема з вибором рандомізованого контейнера**

Необхідна невизначеність щодо *C* досягається вибором кожного контейнера зовсім випадковим чином і збереженням вибору в таємниці. Прикладом такого процесу може бути отримання вибірок з вхідного аналогового сигналу, такого, як мова або відео. Погрішність квантователя забезпечує необхідну невизначеність. Якщо зміни контейнера в процесі вбудовування інформації залишаються в межах похибки квантователя, то така маніпуляція не може бути виявлена.

Визначимо, що для розглянутої ймовірнісної стеганосистеми основна умова стійкості виражається у вигляді:

$$H(M/(S,C_S)) = H(M). \qquad (5.12)$$

Це означає, що невизначеність зловмисника щодо *M* не може бути зменшена знанням *S* і C_S, або *M* є незалежним від *S* і C_S.

Досліджуємо умови, при яких зловмисник не здатний виявити зміни в контейнері, що відбулися при вбудовуванні повідомлення *M* з ентропією *H (M)*, спостерігаючи стеги. Для цього визначимо необхідну величину невизначеності зловмисника щодо контейнера *H (C / S)*. Можна показати, що

$$H(C/S) \geq I(M;(S,C)) = H(M) - H(M/(S,C)). \qquad (5.13)$$

При найгіршому випадку супротивник здатний повністю визначити *M* з *S* і *C*: $H(M/(S,C)) = 0$.

Отже, у загальному випадку виконується:

$$H(C/S) \geq H(M). \qquad (5.14)$$

Оскільки взаємна інформація $I(M;(S,C))$ не може бути більше величини *H (M)*, невизначеність $H(C/S)$ повинна бути, принаймні, тієї ж величини, щоб зробити читання повідомлення неможливим.

У стійкої стеганосистеми зловмисник, спостерігаючи стеги *S*, не повинен отримати інформацію, окрім тієї, що йому відома апріорі із знання множини C_S:

$$H\ (C/C_S) = H\ (C/S), \qquad (5.15)$$

і, тому

$$H\ (C/C_S) \geq H\ (M). \qquad (5.16)$$

Таким чином, для зловмисника, що знає характеристики множини C_S, у стійкій стеганосистемі невизначеність щодо підмножини дійсних контейнерів C повинна бути не менше ентропії приховуваних повідомлень.

Визначимо спільну ентропію H_0 між множинами C і C_S.

$$H_0 = H(C,C_S) = H(C) + H(C_S/C). \quad (5.17)$$

Оскільки $C \subseteq C_S$ і $H(C_S) \geq H(C)$, то

$$H(C_S/C) \geq H(C/C_S).$$

Для стійкої стеганосистеми отримаємо нижню межу величини спільної ентропії

$$H_0 \geq H(C) + H(C/C_S).$$

Використовуючи вираз (5.17), запишемо

$$H_0 \geq H(C) + H(M). \quad (5.18)$$

Оскільки $H(C_S) \geq H(C)$, то $H(C_S,S) \geq H(C,S)$. Отже,

$$H(C_S,S) \geq H(C,S). \quad (5.19)$$

Відповідно до виразу (5.14) отримаємо, що межа може бути визначена у вигляді:

$$H(C_S,S) \geq H(M). \quad (5.20)$$

Сформуємо висновок: при досягненні нижньої межі для $H(C / S)$ (рівняння 5.14), зловмисник, що знає S і C_S, не здатний отримати доступ до приховуваного в стегі S повідомленням M. Фундаментальна умова стійкості (5.12) може бути виконана.

Розглянемо умови, при яких зловмисник не здатний визначити ключ до стеганосистеми. Зажадаємо, щоб зловмисник, що знає S і C_S, не міг отримати ніякої інформації ні про ключі K, ні про повідомлення M. Це може бути виражено у вигляді:

$$I((K,M);(S,C_S)) = H(K,M) - H((K,M)/(S,C_S)) = H(K,M) - \quad (5.21)$$
$$- H(K/(S,C_S)) - H(M/(S,C_S,K)) = 0.$$

При знанні ключа K, множини C_S зі стега S однозначно витягується повідомлення M:

$$H\,(M/(S,C_S,K)) = 0.$$

Тому з виразу (5.21) отримаємо:

$$H\,(K/(S,C_S) = H\,(K,M),$$

або

$$H\,(K/(S,C_S) = H\,(M) + H\,(K/M) \geq H\,(M). \qquad (5.22)$$

Відповідно, оскільки $H\,(K/M) \geq 0$.

Таким чином, для зловмисника невизначеність ключа стійкої стеганосистеми повинна бути не менше невизначеності переданого прихованого повідомлення. Це вимога для стеганосистем дуже схожа на вимогу невизначеності ключа K для досконалих систем шифрування, для яких ентропія ключа K при перехваченій криптограмі E повинна бути не менше ентропії шифрованого повідомлення M [4]:

$$H(K/E) \geq H(M).$$

Робимо висновок, що дійсний контейнер повинен бути невідомим для зловмисника, щоб забезпечити теоретико-інформаційну стійкість стеганосистеми. Зловмисник не здатен ні виявити факт передачі приховуваного повідомлення, ні читати його, якщо виконуються дві умови:

1. Знання S і C_S не зменшує для зловмисника невизначеності про приховані повідомлення:

$$H\,(M/(S,C_S)) = H\,(M/S) = H\,(M).$$

2. Умовна ентропія ключа повинна бути не менше ентропії приховуваного повідомлення:

$$H\,(K/(S,C_S)) \geq H\,(M).$$

За таких умов необхідна стійкість може бути забезпечена в ймовірнісних стеганосистемах.

У роботі [48] наводяться загальні описи можливих імовірнісних стеганосистем. Нехай відправник для вбудовування приховуваних повідомлень в якості дійсних контейнерів використовує цифрове

зображення пейзажу на виході електронної камери. Зловмисник може знати загальний вигляд зображення, що знімається, й характеристики використовуваної камери. Але атакуючий і навіть законний отримувач не знають точне положення камери й кут зйомки. Коливання камери навіть на частку градуса призводить до істотно відмінних знімків. Тому при аналізі зловмисником перехваченого стега він не здатний визначити, яке цифрове зображення є дійсним контейнером, і тим самим не може виявити відмінності між стегом та контейнером. В якості множини контейнерів *CS* в даному прикладі використовуються всілякі варіанти зображення пейзажу під різними кутами з урахуванням неідеального оптико-електронного перетворювача використовуваної камери.

Другим прикладом ймовірнісної стеганосистеми є використання в якості дійсних контейнерів значень відліків аналогового випадкового сигналу, наприклад, речевого. У різних технічних пристроях для перетворення аналогових сигналів до цифрового виду використовуються аналого-цифрові перетворювачі з деякою похибкою квантування відліків, причому моменти дискретизації відліків визначаються тактовим генератором, положення стробіруючіх імпульсів якого також має деяку похибку. Отже, для зловмисника, який точно знає характеристики аналогового сигналу, існує невизначеність між аналоговим і цифровим представленням сигналу. При використанні такого сигналу в якості контейнера потенційно можна побудувати стійку стеганосистему, якщо ентропія вбудовуваного повідомлення не перевищує величини вказаної невизначеності [53].

Раніше розглянуті теоретичні оцінки стійкості стеганосистем, наприклад, теоретико-інформаційні, які припускають, що той, хто приховує інформацію, і зловмисник володіють необмеженими обчислювальними ресурсами для побудови стеганосистем і, відповідно, стеганоатак на них, дотримуються оптимальних стратегій приховування перетворення та стеганоаналізу, мають нескінченний час для передачі і виявлення приховуваних повідомлень і т. д. Зрозуміло, такі ідеальні моделі приховувати інформацію і зловмисника незастосовні для реалій практичних стеганосистем. Тому розглянемо відомі до цього часу практичні оцінки стійкості деяких стеганосистем, що реально використовується для приховування інформації [3; 50; 62].

В останні роки з’явилися програмно реалізовані стеганосистеми, що забезпечують приховування інформації в цифрових відео- та аудіо.

Такі програми вільно поширюються, легко встановлюються на персональні комп'ютери, сполучені з сучасними інформаційними технологіями і не потребують спеціальної підготовки при їх використанні. Вони забезпечують вбудовування тексту в зображення, зображення у зображення, тексту в аудіосигнал і т. п. У сучасних телекомунікаційних мережах типу Internet передаються дуже великі потоки мультимедійних повідомлень, які потенціально можуть бути використані для приховування інформації. Однією з найбільш актуальних і складних проблем цифрової стеганографії є виявлення факту такого приховування. У реальних умовах найбільш типовим видом атаки зловмисника є атака тільки зі стега, оскільки істинний контейнер йому зазвичай невідомий. У цих умовах виявлення прихованого повідомлення можливо на основі виявлення порушень залежностей, притаманних природним контейнерам [50].

Практичний стеганоаналіз цифрових стеганосистем є дуже молодою наукою, однак у його арсеналі вже є ряд методів, які дозволяють з високою ймовірністю виявляти факт наявності стеганоканалів, утворених деякими запропонованими до теперішнього часу стеганосистемами. Серед методів практичного стеганоаналізу розглянемо візуальну атаку і ряд статистичних атак. Ці атаки спочатку були запропоновані для виявлення фактів упровадження приховуваної інформації в молодші розряди елементів контейнера, які прийнято називати найменш значущими бітами (НЗБ).

5.9. Візуальна атака на стеганосистеми

Розглянемо принцип побудови візуальної атаки, що дозволяє виявити факт наявності приховуваного повідомлення, вкладеного у зображення-контейнер [50]. Нехай стеганосистема побудована таким чином, що НЗБ елементів зображення заміняються на біти приховуваного повідомлення. Наприклад, у системі EzStego молодший бітів колірної компоненти кожного піксела, починаючи від початку зображення, послідовно замінюється відповідним бітом приховуваного повідомлення. В інших стеганосистемах біти впроваджуваного повідомлення заміщають молодші біти яскравісної компоненти кожного піксела зображення. Раніше вважалося, що НЗБ яскравісної або колірної компонент пікселів зображення, так само як і молодші біти відліків мовних або аудіосигналів незалежні між собою, а також незалежні від інших бітів елементів

розглянутих контейнерів. Однак насправді це не так. Молодші біти не є чисто випадковими. Між молодшими бітами сусідніх елементів природних контейнерів є істотні кореляційні зв'язки. Також виявлені залежності між НЗБ та рештою бітами елементів природних контейнерів.

На рис. 5.3 показано зображення млинів, ліворуч рисунок представляє порожній контейнер, справа в кожен НЗБ колірної компоненти пікселів послідовно бітів за бітом вкладено приховане повідомлення.

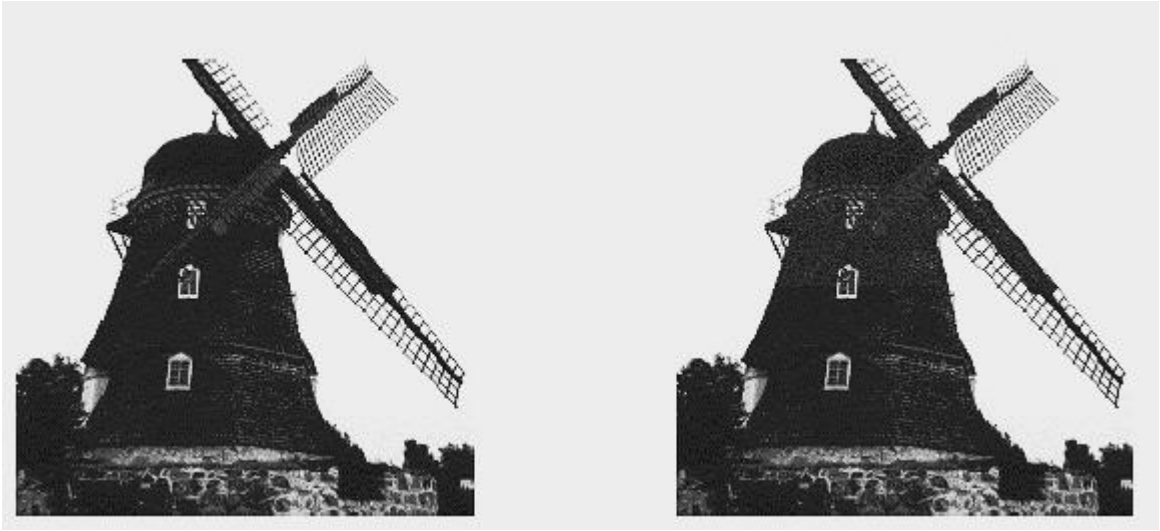


Рис. 5.3. **Зображення млинів: ліворуч – порожній контейнер, праворуч – з вкладеним повідомленням**

Різниця між контейнером і стегом візуально не виявляється. Але якщо зображення сформувати тільки з НЗБ пікселів стега, то можна легко побачити сліди вкладення. На рис. 5.3 ліворуч показано зображення, що складається з НЗБ порожнього контейнера. Видно, що характер зображення істотно, наполовину заповнене прихованим повідомленням контейнера. Видно, що верхня частина зображення, куди впроваджено повідомлення, становить випадковий сигнал.

У розглянутій стеганосистемі приховане повідомлення до вбудовування було зашифровано, тому кожен його бітів практично рівновірогідний і незалежний від сусідніх бітів, що дозволяє легко візуально виявити факт його вбудовування, зіставляючи зображення з молодших бітів стега і порожніх природних контейнерів, відповідно. У деяких стеганосистемах повідомлення до вбудовування стискуються. Це доцільно як для зменшення розміру приховано вбудованої інформації, так і для ускладнення його читання сторонніми особами.

Архіватори даних перетворюють повідомлення, що стискається, в послідовність бітів, досить близьку до випадкової. Чим вище ступінь стиснення, тим ближче послідовність на виході архіватора до випадкової, і тим простіше виявити факт існування стеганоканалу при візуальній атаці. Однак навіть якщо приховане повідомлення до вбудовування не шифрується і не стискається, то його імовірнісні характеристики не збігаються з імовірнісними характеристиками НЗБ використовуваних контейнерів, що знову ж таки можна виявити. Зауважимо, що відправник повідомлення може підібрати контейнер за законом розподілу, що збігається з законом розподілу конкретного вбудованого повідомлення.

У цьому випадку візуальна атака, як і статистичні атаки, неефективна. Але труднощі підбору необхідного контейнера можуть зробити таку стеганосистему непрактичною. У відомій програмі Steganos [66] вбудоване повідомлення будь-якої довжини здійснюється у всі НЗБ пікселів контейнера, тому виявляється візуальної атакою.

На рис. 5.4 наведена візуальна атака на EzStego, ліворуч – зображення з НЗБ порожнього контейнера, праворуч – наполовину заповненого.

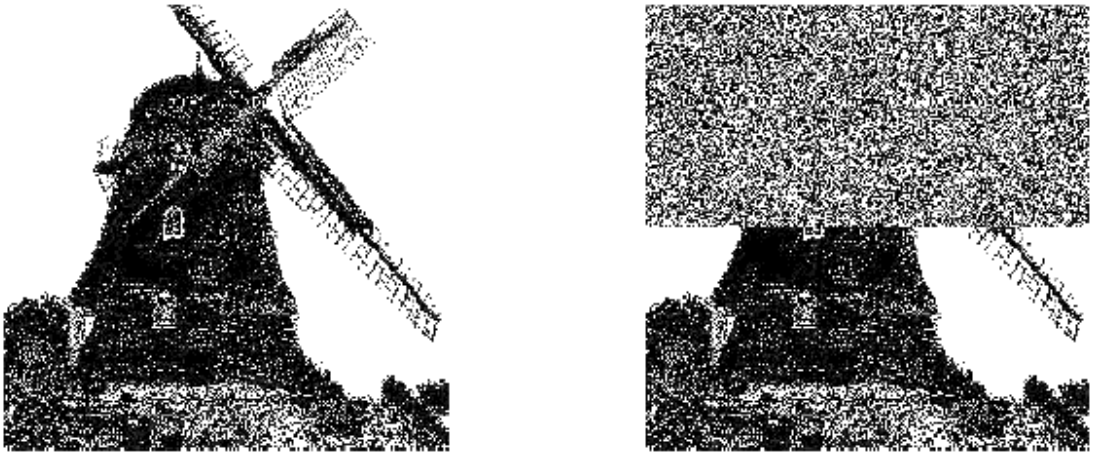


Рис. 5.4. **Візуальна атака на EzStego: ліворуч – зображення з НЗБ порожнього контейнера, праворуч – наполовину заповненого**

Візуальна атака цілком заснована на здатності зорової системи людини аналізувати зорові образи, а також виявляти істотні відмінності в порівнюваних зображеннях. Візуальна атака ефективна при повному заповненні контейнера, але в міру зменшення ступеня його заповнення оку людини все важче помітити сліди вкладення серед збережених елементів контейнера.

У ряді стеганографічних систем елементи приховуваного повідомлення вкладаються в молодші біти коефіцієнтів перетворення Фур’є

контейнера-зображення. Наприклад, 8 × 8 пікселів $f(x,y)$ блоку зображення спочатку перетворюються в 64 коефіцієнти дискретного косинусного перетворення (ДКП) за правилом

$$F(u,v) = \frac{1}{4}C(u)C(v)\left[\sum_{x=0}^{7}\sum_{y=0}^{7} f(x,y)\cdot \cos\frac{(2x+1)u\pi}{16}\cos\frac{(2y+1)v\pi}{16}\right],$$

де $C(u)$ і $C(v) = \frac{1}{\sqrt{2}}$ коли u і v дорівнюють нулю, і $C(u),\ C(v) = 1$ в інших випадках. Отримані коефіцієнти кванта з округленням до цілого:

$$F^{Q}(u,v) = Integer\_Round\left(\frac{F(u,v)}{Q(u,v)}\right),$$

де $Q(u,v)$ – таблиця квантування з 64 елементів.

Найменші значущі біти квантування ДКП коефіцієнтів, за винятком $F^{Q}(u,v) = 0$ і $F^{Q}(u,v) = 1$, у стеганосистемі є надлишковими бітами і замість них впроваджуються біти приховуваного повідомлення.

Проти таких методів приховування візуальні атаки малопридатні, оскільки зміна будь-якого коефіцієнта перетворення призводить до зміни множини пікселів зображення. Наприклад, у програмі Jsteg перетворення виконується над матрицею 16 × 16 пікселів контейнера. Отже, вкладення приховуваного повідомлення в молодші біти коефіцієнтів перетворення призведе до порівняно невеликих змін кожного з 256 пікселів, що візуально малопомітні.

Тому розглянемо другий клас практичних стеганоатак з метою виявлення прихованого каналу передачі інформації, заснованого на аналізі відмінностей між статистичними характеристиками природних контейнерів і сформованих з них стегів.

5.10. Статистичні атаки на стеганосистеми із зображеннями-контейнерами

Одним із найбільш перспективних підходів для виявлення факту існування прихованого каналу передачі інформації є підхід, який представляє введення у файл приховуваної інформації як порушення статистичних закономірностей природних контейнерів. При цьому підході аналізуються статистичні характеристики досліджуваної послідовності і

встановлюється, схожі вони на характеристики природних контейнерів (якщо так, то прихованої передачі інформації немає), або вони схожі на характеристики стега (якщо так, то виявлено факт існування прихованого каналу передачі інформації). Цей клас стеганоатак є імовірнісним, тобто вони не дають однозначної відповіді, а формують оцінки типу "дана досліджувана послідовність з імовірністю 90 % містить приховане повідомлення". Імовірнісний характер статистичних методів стеганоаналізу не є істотним недоліком, оскільки на практиці ці методи часто видають оцінки ймовірності існування стегаканалу, що відрізняються від одиниці або нуля на нескінченно малі величини.

Клас статистичних методів стеганоаналізу використовує множину статистичних характеристик, таких, як: оцінка ентропії, коефіцієнти кореляції, ймовірності появи і залежності між елементами послідовностей, умовні розподілу, помітні розподілів за критерієм Хі-квадрат та багато інших. Найпростіші тести оцінюють кореляційні залежності елементів контейнерів, в які можуть впроваджуватися приховані повідомлення. Для виявлення слідів каналу прихованої передачі інформації можна оцінити величину ентропії елементів контейнерів. Стеги, що містять вкладення приховуваних даних, мають більшу ентропію, ніж природні порожні контейнери. Для оцінки ентропії доцільно використовувати універсальний статистичний тест Маурера [62].

Розглянемо атаку на основі аналізу статистики Хі-квадрат. У програмі EzStego молодший бітів колірної компоненти кожного пікселя контейнера-зображення замінюється бітом приховуваного повідомлення. Досліджуємо закономірності в ймовірності появи значень колірної компоненти в природних контейнерах і сформованих програмою EzStego стегах. При заміні молодшого біта колірної компоненти на черговий бітів попередньо зашифрованого або стисненого повідомлення номер кольору пікселя стега або дорівнює номеру кольору пікселя контейнера, або змінюється на одиницю. В роботі [50] для пошуку слідів вкладення запропонований метод аналізу закономірностей у ймовірності появи сусідніх номерів кольору пікселів. Номер кольору, двійкове представлення якого закінчується нульовим бітом, назвемо лівим (L), а сусідній з ним номер кольору, двійкове представлення якого закінчується одиничним бітом, – правим (R). Нехай колірна гамма вихідного контейнера включає 8 кольорів. Отже, при вбудовуванні повідомлення в НЗБ колірної компоненти пікселів необхідно досліджувати статистичні

характеристики у 4 парах номерів кольору. На рис. 5.5 ліворуч показана одна з типових гістограм ймовірностей появи лівих і правих номерів кольору в природних контейнерах.

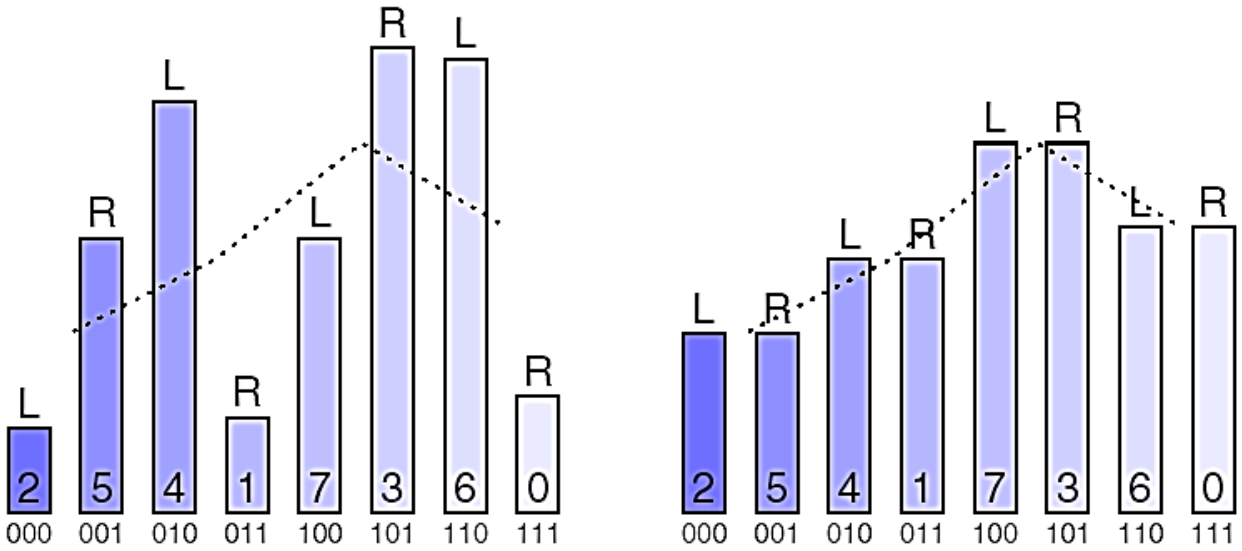


Рис. 5.5. **Гістограма частот появи лівих і правих номерів кольору: ліворуч – до вбудовування, праворуч – після**

Праворуч показана гістограма ймовірностей появи лівих і правих номерів кольору в стегі, сформованому із цього контейнера програмою EzStego. Видно, що ймовірність появи лівих і правих номерів кольору в природних контейнерах істотно різняться між собою у всіх парах, а в стегі ці ймовірності вирівнялися. Це є явним демаскуючою ознакою наявності приховуваної інформації. Зауважимо, що середнє значення ймовірностей для кожної пари в стегі не змінилося порівняно з контейнером (показано на рис. 5.5 пунктирною лінією).

При заміщенні бітами конфіденційного повідомлення у молодших бітів яскраві пікселів контейнера-зображення проявляються аналогічні статистичні відмінності.

Ступінь відмінності між ймовірнісними відділами елементів природних контейнерів та отриманих з них стегів може бути використана для оцінки ймовірності існування стеганоканалу. Цю ймовірність зручно визначити з використанням критерію згоди Хі-квадрат [72]. За критерієм Хі-квадрат порівнюється, наскільки розподіл досліджуваної послідовності близький до характерного для стеганограми розподілу. У досліджуваній послідовності підраховується, скільки разів n_i її елемент x_i прийняв розглянуті значення, де k всього елементів. Наприклад, у гістограми

лівих і правих номерів кольору в лівій частині рис. 5.5 номер кольору 000 з'явився 2 рази ($n_0^* = 2$), а номер 001 – 5 разів ($n_1^* = 5$). При вбудовуванні чергових бітів приховуваного повідомлення в НЗБ цієї пари номер кольору 000 повинен з'являтися в середньому n_0 разів:

$$n_0 = \frac{n_0^* + n_1^*}{2}.$$

Знаючи загальне число n появи всіх елементів досліджуваної послідовності, легко підрахувати очікувану ймовірність появи цих елементів у стегі за правилом: $p_i = n_i / n$. Відповідно, для досліджуваної послідовності ймовірності рівні: $p_i^* = n_i^* / n$.

Величина Хі-квадрат для порівнюваних розподілів досліджуваної послідовності та очікуваного розподілу стега дорівнює:

$$\chi^2 = \sum_{i=1}^{\nu} \frac{(n_i - np_i)^2}{np_i}$$

де ν – кількість ступенів свободи. Кількість ступенів свободи дорівнює числу k мінус кількість незалежних умов, накладених на ймовірності p_i^*. Наведемо одну з незалежних умов у вигляді:

$$\sum_{i=1}^{k} p_i^* = 1.$$

Ймовірність того, що два розподіли однакові, визначається як:

$$p = 1 - \int_0^{\chi^2} \frac{t^{(\nu-2)/2} e^{-t/2}}{2^{\nu/2} \Gamma(\nu/2)} dt,$$

де Γ – гамма-функція Ейлера.

Чим більше значення *p*, тим вища ймовірність вбудовування приховуваної інформації в досліджуваних послідовностях.

Розглянемо використання критерію Хі-квадрат для відшукання слідів стеганоканалу, утвореного з використанням програми EzStego. Нехай у контейнерне зображення "Млин", показане в лівій частині рис. 5.3, в НЗБ спектральних коефіцієнтів зображення, починаючи з його верхнього краю до його середини, послідовно впроваджено 3 600 байтів приховуваного повідомлення.

На рис. 5.6 показана ймовірність вбудовування приховуваної інформації залежно від розміру досліджуваної послідовності. Початок графіка отримано при аналізі першого фрагмента стега, що становить одну соту частину всього стега. Значення *p* склало 0,8826. Потім до аналізованого фрагмента була додана ще одна сота частина стега і т. д.. На другому кроці ймовірність склала 0,9808 і далі при аналізі стега не опускалася нижче 0,77. При переході до аналізу нижньої частини зображення, що не містить приховуваної інформації, величина p стрибком зменшилася до нуля.

У програмі Steganos вбудоване двійкове повідомлення будь-якої довжини доповнюється до довжини контейнера (до кількості пікселів зображення). Тому критерій Хі-квадрат при вбудовуванні як завгодно малого повідомлення з використанням Steganos дає ймовірність існування стеганоканалу, практично не відрізняється від одиниці.

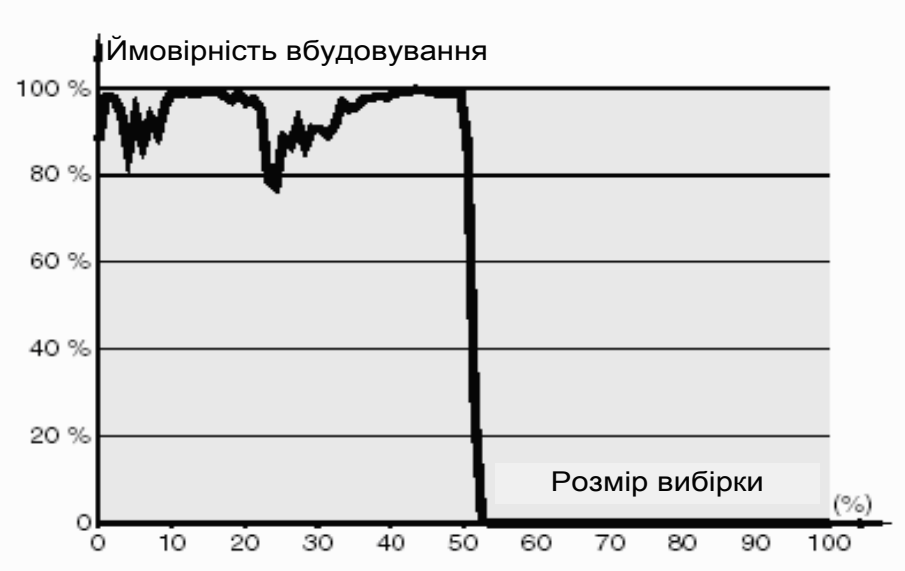


Рис. 5.6. **Ймовірність вбудовування за критерієм Хі-квадрат при аналізі EzStego**

У програмі S-Tools вбудоване повідомлення рівномірно розподіляється по всьому контейнеру. При повністю заповненому контейнері за критерієм Хі-квадрат впевнено виявляються сліди вкладення сторонньої інформації з зневажливо малою ймовірністю помилки, але при заповненому контейнері на третину і менші сліди стеганоканалу не виявляють.

Як і в EzStego, у програмі Jsteg приховуване повідомлення послідовно вбудовується в коефіцієнти перетворення контейнера. На рис. 5.7 показана ймовірність вбудовування за критерієм Хі-квадрат при аналізі

стега, сформованого з використанням Jsteg. Видно, що статистична атака успішно виявляє сліди приховуваної інформації в першій частині досліджуваної послідовності, яка містить приховуване повідомлення, і не дає помилкової тривоги у другій її частині, яка є порожнім контейнером.

Рис. 5.7. **Ймовірність вбудовування за критерієм Хі-квадрат при аналізі Jsteg**

Для стиснення зображень дуже часто використовується алгоритм JPEG. На рис. 5.8 показано, що ймовірність помилкового спрацьовування за критерієм Хі-квадрат при аналізі порожніх контейнерів, стиснених алгоритмом JPEG, не перевищує зневажливо малої величини 0,407 %.

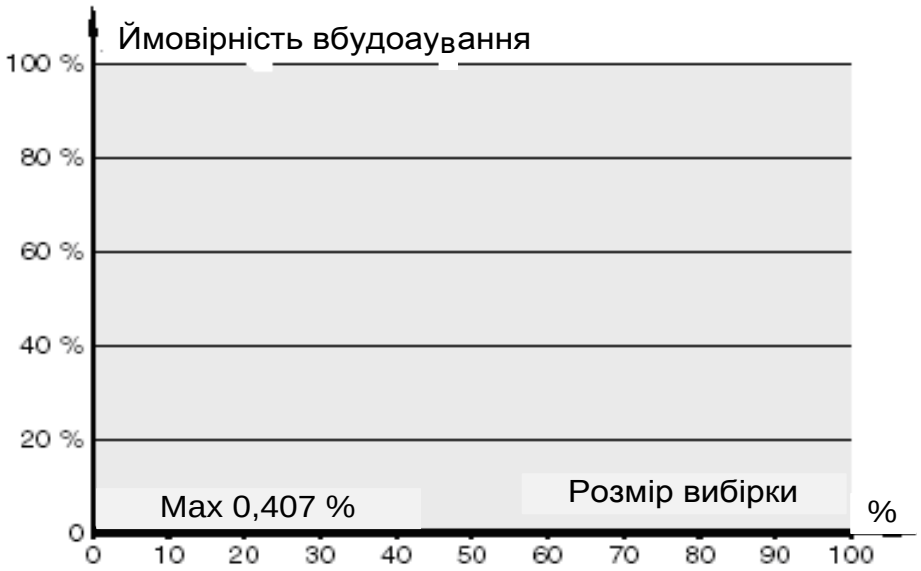


Рис. 5.8. **Ймовірність помилкового спрацьовування за критерієм Хі-квадрат при стисканні JPEG порожнього контейнера**

5.11. Статистичні атаки на стеганосистеми з аудіоконтейнерами

Розглянемо статистичні атаки, розроблені з метою виявлення прихованих каналів передачі інформації в аудіофайлах. У роботі [62] показано, що сліди приховування проявляються при аналізі таких статистичних характеристик мовлення і музики, як розподіл НЗБ відліків, умовні розподіли молодших і інших розрядів відліків, величини коефіцієнта кореляції між сусідніми відліками та ін.

Було досліджено понад 1 200 аудіофайлів, що записані на CD-дисках та становлять різні музичні та вокальні твори різних авторів. Показано, що для порожніх аудіоконтейнерів НЗБ та інші біти статистично взаємозалежні, причому на характер цієї залежності впливає рівень запису (усереднена амплітуда відліків аудіосигналу).

На рис. 5.9 показана отримана для аудіофайлів залежність статистики Хі-квадрат. За критерієм Хі-квадрат обчислювався ступінь відмінності між розподілом порожніх і заповнених контейнерів від характерного для стега бернуллієвського розподілу. До теперішнього часу відомі різні програмні засоби приховування інформації в аудіофайлах. Використовуючи статистику Хі-квадрат і коефіцієнт кореляції, в роботі [62] проведено стеганоаналіз програм Steganos (version 1.0a) і S-Tools (Steganography Tools for Windows, version 4.0), які приховують інформацію в найменш значущих бітах звукових відліків.

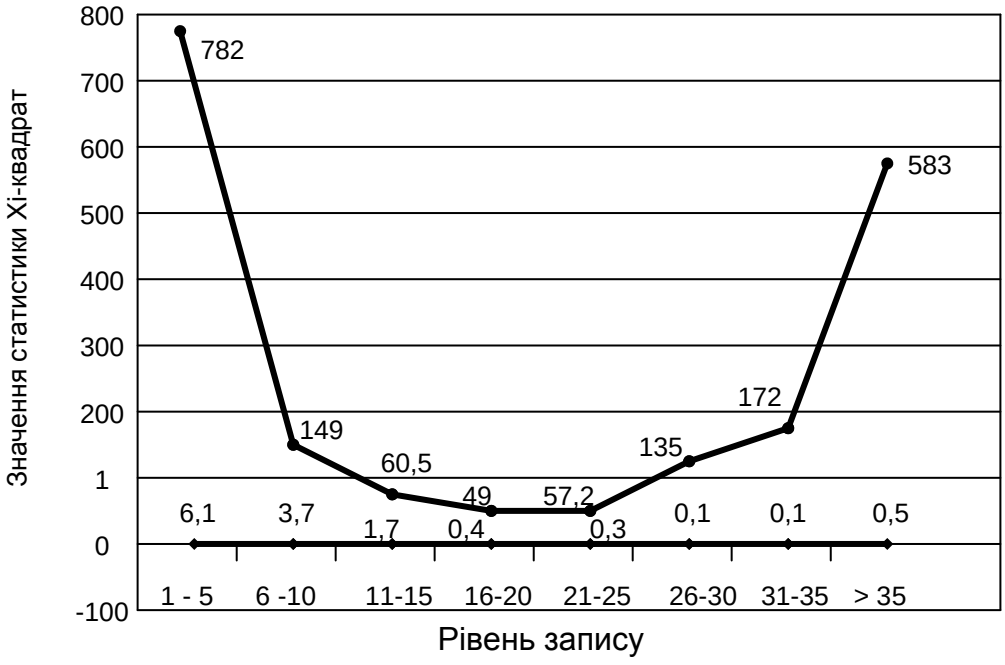


Рис. 5.9. **Залежність величини Хі-квадрат від амплітуди відліків аудіосигналу**

У якості вихідних контейнерів досліджувалися зчитані з CD-дисків 100 музичних фрагментів різних виконавців тривалістю звучання 15 с кожен (як зі стандартних музичних компакт-дисків, так і з дисків у форматі MP3). В якості приховуваного повідомлення використовувалася псевдовипадкова послідовність обсягом 83 Кбайти і побітно упроваджувалася в кожен НЗБ контейнера. За критерієм Хі-квадрат визначався ступінь відмінності розподілу НЗБ відліків досліджуваної послідовності від бернуллієвського розподілу. Результати статистичних обчислень для музичних контейнерів і сформованих з них повністю заповнених стегів наведені у вигляді гістограми на рис. 5.10а, б, в.

При цьому область значень статистики (вісь абсцис) розбита на непересічні і різні за розмірами інтервали. Висота стовпчика (вісь ординат) показує число значень статистики, що потрапили в заданий інтервал.

На рис. 5.10 наведена частота зустрічності значень статистики Хі-квадрат (а – для S-Tools, б – для Steganos) та коефіцієнта кореляції (в – для S-Tools). Праві стовпчики відповідають порожнім контейнерам, а ліві – заповненим стегам. Для стегів величина Хі-квадрат дорівнює одиницям, а для порожніх контейнерів – десяткам і сотням.

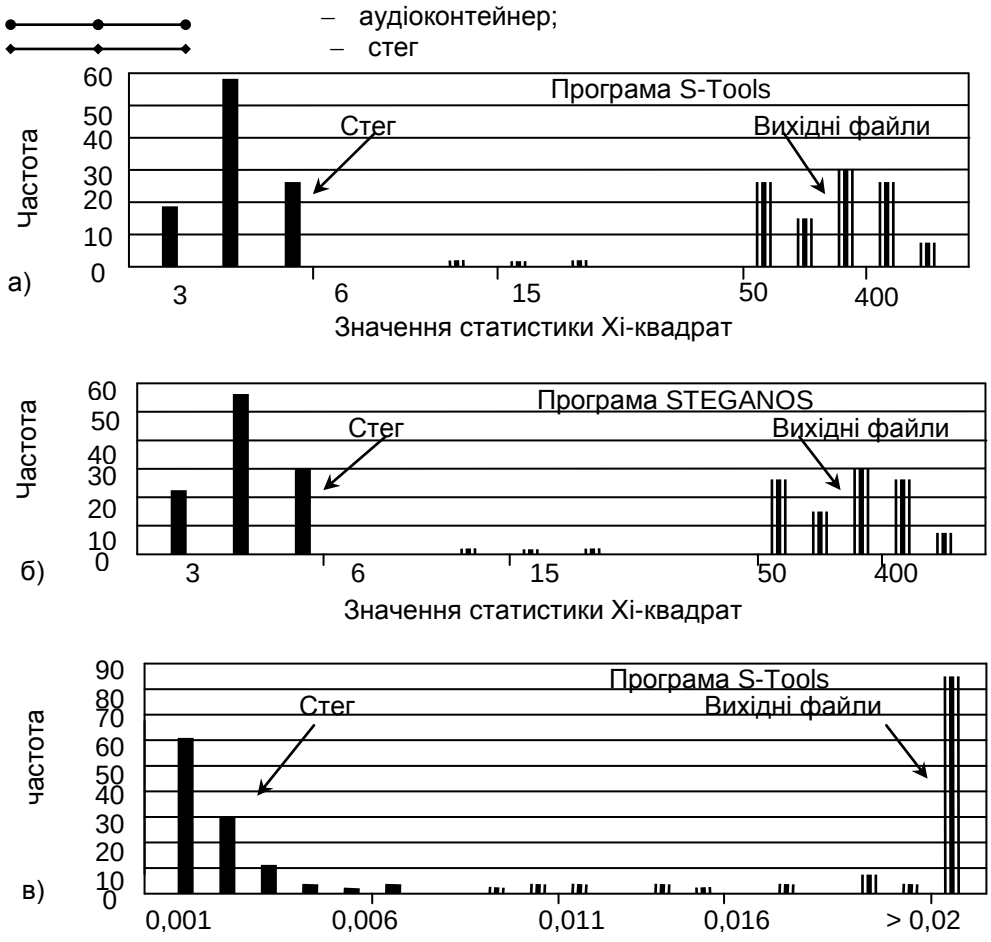


Рис. 5.10. **Статистичні розбіжності стега і порожніх аудіоконтейнерів: а, б – за критерієм Хі-квадрат; в – за модулем коефіцієнта кореляції**

Після вбудовування середнє значення коефіцієнта кореляції сусідніх відліків зменшилося в десятки разів.

Зазначимо, що діапазони значень статистики Хі-квадрат, отримані до і після утворення стеганоканалу, так само як діапазони значень коефіцієнтів кореляції, не перетинаються. Ці ознаки дозволяють при використанні статистичних атак з великою ймовірністю відділити порожні аудіоконтейнери від заповнених стегів.

5.12. Напрями підвищення захищеності стеганосистем від статистичних атак

Таким чином, різні стеганосистеми, що використовують принцип заміни молодших бітів елементів контейнерів на біти вбудованого повідомлення, виявилися нестійкими до статистичних атак. Підвищити їх стійкість можна різними способами, наприклад, переходом до операцій вбудовування виду визначається складанням елементів контейнера з елементами вбудованого повідомлення. Подібні операції не зберігають баланс ймовірностей появи відповідних елементів контейнера і стега, і тому мають більш високу стійкість до аналізу їх статистик.

Очевидним способом є зменшення ступеня заповнення контейнера бітами приховуваного повідомлення, тобто зменшення пропускної здатності стеганоканалу в обмін на підвищення його захищеності. Запропоновані в роботі [50] статистичні атаки на основі критерію Хі-квадрат у більшості випадків не здатні виявити стеганоканал при заповненні контейнера на 50 % і менше, особливо якщо запроваджені повідомлення розосереджені по контейнеру. Ці атаки завжди стартують від початку досліджуваної послідовності і використовуються рівномірно збільшуючи вікно аналізу. Вони виявляють існування стегаканалу, якщо статистичні характеристики деформуються безперервно від початку контейнера. Проміжні множини в контейнері, які не мають спотворень, можуть викликати неправильний результат тесту. Тому в роботі [62] запропонована удосконалена статистична атака, названа автором розширеним тестом Хі-квадрат. Тест використовує фіксований розмір вікон аналізу, переміщуваного вздовж досліджуваної послідовності. Така атака здійснює локальний пошук і дозволяє вказати на місце вкладення приховуваного повідомлення. У цій же роботі пропонується спосіб підвищення захищеності від статистичних атак стеганосистем із вкладенням

приховуваного повідомлення в НЗБ контейнера. Процес вбудовування прихованої інформації в контейнер розділений на 3 етапи:

1) визначення надлишкових бітів, які можна змінювати без шкоди для контейнера;

2) вибір НЗБ, в які буде вбудовуватися приховувана інформація;

3) корекція статистичних змін у сформованому стегі.

На першому етапі оцінюється кількість НЗБ контейнерів, які можна замінити на біти приховуваного повідомлення без втрати якості контейнера типу зображення. Реально для вбудовування можна використовувати не більше половини виявлених бітів. Якщо знайдених надлишкових бітів не достатньо, треба поміняти контейнер. Потім за допомогою секретного ключа визначаються рівновірогідно розподілені в межах контейнера НЗБ, замінюються на біти прихованої інформації. Потім сформований стег оцінюються статистичними тестами і при виявленні відхилень від статистичних характеристик природних контейнерів залишені надлишкові біти використовуються для виправлення цих відхилень. Простим методом корекції є збереження взаємної кореляції і величини ентропії, що обчислюється за тестом Маурера. Дійсно, якщо деякий молодший бітів при вбудовуванні змінюється від 0 до 1, то доцільно змінити сусідній НЗБ від 1 до 0 і т. п. Хоча цей метод дозволяє зберегти величину ентропії і коефіцієнт кореляції при вкладенні в контейнер приховуваного повідомлення, він має статистичні слабкості макроскопічного характеру. Це виражається в спотворенні гістограми коефіцієнтів ДКП, аналогічно тому, як це показано на рис. 5.10. Якщо лівий коефіцієнт змінився, то щоб гістограма стега не відрізнялася від гістограми вихідного контейнера, необхідно змінити правий коефіцієнт на ту ж величину.

Корегуючі перетворення повинні задовольняти вимогам:

1) для будь-якого фрагмента зображення розподіл коефіцієнтів ДКП стега повинен бути аналогічним їх розподілу в порожньому контейнері;

2) кількість виправлень, необхідних для корекції статистичних характеристик, має бути малим.

У роботі [62] наведені результати дослідження алгоритму корекції при вбудовуванні повідомлення у контейнерні зображення розміром 640 × 480 пікселів. Середня кількість коефіцієнтів ДКП, які можна використовувати для вбудовування, так само 46 000 і змінювалася від 30 000 до 97 000. До вбудовування ймовірність збігу сусідніх надлишкових бітів дорівнює 63,8 % із середнім квадратичним відхиленням ± 3,4 % з множини зображень. Довжина стисненого приховуваного

повідомлення дорівнює 14 700 бітів. Корегуючі перетворення привели до 2967 ± 434 додаткових змін у надлишкових бітах. Це склало приблизно 20 % від розміру приховуваного повідомлення. Середня кількість спотворень, які не вдалося змінити, склало 186 ... 400.

У табл. 5.1 показані результати статистичних тестів для досліджуваного алгоритму. Видно, що в стегі без корекції помітно зменшився коефіцієнт кореляції між надлишковими бітами і збільшилася їх ентропія. Корекція дозволяє зробити вбудовування приховуваних повідомлень статистично невиявленим.

Таблиця 5.1

Результати статистичних тестів

Досліджувана послідовність	Коефіцієнт кореляції	Універсальний тест Маурера
Початковий контейнер	63,77 % ± 3,37 %	6,704 ± 0,253
Стег без корекції	59,10 % ± 3,19 %	6,976 ± 0,168
Стег з корекцією	62,91 % ± 3,36 %	6,775 ± 0,231

Таким чином, якщо застосувати коригуючі перетворення до стега, то використані методи статистичного стеганоаналізу не здатні виявити факт існування стеганоканалу. Однак заради справедливості треба зазначити, що можуть бути побудовані інші статистичні атаки, для нейтралізації яких потрібно додатково використовувати надлишкові біти, що ще більше зменшить швидкість передачі приховуваної інформації.

Вдосконалення стеганосистем у загальному випадку може бути описано деякими ітеративними процесами. Стеганосистеми розробляються і пропонуються авторами до використання. Вони досліджуються відомими методами стеганоаналізу, при необхідності для них розробляються нові методи аналізу, і так до тих пір, поки не вдається їх зламати.

Потім з урахуванням виявлених слабкостей принципи побудови стеганосистем удосконалюються, але одночасно розвиваються і стеганоатаки. Цей ітеративний процес триває, поки не вдається довести, що при поточному рівні розвитку стеганоаналізу дана стеганосистема є практично стійкою. Такий процес склався для аналізу та синтезу криптосистем, і очевидно, що він правильний і для стеганосистем. Однак треба враховувати, по-перше, що порівняно з криптосистемами в стеганосистемах є додатковий параметр – контейнер, а по-друге, практична стійкість стеганосистем може мати значно більшу кількість тлумачень.

5.13. Теоретико-складнісний підхід до оцінки стійкості стеганографічних систем

Інформаційно-теоретичні моделі стійкості стеганографічних систем мають істотні недоліки. Вперше на це було звернуто увагу в роботі [62].

Як зазначено в цій роботі, інформаційно-теоретичні методи, що успішно застосовуються для аналізу криптосистем, погано підходять для аналізу стеганосистем. Причина цього в тому, що процедура виявлення прихованого повідомлення не може бути змодельована як безперервний процес. Насправді, зловмисник може отримати лише два результати аналізу підозрілого каналу зв'язку: або він виявить факт присутності стеганосистеми, або ні. Таким чином, маємо справу з переривчастим процесом, до якого незастосовні методи теорії інформації. У криптографії не так, там зловмисник може отримувати часткове знання про відкриті повідомлення (або ключі), і тим не менш система буде практично стійкою. Стеганосистема ж зобов'язана бути зовсім стійкою за К. Шенноном.

На рис. 5.11 на якісному рівні показана різниця між криптосистемами і стеганосистемами (по осі ординат відкладений ступінь таємності систем, по осі абсцис – обчислювальні ресурси зловмисника). Усвідомлення факту малопридатних інформаційно-теоретичних моделей для аналізу стеганосистем спричинило появу теоретико-складнісних підходів для оцінки їх стійкості. У цій роботі по-новому розглянуто поняття стійкості стеганосистем і побудована конструктивна модель стійкої стеганосистеми у вигляді ймовірносно поліноміальної за часом гри між зловмисником і приховувачем інформації.

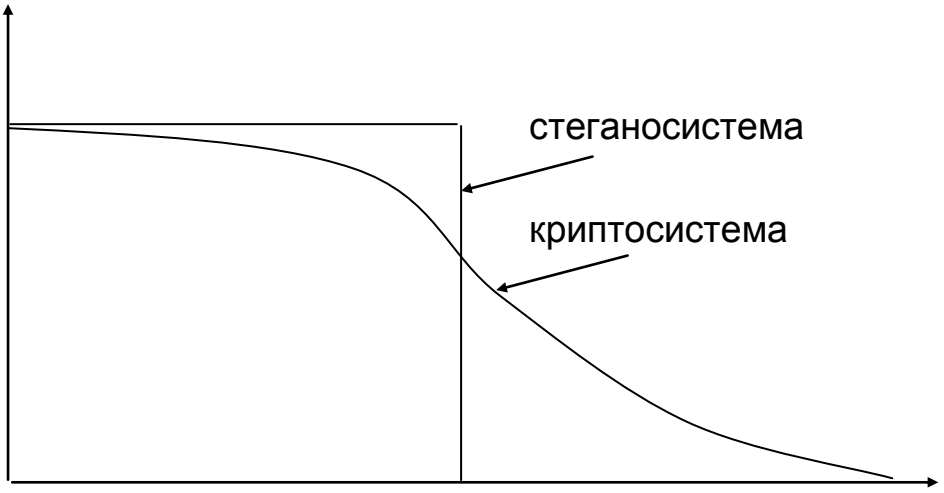


Рис. 5.11. **Порівняння криптосистем і стеганосистем**

До основних недоліків інформаційно-теоретичних моделей стеганосистем можна віднести такі:

1. На практиці неможливо реалізувати абсолютно стійку стеганосистему. Можна показати, що реалізація такої стеганосистеми зводиться до одноразового блокноту (так званого шифру Вернама). Таким чином, інформаційно-теоретичні моделі стеганосистем неконструктивні.

2. Розподіл ймовірностей контейнерів на практиці невідомий, чи відомий з точністю до деякої дуже і дуже приблизної моделі.

3. Використані контейнери аж ніяк не є реалізацією випадкового процесу, а, частіше за все, оцифрованими образами реальних фізичних об'єктів.

4. Цілком реалістично було б припустити, що зловмисник має доступ лише до обмежених обчислювальних ресурсів. Як і в криптографії, досить вимагати, щоб стеганосистема витримувала б всі поліноміальні тести з її виявлення. Цей момент також не враховують інформаційно-теоретичні моделі.

Розглянемо модель стеганосистеми, запропоновану в роботі [62]. Припустимо, що є множина можливих контейнерів C, елементи якої $c \in C$ породжуються деяким поліноміальним алгоритмом. Вбудоване повідомлення $m \in M$ вибирається з множини можливих повідомлень $M = \{0,1\}^l$. Стеганосистема визначається трійкою $\langle G, E, D \rangle$ поліноміальних алгоритмів.

Алгоритмом є процес генерації ключа, який у відповідь на вхідний рядок з одиниць породжує псевдовипадковий стеганоключ $k \in \{0,1\}$. Відповідно до принципу Керхгофа стійкість залежить від ключа, а його довжина є параметром секретності стеганосистеми. Алгоритм виконує впровадження інформації, формуючи на основі $c \in C$, $m \in M$, і k стег $s \in C$. Алгоритм D витягує з s з використанням ключа k повідомлення m'. У випадку, якщо контейнер s дійсно містив убудоване повідомлення, то $m' = m$. Для визначення наявності стеганосистеми зловмисник повинен вирішити таку задачу:

На основі контейнера $s \in C$ визначити, чи існує ключ $k \in \{0,1\}$, породжуваний G і повідомлення $m \in M$ такі, що $D(s,k) = m$.

Цікаво відзначити, що якщо на структуру прихованого повідомлення не накладається ніяких обмежень, то для багатьох стеганосистем це завдання неможливе. Насправді, будь-яка комбінація бітів може бути

вкладенням, і навіть якщо зловмисник якимось чином і запідозрить наявність прихованого зв'язку, все одно йому неможливо буде довести це третій стороні. Тому в роботі [62] на структуру прихованого повідомлення накладається обмеження: воно повинно мати який семантичний зміст.

Далі вважається, що у зловмисника є стеганосистема у вигляді "чорного ящика", тобто він має можливість породжувати стеги з обраних ним контейнерів і прихованих повідомлень, не знаючи при цьому ключа. Для цієї мети у нього є два оракули: один для генерації порожніх контейнерів (стеганографічний оракул), інший – для отримання з них стега, тобто імітації алгоритму впровадження (оракул оцінки). Оскільки обидва оракули імовірнісні, то у випадку вибору першого оракула кілька разів поспіль одного й того самого контейнера стеги будуть виходити різними. Це допомагає зловмиснику з'ясовувати структуру алгоритму впровадження, вибравши в якості контейнера, наприклад, однотонове зображення.

Атака (гра) полягає в наступному. Зловмисник має кількаразову можливість генерувати контейнери і відповідні їм стеги, намагаючись з'ясувати структуру стеганоалгоритму. При цьому є те обмеження, що вся процедура має бути поліноміальною за довжиною ключа та розміром контейнера. Після того, як він закінчив роботу, йому пред'являються два випадково обраних контейнери: один – порожній, інший – заповнений. *Стеганосистема* називається *умовно стійкою*, якщо у зловмисника немає можливості правильного визначення стега з імовірністю, що незначно відрізняється від 1/2. У роботі [62] дано визначення поняття "незначно відрізняється" і наведено математичний опис вербально викладеної вище моделі. Умовно стійка стеганосистема зберігає цю властивість для всіх можливих ключів і всіх можливих контейнерів.

Ясно, що поняття умовно стійкої стеганосистеми більш слабке, ніж поняття стеганосистеми, стійкої з інформаційно-теоретичної точки зору, і включає її як окремий випадок. Безумовно стійка стеганосистема у наведеній вище моделі виходить у випадку, якщо зняти обмеження поліноміальності під час гри.

Яким чином побудувати умовно стійку стеганосистему? Одна з можливостей, що широко використовується і в криптографії, полягає у взятті за основу якої-небудь важкої в обчислювальному сенсі математичної завдання, наприклад, звернення односторонньої функції (розкладання на множники, дискретне логарифмування і т. д.). Тоді залишиться

показати зв'язок між неможливістю вирішення цієї завдання і неможливістю розтину стеганосистеми – і умовно стійка стеганосистема побудована. З криптографії відомо, що, на жаль, питання побудови доказу односторонньої функції невирішене. У роботі [62] показано, як можна побудувати стеганосистему на основі відомого криптоалгоритму RSA.

5.14. Імітостійкість системи передачі приховуваних повідомлень

Раніше була досліджена стійкість стеганосистем до спроб пасивного зловмисника встановлення факту приховування переданих повідомлень. Додатково до вимог прихованості зв'язку можуть висуватися вимоги у вилученні нав'язування в стеганоканалі помилкових повідомлень активним зловмисником. Наприклад, у роботі Г. Сіммонс описане так зване завдання ув'язнених [61]. У цьому завданні заарештовані Аліса і Боб намагаються по прихованому канальному зв'язку домовитися про втечу. Тюремник Віллі намагається не тільки виявити факт обміну інформацією, але і від імені Аліси нав'язати Бобу неправдиву інформацію. Тому розглянемо особливості побудови стеганосистем з можливістю автентифікації переданих повідомлень, можливі атаки зловмисника і визначимо оцінки імітостійкості стеганосистем.

Формально опишемо побудову стеганосистеми з автентифікацією приховано переданих повідомлень. Нехай стеганосистема використовує секретний ключ, який приймає значення $K_1, K_2, \ldots, K_n$. Множина контейнерів C розбивається на n підмножин $C_1, C_2, \ldots, C_n$, кожне з яких описується своїм імовірнісним розподілом $P_{C_1}, P_{C_2}, \ldots, P_{C_n}$. Поставимо підмножини C_i контейнерів у відповідність секретним ключам $K_i, i = 1, 2, \ldots, n$. При діючому ключі автентифікації K_i повідомлення, доставлене за допомогою прихованого зв'язку, вважається одержувачем справжнім, якщо воно вкладено в контейнер, що належить підмножині з розподілом P_{C_i}. Якщо при діючому ключі K_i заповнений контейнер не належить підмножині C_i, то витягнення з нього повідомлення визнається одержувачем неправдивим. Таким чином, при діючому ключі уся множина контейнерів розділена на допустимі, в яких справжність вкладених у них повідомлень визнається одержувачем, і недопустимі, які не можуть бути обрані для передачі відправником приховуваних повідомлень. Отже, отримання таких контейнерів з вкладеними повідомленнями означає, що вони нав'язані зловмисником.

Якщо прийнятий стег S має розподіл P_s, співпадає з розподілом P_{C_i} множини припустимих контейнерів при діючому ключі K_i, то функція перевірки автентичності приховуваних в них повідомлень $X(S,K_i)$ приймає одиничне значення і отримане повідомлення визнається справжнім, а якщо розподіли не збігаються, то функція приймає нульове значення і повідомлення відкидається як імітонав'язане:

$$X(S,K_i)=\begin{cases}1, & \text{якщо} \quad P_S \in P_{C_i},\\ 0, & \text{якщо} \quad P_S \notin P_{C_i}.\end{cases}$$

Функція перевірки автентичності при побудові стеганосистеми з автентифікацією повідомлень може бути задана аналітично, графічно або у вигляді таблиці. При аналітичному завданні кожне значення ключа ставиться у відповідність своїй підмножині допустимих контейнерів. Ці підмножини відрізняються один від одного законами розподілу або їх параметрами. Наприклад, використовуються різні розподіли ймовірностей безперервних контейнерів (нормальний, Райса, Накагамі та ін.). Або підмножини контейнерів-зображень відрізняються спектральними характеристиками. Наприклад, у кожній підмножині енергія спектра зображень зосереджена у своєму діапазоні частот. Відомо, що зображення можна поділити на високочастотні, основна енергія спектра яких належить верхній смузі частот, і на низькочастотні. Також можна розділити контейнери-зображення на підмножини за типом сюжету: пейзаж, портрет, натюрморт і т. п. Хоча при сюжетному розбитті важко математично строго задати функцію $\boldsymbol{X}(S,K_i)$ в термінах законів розподілу, на практиці завдання такої функції не становить труднощів. Множина всіх контейнерів розбивається на n непересічних підмножин контейнерів $C_1, C_2, \ldots, C_n$. Наприклад, контейнери можуть бути розбиті на підмножини їх перетином. При діючому ключі K_i відправник вибирає підмножину контейнерів C_i. Приховуване повідомлення M_j, де $j=\overline{1,k}$, вбудовується в контейнер цієї підмножини, утворюючи стеганограму $S_{i,j}$. Одержувач стеганограми перевіряє її відповідність чинному ключу. Він переконується, що отримана стеганограма допустима при ключі K_j, якщо виконується $\boldsymbol{X}(S_{i,j},K_j)=1$. Ця рівність виконується, якщо стеганограма $S_{i,j}$ належить підмножині контейнерів C_j. Отже, вилучення

з цієї стеганограми повідомлення $\hat{M}_j$ істинно. Але якщо прийнята стеганограма не належить допустимій підмножині контейнерів, то функція перевірки приймає нульове значення, і прийняте повідомлення M_j^* відхиляється як помилкове. Графічний опис функції перевірки автентичності наведено на рис. 5.12. Нехай по стеганоканалу можуть передаватися k різних повідомлень: $M_1, M_2, \ldots, M_к$. Множина ключів стеганосистеми складається з n ключів, з яких рівновірогідно і випадково вибирається діючий ключ.

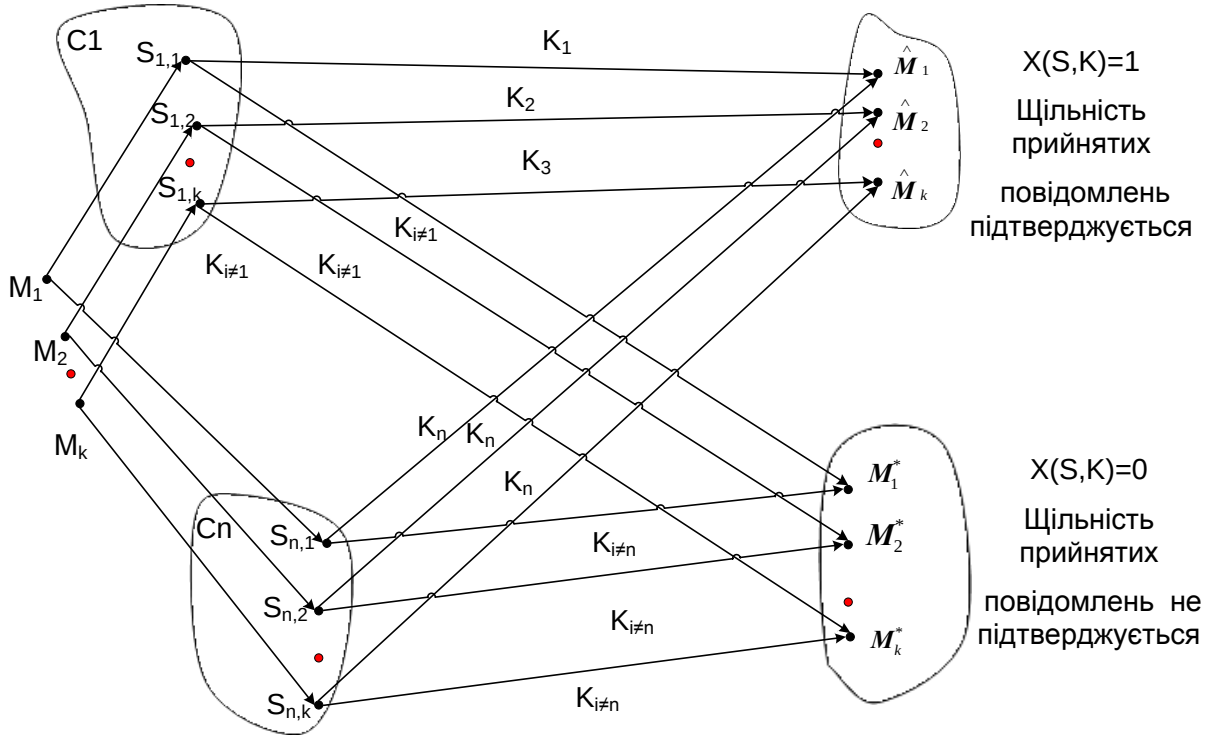


Рис. 5.12. **Графічний опис функції перевірки автентичності приховуваних повідомлень**

З рис. 5.12 легко помітити, що підмножини контейнерів мають однакові розміри. Якщо приховувані повідомлення рівновірогідні і рівновірогідно вибирається ключова інформація, то для зловмисника, не знаючого діючий ключ, множина повідомлень, справжність яких підтверджується під час перевірки, у n – 1 разів менше множини повідомлень, що відкидаються при перевірці як помилкові.

Розглянемо можливі атаки зловмисника на справжність приховуваних повідомлень та оцінки імітостійкості стеганосистем при цих атаках. З криптографії відомо, що активний зловмисник може виконати атаку імітації або атаку заміни [3]. При атаці імітації, що інакше називається

імітонав'язуванням у порожньому каналі, зловмисник, не чекаючи перехоплення завіреного повідомлення, від імені відправника формує помилкове повідомлення.

Позначимо ймовірність успіху зловмисника в атаці імітації через P_i.

З рис. 5.12 очевидно, що для порушників, не знаючих чинного ключа і нав'язуючих будь-яке повідомлення з множини $M_1, M_2, \ldots, M_K$, ймовірність успіху не може бути меншою, ніж кількість всіх повідомлень, поділене на кількість всіх стеганограм $S_{i,j}$ при $i = \overline{1,n}$ і $j = \overline{1,k}$.

$$P_i \geq \frac{|M|}{|S|} = \frac{k}{n \cdot k}. \qquad (5.23)$$

Г. Сіммонс для систем автентифікації визначає, що вираз (5.23) виконується з рівністю при задоволенні двох умов:

1. Атака імітації оптимальна, тобто має однакову ймовірність успіху зловмисника при рівновірогідному випадковому виборі їм будь-якої нав'язуваної стеганограми.

2. Для кожної стеганограми $S_{i,j}$ ймовірність її формування відправником однакова при всіх ключах автентифікації, для яких виконується $\boldsymbol{X}(S_{i,j}, K_j) = 1$.

Якщо ці умови виконуються, то при заданих розмірах множини приховуваних повідомлень і множини стеганограм ймовірність обману є мінімальною. Дотримуючись Сіммонса, *стеганосистему з автентифікації приховуваних повідомлень* можна назвати *досконалою* щодо атаки імітації, якщо вона задовольняє рівності у виразі (5.23). З виразу (5.24) випливає, як що мала ймовірність обману, то є висока імітозахищеність стеганоканалу забезпечується за $|S| >> |M|$. Відзначимо, що ні на яких принципах побудови стеганосистеми величина не може бути отримана меншою, ніж у виразі (5.23).

При другій стратегії імітонав'язування в стеганоканалі, що називається *атакою заміни першого порядку*, зловмисник, перехопивши стеганограму від законного відправника, підмінює її на помилкову. Атака заміни вважається успішною, якщо нав'язаний стег декодується одержувачем у будь-яке допустиме для даної стеганосистеми повідомлення, причому неправдиве повідомлення не повинно співпадати зі справжнім повідомленням законного відправника. Позначимо ймовірність обману при

атаці заміни через P_d. Якщо зловмисник перехоплений стег, що містить деяке невідоме йому повідомлення, замінив на будь-який інший стег, то очевидно (див. рис. 5.12), що при непересічних підмножинах $C_1, C_2, \ldots, C_n$, ні з якого стега добуті повідомлення при діючому ключі не будуть одночасно визнані одержувачем справжнім і збігатися з істинними, які передаються законним відправником повідомлень. Отже, у зловмисника є шанси нав'язати одне з залишившихся $k-1$ повідомлень, використовуючи один з $n \cdot k - 1$ стегів. Таким чином, імовірність успішного нав'язування в атаці заміни першого порядку не перевищує:

$$P_d \le \frac{|M|-1}{|S|-1} = \frac{k-1}{n \cdot k - 1}. \qquad (5.24)$$

Відзначимо, що як і при атаці імітації, висока імітозахищеність стеганоканалу при атаці заміни першого порядку забезпечується за $|S| >> |M|$. Перераховані раніше умови є необхідними, але вже недостатніми умовами виконання виразу (5.24) зі знаком рівності. Визначимо стеганосистему з автентифікації приховуваних повідомлень досконалої щодо атаки заміни першого порядку, якщо вона задовольняє рівності у виразі (5.24).

Пояснимо на простому прикладі стратегії імітонав'язування та оцінки захищеності від обману для стеганосистеми наступного вигляду. Задамо табличний опис функції перевірки автентичності, наведеної в табл. 5.2.

Таблиця 5.2

Приклад стратегії імітонав'язування

Приховуване повідомлення	Номер умовного сигналу	Приховуване повідомлення	Номер умовного сигналу	Чинний ключ автентифікації
Втеча сьогодні	2	Втечу відмінено	6	K_1
Сьогодні втечу	5	Відмінено втечу	3	K_2
Втечу назначено на сьогодні	1	Втечу сьогодні відмінено	4	K_3

Нехай двоє ув'язнених, Аліса і Боб, домовилися про наступну побудову прихованого каналу передачі з автентифікації повідомлень. Для цього вони заздалегідь (до арешту) домовилися про відповідність

приховуваних повідомлень умовним сигналам. Вони також встановили, що при діючому ключі частина повідомлень є допустимими (Аліса їх може передавати), а решта повідомлень – недопустимими (Аліса їх передавати не буде).

У табл. 5.2 зазначено, які повідомлення є допустимими при діючому ключі автентифікації (K_1,K_2 або K_3).

Нехай Аліса і Боб організували передачу приховуваних повідомлень таким чином. Щоранку Боба виводять на прогулянку і він спостерігає вікно камери Аліси. Для прихованої передачі повідомлень Аліса виставляє у вікні своєї камери горщики з геранню, кількість яких дорівнює номерам умовного сигналу згідно з табл. 5.2. Якщо на цей день діє ключ автентифікації, то повідомленню "втеча сьогодні" відповідає 2 горщика з квітами, а повідомленню "втечу відмінено" – 6 горщиків.

Розглянемо можливі стратегії введення неправдивої інформації в цей канал прихованого зв'язку тюремником Віллі. Перший варіант дій Віллі реалізується атакою імітації. Тюремник припускає, що за допомогою кольорів передається прихована інформація. Не чекаючи дій Аліси, він виставляє у вікно її камери деяку кількість горщиків з геранню. При 2 або 6 предметах Боб, отримавши неправдиве повідомлення, вважає, що воно дійсно передано Алісою, оскільки ці повідомлення допустимі при діючому ключі K_1.

У цих випадках порушнику вдалося нав'язати помилкове повідомлення, хоча Віллі не знає, яке саме. Але якщо Віллі вибере для імітонав'язування умовні сигнали 1, 3, 4 або 5, то Боб однозначно визначить, що прийняте повідомлення інспіровано зловмисником.

Таким чином, при рівновірогідному виборі помилкового повідомлення ймовірність успіху Віллі в атаці імітації дорівнює $P_i = 1/3$.

Розглянемо другу стратегію імітонав'язування – атаку заміни першого порядку. Віллі зауважує, що Аліса виставила у вікно, наприклад, 2 горщика з квітами. Тюремник припускає, що це таємно передається повідомлення, і змінює умовний сигнал на інший. Якщо Віллі нав'язує умовний сигнал 1, 3, 4 або 5, то Боб визначить, що отримане повідомлення є хибним. Але якщо Віллі використовує умовний сигнал номер 6, то імітовведення виявиться успішним і Боб отримає замість сигналу "втеча сьогодні" сигнал "втечу відмінено" з усіма наслідками, що випливають для нього. Таким чином, у даній атаці замінена ймовірність

успішного нав’язування помилкового повідомлення за рівновірогідний їх вибір дорівнює $P_d = 1/5$. Виявилося, що $P_d < P_i$, але слід урахувати, що успіх зловмисника в атаці заміни завдає більших збитків порівняно з атакою імітації, оскільки при успіху в атаці заміни зловмиснику вдається нав’язати діаметрально протилежне повідомлення. Зауважимо, що на відміну від цього в атаці імітації нав’язування вважається успішним, якщо порушнику вдалося нав’язати будь-яке повідомлення, навіть співпадає з тим, що збиралася передавати Аліса.

В описаній стеганосистемі фактично використовуються тільки 2 приховуваних повідомлення виду "втеча сьогодні" та "втечу сьогодні скасовано", переданих за допомогою 6 стеганограм. Відзначимо, що, незважаючи на простоту цієї стеганосистеми, при її використанні забезпечується рівність у виразах (5.23) та (5.24), тобто вона є одночасно досконалою при атаці імітації і при атаці заміни першого порядку.

У стеганосистемах з автентифікацією порівняно з криптосистемами, що забезпечують контроль автентичності переданих повідомлень, виникає практична проблема такого порядку. При атаці імітації не настільки важливо як розділено множину контейнерів на підмножини, оскільки для зловмисника в момент нав’язування всі контейнери (стеганограми) рівновірогідні. Інша ситуація в атаці заміни. Якщо, перехопивши стеганограму, зловмисник здатний виявити, до якої підмножини контейнерів вона належить, то тим самим зловмисник повністю або частково визначив діючий ключ і знайшов здатність нав’язувати з недопустимо високою ймовірністю. Тому для забезпечення високої імітозахищеності стеганосистеми повинно бути складно (обчислювально складно) визначити, до якого підмножини належить будь-який стега. Очевидний спосіб досягнення цього полягає у випадковому рівновірогідному розбитті множини C на підмножини $C_1, C_2, \ldots, C_n$. Результат цього розбиття є секретним ключем автентифікації і повинен бути відомий тільки законним відправнику і одержувачу для підтвердження повідомлень. Однак обсяг цієї секретної інформації є надмірно великим для практичних стеганосистем. Другим способом є формування або відбір контейнерів за функціями формування або вибору з використанням секретної інформації автентифікації обмеженого обсягу при забезпеченні автентичності. Якщо отриманий стег може бути сгенерованим або обраним при діючому ключі, то витягнення з нього повідомлення значиться справжнім. У криптографії відомі подібні функції, стійкі

до їх аналізу зловмисником [51]. Однак істотні складності полягають в тому, що такі стійкі функції повинні породжувати не просто послідовності, які обчислювально не відрізняються від випадкових, а послідовності, що не відрізняються також від послідовностей, генерованих природними джерелами (мова, відео).

У криптографічних системах контроль автентичності переданої інформації забезпечується за допомогою імітовставок або цифрових підписів [51]. Імітовставки та цифрові підписи повідомлень, що завіряються описуються бернулієвським законом розподілу [50]. Отже, вони можуть бути легко помітні зловмисником від контейнерів природних джерел, що погіршує скритність стеганоканалу повідомлень, що завіряються. Отже, імітостійкі стеганосистеми не можуть копіювати принципи побудови криптографічних систем контролю справжності переданої інформації.

Стеганосистеми з автентифікації приховано переданих повідомлень у теоретичному і практичному плані знаходяться на початковому етапі свого розвитку і чекають своїх дослідників.

Контрольні запитання

1. Охарактеризуйте атаки проти систем прихованої передачі повідомлень.
2. Охарактеризуйте атаки на системи цифрових водяних знаків.
3. Назвіть класифікацію пасивних атак на стеганосистеми.
4. Розгляньте атаку на основі відомого заповненого контейнера.
5. Розгляньте атаку на основі відомого збудованого повідомлення.
6. Розгляньте атаку на основі обраного прихованого повідомлення.
7. Розгляньте атаку на основі обраного заповненого контейнера.
8. Розгляньте атаку на основі відомого порожнього контейнера.
9. Розгляньте атаку на основі обраного порожнього контейнера.
10. Назвіть класифікацію атак на стеганосистеми ЦВДЗ.
11. Охарактеризуйте атаки проти збудованого повідомлення.
12. Охарактеризуйте атаки проти стеганодетектора.
13. Охарактеризуйте атаки проти протоколу використання ЦВДЗ.
14. Охарактеризуйте атаки проти самого ЦВДЗ.
15. Охарактеризуйте атаки, спрямовані на видалення ЦВДЗ.
16. Охарактеризуйте атака шляхом статистичного усереднення.
17. У чому суть геометричної атаки.
18. Розгляньте атаки на детектор з жорстким рішенням.
19. Розгляньте атаки проти протоколу, що використовується.
20. Назвіть основні методи протидії атакам на системи ЦВДЗ.

Перелік скорочень

АРМ – автоматизоване робоче місце.
АЦП – аналоговий цифрової перетворювач.
ВЧ – високочастотний шум.
ГПВП – генератор псевдовипадкової перестановки.
ГПВФ – генератор псевдовипадкової функції.
ДКП – дискретне косинусне перетворення.
ЕЦП – електронний цифрової підпис.
ЗСЛ – зорова система людини.
ІС та Т – інформаційні системи та технології.
ІВК – інфраструктура відкритих ключів.
КС – комп'ютерна стеганографія.
КП – матриця косинусного перетворення.
НІСТ – Національний інститут стандартів та технологій США.
НЗБ – метод заміни найменш значущого біта.
НЧ – низькочастотний (шум).
ПВП – псевдовипадкова послідовність.
ПВСШ – пікове відношення сигнал – шум.
СКВ – середньоквадратичне відхилення.
УПФ – узагальнене перетворення Фур'є.
ЦАП – цифрової аналоговий перетворювач.
ЦВДЗ – цифрові відеозображення.
ЦВЗн – цифрові водяні знаки.
ЦОС – цифрова обробка сигналів.
ШСС – широкосмугові сигнали.
CR/LF – метод зміни порядку проходження маркерів кінця рядка.
ASCII – American Standard Code for Information Interchange – американський стандартний код для обміну інформацією. Набір з 128 кодів символів для машинного представлення великих і малих літер латинського алфавіту, чисел, роздільних знаків і спеціальних символів, кожному з яких відповідає конкретне 7-бітне двійкове число.
AAC – Advanced Audio Coding – широкосмуговий алгоритм кодування, формат аудіофайла з меншою втратою якості при кодуванні, ніж у MP3 при однакових розмірах.
BMP – BitMaP – бітове (растрове) відображення графічного об'єкта. Використовується для представлення зображень. Стандартний формат графічних файлів, який передбачає 4, 8 і 24 біти квантування на один піксел.

DCT – дискретні косинусні перетворення.

EBCOT – алгоритм "ієрархічне кодування блоків з оптимізованим усіканням".

EZW – алгоритм "вкладене нуль-дерево".

GIF – Graphics Interchange Format – формат обміну графічними даними.

ISO – International Standards Organization – Міжнародна організація стандартів.

JPEG – стандарт стиску із втратами для нерухливих повнобарвних відеозображень на основі алгоритму дискретного косінусного перетворення з коефіцієнтом компресії даних більше 25:1.

LSB – метод заміни найменш значущого біта.

MP3 – формат зберігання звукової інформації, забезпечує найкращий стиск.

MPEG – Moving Pictures Expert Group – група експертів у галузі зображень, які рухаються, основним завданням якої є розробка стандартів кодування рухливих зображень, звуку і їх комбінації.

NTSC – Національний комітет з телевізійних стандартів США.

OGG Vorbis – OGG Vorbis – відкритий стандарт мультимедіа-контейнера вільного формату стиску звуку з втратами.

RLE – алгоритм кодування повторів.

RGB – Red-Green-Blue – "червоний-зелений-синій" – основна палітра, використовувана в програмуванні й комп’ютерній графіці.

VPN – Virtual Private Network – віртуальна приватна мережа – підмережа корпоративної мережі, що забезпечує безпечний вхід у неї віддалених користувачів. Використовується для безпечного пересилання мережею Internet конфіденційних даних за рахунок інкапсуляції (тунелювання) IP-пакетів усередині інших пакетів, які потім маршрутизуються.

TIFF – Tagged Image File Format – формат зберігання растрових графічних зображень.

WAV – WAVE – формат файла-контейнера для зберігання записів оцифрованого аудіопотоку.

WMA – Windows Media Audio – ліцензований формат файла, розроблений компанією Microsoft для зберігання і трансляції аудіоінформації.

Використана література

1. Анин Б. Ю. Защита компьютерной информации / Б. Ю. Анин. – СПб. : BHV-Санкт-Петербург, 2000. – 284 с.

2. Барсуков В. С. Компьютерная стеганография: вчера, сегодня, завтра. Технологии информационной безопасности XXI века [Электронный ресурс] / В. С. Барсуков, А. П. Романцов // Специальная техника. – Режим доступа : http://st.ess.ru.

3. Вентцель Е. С. Теория вероятностей и ее инженерные приложения / Е. С. Вентцель, Л. А. Овчаров. – М. : Наука. Гл. ред. физ.-мат. лит., 1988. – 480 с.

4. Грибунин В. Г. Стеганографическая защита речевых сигналов в каналах открытой телефонной связи / В. Г. Грибунин, И. Н. Оков, И. В. Туринцев // Сборник тезисов Российской НТК "Методы и технические средства обеспечения безопасности информации". – СПб. : ГТУ, 2001. – С. 83–84.

5. Грибунин В. Г. Цифровая стеганография / В. Г. Грибунин, И. Н. Оков, И. В. Туринцев. – М. : Солон-Пресс, 2002. – 272 с.

6. Кан Д. Взломщики кодов / Д. Кан. – М. : Центрполиграф, 2000. – 473 с.

7. Конахович Г. Ф. Компьютерная стеганография. Теория и практика / Г. Ф. Конахович, А. Ю. Пузыренко. – К. : "МК-Пресс", 2006. – 288 с.

8. Некоторые проблемы противоборства в современных информационных системах / Ю. В. Коротков, Р. М. Ковалев, И. Н. Оков и др. // Сборник научных трудов Военного университета связи. – СПб. : 2001. – С. 56.

9. Оков И. Н. О требуемой пропускной способности каналов передачи аутентифицированных сообщений в безусловно стойких системах / И. Н. Оков // Проблемы информационной безопасности. Компьютерные системы. – 2000. – № 3(7). – С. 78–64.

10. Оков И. Н. Электронные водяные знаки как средство аутентификации передаваемых сообщений / И. Н. Оков, Р. М. Ковалев // Защита информации. Конфидент. – 2001. – № 3. – С. 80–85.

11. Основи комп'ютерної стеганографії : навч. посібн. для студентів і аспірантів / В. О. Хорошко, О. Д. Азаров, М. В. Шелест та ін. – Вінниця : ВДТУ, 2003. – 143 с.

12. Скляр Б. Цифровая связь: Теоретические основы и практическое применение / Б. Скляр. – 2-е изд, исправл. – М. : Вильямс, 2003. – 1104 с.

13. Соколов А. В. Защита информации в распределенных корпоративных сетях и системах / А. В. Соколов, В. Ф. Шаньгин. – М. : ДМК Пресс, 2002. – 656 с.

14. Стасєв Ю. В. Основи теорії побудови сигналів / Ю. В. Стасєв. – Х. : ХВУ, 1999. – 87 с.

15. Теория электрической связи : учебник для вузов / Зюко А. Г., Кловский Д. Д., Коржик В. И. и др. – М. : Радио и связь, 1999. – 432 с.

16. Термінологічний довідник з питань технічного захисту інформації / В. О. Хорошко, І. М. Огаркова, Д. В. Чирков та ін. ; за ред. проф. Хорошка В. О. – 3-тє вид., доп. і перероб. – К. : ТОВ "ПоліграфКонсалтинг", 2003. – 286 с.

17. Цифровые методы в космической связи / под ред. С. Голомба. – М. : Связь, 1969. – 272 с.

18. Чиссар И. Теория информации: Теоремы кодирования для дискретных систем без памяти / И. Чиссар, Я. Кернер ; пер. с англ. – М. : Мир, 1985. – 400 с.

19. Шеннон К. Работы по теории информации и кибернетики / К. Шеннон ; пер. с англ. – М. : Иностр. литература, 1963. – 829 с.

20. A DWT-based technique for spatio-frequency masking of digital signatures / M. Barni, F. Bartolini, V. Cappellini, A. Lippi, A. Piva // Proceedings of the 11th SPIE Annual Symposium, Electronic Imaging '99, Security and Watermarking of Multimedia Contents. – 1999. – Vol. 3657.

21. Anderson R. Proc. Int. Workshop on Information Hiding: Lecture Notes in Computer Science. Springer-Verlag / R. Anderson. – Cambridge, 1996.

22. Arnold M. MP3 robust audio watermarking / M. Arnold, S. Kanka // International Watermarking Workshop, 1999.

23. A secure, robust watermark for multimedia / I. J. Cox, J. Kilian, T. Leighton, T. G. Shamoon // Information hiding: first international workshop. Lecture Notes in Comp. Science. – 1996. – Vol. 1174. – P. 183–206.

24. Attack Modelling: Towards a Second Generation Watermarking Benchmark / S. Voloshynovskiy, S. Pereira, V. Iquise, T. Pun. // Preprint. University of Geneva, 2001. – 58 p.

25. Barlow J. P. The economy of ideas / J. P. Barlow // Wired Magazine. – 1994. – № 2.

26. Cachin C. An Information-Theoretic Model for Steganography / C. Cachin // Proceeding of the Workshop on Information Hiding. – 1998.

27. Chae J. J. A robust embedded data from wavelet coeffcients / J. J. Chae, D. Mukherjee, B. S. Manjunath // Proceedings of SPIE, Electronic

Imaging, Storage and Retrieval for Image and Video Database. – 1998. – Vol. 3312. – P. 308–317.

28. Collberg C. On the limits of software watermarking / C. Collberg, C. Thomborson // Technical report, University of Auckland, New Zealand, 1998.

29. Corvi M. Wavelet-based image watermarking for copyright protection / M. Corvi, G. Nicchiotti // Scandinavian Conference on Image Analysis. – 1997.

30. Diffie W. New directions in cryptography / W. Diffie, M. E. Hellman // IEEE Trans. on Information Theory. – 1976. – Vol. 22. – № 6. – P. 644–654.

31. Embedded block coding in JPEG 2000 / D. Taubman, E. Ordentlich, M. Weinberger, G. Seroussi // Signal Processing: Image Communication. – 2002. – № 17. – P. 49–72.

32. Fridrich J. Steganalysis ofLSB Encoding in Color Images / J. Fridrich, R. Du, M. Long // Proceedingsof ICME. – New York, 2000. – July 31 – August 2.

33. Girod B. The information theoretical significance of spatial and temporal masking in video signals / B. Girod // Proc. of the SPIE Symposium on Electronic Imaging. – 1989. – Vol. 1077. – P. 178–187.

34. Hsu C.-T. DCT-Based Watermarking for Video / C.-T. Hsu, J.-L. Wu // IEEE Transactions on Consumer Electronics. – 1998. – Vol. 44. – № 1, Feb. – P. 206–216.

35. Husrev T. Sencar. Data Hiding Fundamentals And Applications / Husrev T. Sencar, Mahalingam Ramkumar, Ali N. Akansu // Digital Multimedia. ELSEVIER science and technology books. – 2004. – 364 p.

36. Kerckhoffs A. La Cryptographic Militaire / A. Kerckhoffs // Journal des sciences militaires. – 1983. – Jan. – P. 5 – 83; 1983. – Feb. – P. 161–191.

37. Kim J. R. A robust wavelet-based digital watermark using level-adaptive thresholding / J. R. Kim, Y. S. Moon // Proceedings of the 6th IEEE International Conference on Image Processing. – 1999. – P. 202.

38. Kim Y.-S. Wavelet based watermarking method for digital images using the human visual system / Y.-S. Kim, O.-H. Kwon, R.-H. Park // Electronic Letters. – 1999. – № 35(6). – P. 466–467.

39. Koch E. Towards Robust and Hidden Image Copyright Labeling / E. Koch, J. Zhao // IEEE Workshop on Nonlinear Signal and Image Processing. – 1995. – P. 123–132.

40. Kundur D. A robust digital image watermarking method using wavelet-based fusion / D. Kundur, D. Hatzinakos // Proceedings of the IEEE International Conference on Image Processing. – 1997. – Vol. 1. – P. 544–547.

41. Kutter M. Digital signature of color images using amplitude modulation / M. Kutter, F. Jordan, F. Bossen // Proc. of the SPIE Storage and Retrieval for Image and Video Databases V. – 1997. – Vol. 3022. – P. 518–526.

42. Lewis A. Image compression using the 2-d wavelet transform / A. Lewis, G. Knowles // IEEE Transactions on Image Processing, 1992. – № 2. – P. 244–250.

43. Low cost spatial watermarking / V. Darmstaedter, J.-F. Delaigle, J. Quisquater, B. Macq // Computers and Graphics. – 1998. – Vol. 5. – P. 417–423.

44. Lu C.-S. Oblivious watermarking using generalized gaussian / C.-S.Lu, H.-Y. M. Liao // Proceedings of the 7th International Conference on Fuzzy Theory and Technology.– 2000. – P. 260–263.

45. Maes M. Digital image waermarking by salient point modification practical results / M. Maes, P. Rongen, van Overveld C. // SPIE Conference on Security and Watermarking of Multimedia Contents. – 1999. – Vol. 3657. – P. 273–282.

46. Marvel L. Image Steganography for hidden communication. PhD Thesis. / L. Marvel // Univ.of Delaware, 1999. – 115 p.

47. Marvel L. Reliable Blind Information Hiding for Images / L. Marvel, C. Boncelet, J. Retter // Proceedings of 2nd Workshop on Information Hiding. Lecture Notes in Computer Science. – 1998.

48. Menezes A.J. Handbook of applied cryptography / A. J. Menezes, P. C. Oorschot, S. A.Vanstone. – CRC Press, 1996. – p. 780

49. Modeling the Security of Steganographic Systems / J. Zollner, H. Federrath, H. Klimant, A. Pfitzmann, R. Piotraschke, A. Westfeld, G. Wicke, G. Wolf // Proceeding of the Workshop on Information Hiding. – 1998.

50. Moskowitz I. S. A new paradigm hidden in Steganography / I. S. Moskowitz, G. E. Longdon, L. Chang // Proceedings of Workshop "New Security Paradigms". – ACM Press. – 2000. – P. 41–50.

51. Osborne C. A Digital Watermark / C. Osborne, R. van Schyndel, A. Tirkel // IEEE In-tern. Conf. on Image Processing. – 1994. – P. 86–90.

52. Provos N. Defending Against on Statistical Steganalysis / N. Provos // Proceeding of the 10 USENIX Security Symposium. – 2001. – P. 323–335.

53. Provos N. Detecting Steganographic Content on the Internet / N. Provos, P. Honeyman // Proceeding of the 10 USENIX Security Symposium. – 2001. – P. 323–335.

54. Said A. A new, fast, and efficient image codec based on set partitioning in hierarchical trees / A. Said, W. Pearlman // IEEE Trans. on Circuits and Systems for Video Technology. – 1996. – № 3. – P. 243–250.

55. Schneier B. Applied Cryptography: Protocols, Algorithms, and Source Code in C / B. Schneier. – 2-d ed. – New York : John Wiley and Sons, 1996.

56. Secure spread spectrum watermarking for images, audio and video / I. J. Cox, J. Kilian, T. Leighton, T. G. Shamoon // Proceedings of the IEEE International Conference on Image Processing. – 1996. – P. 243–246.

57. Secure spread spectrum watermarking for multimedia / I. J. Cox, J. Kilian, T. Leighton, T. G. Shamoon // Proceedings of the IEEE International Conference on Image Processing. – 1997. – Vol. 6. – P. 1673–1687.

58. Shannon C. E. A Mathematical Theory of Communication / C. E. Shannon // Bell System Technical Journal. – 1948. – № 27. – P. 379–423, 623–656.

59. Shapiro J. Embedded image coding using zerotrees of wavelet coefficients / J. Shapiro // IEEE Trans. on Signal Processing. – 1993. – № 12. – P. 3445–3462.

60. Shoham Y. Efficient bit allocation for an arbitrary set of quantizers / Y. Shoham, A. Gersho // IEEE Trans. Acoustics, Speech, and Signal Processing. – 1988. – № 9. – P. 1445–1453.

61. Simmons G. J. Autentication theory/coding theory / G. J. Simmons // Advances in Cryptology. Proc. CRYPTO-84. Proceedings. – P. 411–431.

62. Simmons G. J. The subliminal channel and digital signatures / Simmons G. J. // Advances in Cryptology. Proc. EUROCRYPT-84. – P. 364–378.

63. Simmons G. The History of Subliminal Channels / G. Simmons // IEEE Journal on Selected Areas of Communications. – 1998. – Vol. 16. – № 4. – P. 452–461.

64. Smith J. Modulation and Information hiding in Image / J. Smith, B. Comiskey // Information hiding: First Int. Workshop "InfoHiding’96", Springer as Lecture Notes in Computing Science. – 1996. – Vol 1174. – P. 207–227.

65. Techniques for Data Hiding / W. Bender, D. Gruhl, N. Morimoto, A. Lu // IBM Systems Journal. – 1996. – Vol. 35. – P. 313–336.

66. Watson A. The cortex transform: rapid computation of simulated neural images / A. Watson // Computer Vision, Graphics, and Image Processing. – 1987. – Vol. 39. – № 3. – P. 311–327.

67. Westfeld A. Attacks on Steganographic Systems. Breaking the Steganographic Utilities EzStego, Jsteg, Steganos, and S-Tools – and Some Leassons Learned / A. Westfeld, A. Pfitzmann // Proceeding of the Workshop on Information Hiding. – 1999.

Додатки

Додаток А

Приховування інформації. Метод заміни найменш значущого біта

ORIGIN := 1

C := READRGB("C.bmp")

R := READ_RED("C.bmp")

G := READ_GREEN("C.bmp")

B := READ_BLUE("C.bmp")

M := READBIN("M.TXT", "byte")

i := 1..256

$A_i := i - 1$

K := "@J|eKc-|9

Nk := strlen(K)

Na := rows(A)

Nm := rows(M)

K' :=
K ← str2vec(K)
for i ∈ 1..Nm
  r ← mod(i, Nk)
  $K'_i \leftarrow K_r$ if r
  $K'_i \leftarrow K_{Nk}$ if
K'

M_cod :=
for j ∈ 1..Nm
  for i ∈ 1..Na
    m ← i if $M_j = A_i$
    n ← i if $K'_j = A_i$
  r ← mod(m + n, Na)
  $M\_cod_j \leftarrow A_r$ if r > 0
  $M\_cod_j \leftarrow A_{Na}$ if r = 0
M_cod

$\mu s := \text{"n0ch@m0k"}$ $\mu e := \text{"KiHeu,6"}$

$sMe := stack(str2vec(\mu s), M\_cod, str2vec(\mu e$

$8 rows(sMe) = 9.016 \times 10^3$

$$B2D(x) := \sum_{i=1}^{8} \left(x_i \cdot 2^{i-1} \right)$$

D2B(x) :=
- for $i \in 1..8$
  - $V_i \leftarrow mod(x, 2)$
  - $x \leftarrow floor\left(\frac{x}{2}\right)$
- V

Cv :=
- $C' \leftarrow augment(B, G, R)$
- $Cv \leftarrow C'^{\langle 1 \rangle}$
- for $i \in 2..cols(C')$
  - $Cv \leftarrow stack(Cv, C'^{\langle i \rangle})$

Sv :=
- for $\mu \in 1..rows(sMe)$
  - $b \leftarrow D2B(sMe_\mu)$
  - for $i \in 1..8$
    - $P \leftarrow D2B[Cv_{i+8\cdot(\mu-1)}]$
    - $P_1 \leftarrow b_i$
    - $Sv_{i+8\cdot(\mu-1)} \leftarrow B2D(P)$
- for $j \in rows(Sv) + 1..rows(Cv)$
  - $P \leftarrow D2B(Cv_j)$
  - $P_1 \leftarrow round(rnd(1))$
  - $Sv_j \leftarrow B2D(P)$
- Sv

S' := for $i \in 1..cols(C)$

$S'^{\langle i \rangle} \leftarrow submatrix[Sv, (i-1)\cdot rows(C) + 1, i\cdot rows(C), 1, 1]$

$$Bm := submatrix\left(S', 1, rows(C), 1, \frac{cols(C)}{3}\right)$$

$$Gm := submatrix\left(S', 1, rows(C), \frac{cols(C)}{3} + 1, 2\cdot\frac{cols(C)}{3}\right)$$

$$Rm := submatrix\left(S', 1, rows(C), 2\cdot\frac{cols(C)}{3} + 1, 3\cdot\frac{cols(C)}{3}\right)$$

$S := augment(Rm, Gm, Bm)$

$WRITERGB(\text{"2.bmp"}) := S$

Вилучення інформації. Метод заміни найменш значущого біта

$$B2D(x) := \sum_{i=1}^{8} \left(x_i \cdot 2^{i-1} \right)$$

D2B(x) :=
for $i \in 1..8$
  $V_i \leftarrow \mathrm{mod}(x, 2)$
  $x \leftarrow \mathrm{floor}\left(\frac{x}{2}\right)$
V

R := READ_RED("2.bmp")
G := READ_GREEN("2.bmp")
B := READ_BLUE("2.bmp")

Sv :=
S' ← augment(B, G, R)
Sv ← S'⟨1⟩
for $i \in 2..\mathrm{cols}(S')$
  $Sv \leftarrow \mathrm{stack}\left(Sv, S'^{\langle i \rangle}\right)$

Mf :=
for $\mu \in 1..\frac{\mathrm{rows}(Sv)}{8}$
  for $i \in 1..8$
    $P \leftarrow D2B\left[Sv_{i+8\cdot(\mu-1)}\right]$
    $b_i \leftarrow P_1$
  $Mf_\mu \leftarrow B2D(b)$
  $Mf_\mu \leftarrow Mf_\mu + 32.5 \text{ if } Mf_\mu < 32$
Mf

N :=
for $s \in 0..31$
  $i \leftarrow 1$
  for $\mu \in 1..\mathrm{rows}(Mf)$
    if $Mf_\mu = s + 32.5$
      $N_{i,s+1} \leftarrow \mu$
      $i \leftarrow i + 1$
N

μs := "n0ch@m0k"

M_cod :=
s ← 0
e ← 0
βs ← strlen(μs)
βe ← strlen(μe)
Mf ← vec2str(Mf)
for $\mu \in 1..\mathrm{strlen}(Mf)$
  $s \leftarrow \mu + \beta s \text{ if } \mathrm{substr}(Mf, \mu, \beta s) = \mu s \wedge s = 0$
  $e \leftarrow \mu - 1 \text{ if } \mathrm{substr}(Mf, \mu, \beta e) = \mu e \wedge e = 0$
  break if $s \neq 0 \wedge e \neq 0$
Mf ← substr(Mf, s, e − βs)
M_cod ← str2vec(Mf)
for $n \in 1..\mathrm{cols}(N)$
  for $i \in 1..\mathrm{rows}(N)$
    break if $N_{i,n} = 0$
    $M\_cod_{N_{i,n}-\beta s} \leftarrow n - 1 \text{ if } 0 < N_{i,n} \leq \mathrm{rows}(M\_cod) + \beta s$
M_cod

μe := "KiHeu,6" i := 1..256

```
K' := | K ← str2vec(K)
      | for i ∈ 1..Nm
      |     r ← mod(i, Nk)
      |     K'_i ← K_r   if r > 0
      |     K'_i ← K_Nk  if r = 0
      | K'

M := | for j ∈ 1..Nm
     |     for i ∈ 1..Na
     |         m ← i  if M_cod_j = A_i
     |         n ← i  if K'_j = A_i
     |     r ← mod(Na + m − n, Na)
     |     M_j ← A_r   if r > 0
     |     M_j ← A_Na  if r = 0
     | M

A_i := i − 1

Na := rows(A)

Na = 256

WRITEBIN("M_dec.txt", "byte", 0) := M

K := "@J|eKc-|98o"

Nk := strlen(K)

Nk = 11

Nm := rows(M_cod)

Nm = 1.112 × 10^3
```

Приховування інформації. Метод псевдовипадкової перестановки

$$B2D(x) := \sum_{i=1}^{8} \left(x_i \cdot 2^{i-1} \right)$$

$$D2B(x) := \left| \begin{array}{l} \text{for } i \in 1..8 \\ \quad \left| \begin{array}{l} V_i \leftarrow \text{mod}(x, 2) \\ x \leftarrow \text{floor}\left(\frac{x}{2} \right) \end{array} \right. \\ V \end{array} \right.$$

M := "My name is Alexandr, 2

μs := 15

μe := "KiHeu,6"

B := READ_BLUE("C.bmp")

R := READ_RED("C.bmp")

G := READ_GREEN("C.bmp")

Koef := 9

$$Step(x) := Koef \cdot \sum_{i=1}^{rows(x)} x_i$$

$$Me := str2vec\left(concat\left(M, \mu e\right)\right)$$

$$Cv := \left| \begin{array}{l} Cv \leftarrow B^{\langle 1 \rangle} \\ \text{for } i \in 2..cols(B) \\ \quad Cv \leftarrow stack\left(Cv, B^{\langle i \rangle}\right) \end{array} \right.$$

$$Sv := \left| \begin{array}{l} Sv \leftarrow Cv \\ z \leftarrow \mu s \\ \text{for } \mu \in 1..rows(Me) \\ \quad \left| \begin{array}{l} b \leftarrow D2B\left(Me_{\mu}\right) \\ \text{for } i \in 1..8 \\ \quad \left| \begin{array}{l} z \leftarrow z + Step(D2B(z)) \\ P \leftarrow D2B\left(Cv_z\right) \\ P_1 \leftarrow b_i \\ Sv_z \leftarrow B2D(P) \end{array} \right. \end{array} \right. \\ Sv \end{array} \right.$$

$$S' := \text{for } i \in 1..cols(B)$$

$$S^{\langle i \rangle} \leftarrow submatrix[Sv, (i-1) \cdot rows(B) + 1, i \cdot rows(B), 1, 1]$$

S := augment(R, G, S')

WRITERGB("C_cod.bmp") := S

Вилучення інформації. Метод псевдовипадкової перестановки

$$B2D(x) := \sum_{i=1}^{8} \left(x_i \cdot 2^{i-1} \right)$$

$$D2B(x) := \left| \begin{array}{l} \text{for } i \in 1..8 \\ \quad \left| \begin{array}{l} V_i \leftarrow \text{mod}(x, 2 \\ x \leftarrow \text{floor}\left(\frac{x}{2}\right) \end{array} \right. \\ V \end{array} \right.$$

Koef := 9

$$step(x) := Koef \cdot \sum_{i=1}^{rows(x)}$$

B := READ_BLUE("C_cod.bmp")

μs := 154

μe := "KiHeu,6"

$$Cv := \left| \begin{array}{l} Cv \leftarrow B^{\langle 1 \rangle} \\ \text{for } i \in 2..cols(B) \\ \quad Cv \leftarrow stack\left(Cv, B^{\langle i \rangle}\right) \end{array} \right.$$

$$Mf := \left| \begin{array}{l} z \leftarrow \mu s \\ \text{for } \mu \in 1..rows(Cv) \\ \quad \left| \begin{array}{l} \text{for } i \in 1..8 \\ \quad \left| \begin{array}{l} z \leftarrow z + step(D2B \\ \text{break if } z > rows \\ P \leftarrow D2B\left(Cv_z\right) \\ b_i \leftarrow P_1 \end{array} \right. \\ Mf_\mu \leftarrow B2D(b) \\ \mu \leftarrow \mu + 1 \end{array} \right. \\ Mf \end{array} \right.$$

mu := str2vec(μe)

μe = "KiHeu,6"

$$mu = \begin{pmatrix} 75 \\ 105 \\ 72 \\ 101 \\ 117 \\ 44 \\ 54 \end{pmatrix}$$

Приховування інформації. Метод псевдовипадкового інтервалу

$$B2D(x) := \sum_{i=1}^{8} \left(x_i \cdot 2^{i-1} \right)$$

D2B(x) :=
for $i \in 1..8$
  $V_i \leftarrow \mathrm{mod}(x, 2)$
  $x \leftarrow \mathrm{floor}\left(\frac{x}{2}\right)$
V

M := "My name is Alexandr, 2

C := READ_BLUE("C.bmp")

Lm := strlen(M)·8

Lm = 248

X := rows(C)

Y := cols(C)

K0 := 125 ψ := 6

K :=
for $s \in 1..2 \cdot \psi$
  $K_s \leftarrow K0$ if $s = 1$
  $K_s \leftarrow \mathrm{str2num}\left[\mathrm{substr}\left[\mathrm{num2str}\left[\left(K_{s-1}\right)^2\right], 1, 3\right]\right]$ if $s > 1$
  $K_s \leftarrow \mathrm{str2num}\left(\mathrm{substr}\left(\mathrm{num2str}\left(K_s\right), 1, 2\right)\right)$ if $K_s > 255$
K

$K^T =$

	1	2	3	4	5	6	7	8	9	10	11	12
1	125	156	243	59	34	115	132	174	30	90	81	65

S :=
S ← C
Mvec ← str2vec(M)
Mvec_bin ← D2B(Mvec_1)
for $j \in 2..\mathrm{rows}(Mvec)$
  Mvec_bin ← stack(Mvec_bin, D2B(Mvec_j))
for $i \in 1..Lm$
  $x \leftarrow \mathrm{floor}\left(\frac{i}{Y}\right) + 1$
  $y \leftarrow \mathrm{mod}(i, Y) + 1$
  for $s \in 1..\psi$
    $x \leftarrow \mathrm{mod}\left(x + B2D\left(\overrightarrow{\left(D2B\left(K_{2 \cdot s-1}\right) \oplus \mathrm{submatrix}(D2B(y), 1, 8, 1, 1)\right)}\right), X\right)$
    $y \leftarrow \mathrm{mod}\left(y + B2D\left(\overrightarrow{\left(D2B\left(K_{2 \cdot s}\right) \oplus \mathrm{submatrix}(D2B(x), 1, 8, 1, 1)\right)}\right), Y\right)$
  $P \leftarrow D2B\left(C_{x,y}\right)$
  $P_1 \leftarrow Mvec\_bin_i$
  $S_{x,y}$
S

S' := augment(READ_RED("C.bmp"), READ_GREEN("C.bmp"), S)

WRITERGB("C_cod.bmp") := S

Вилучення інформації. Метод псевдовипадкового інтервалу

$$B2D(x) := \sum_{i=1}^{8} \left(x_i \cdot 2^{i-1} \right)$$

$$D2B(x) := \left| \begin{array}{l} \text{for } i \in 1..8 \\ \quad \left| \begin{array}{l} V_i \leftarrow \text{mod}(x, 2) \\ x \leftarrow \text{floor}\left(\frac{x}{2} \right) \end{array} \right. \\ V \end{array} \right.$$

$$\psi := 6$$

$$K0 := 125$$

$$K := \left| \begin{array}{l} \text{for } s \in 1..2 \cdot \psi \\ \quad \left| \begin{array}{l} K_s \leftarrow K0 \text{ if } s = 1 \\ K_s \leftarrow \text{str2num}\left[\text{substr}\left[\text{num2str}\left[\left(K_{s-1} \right)^2 \right], 1, 3 \right] \right] \text{ if } s > 1 \\ K_s \leftarrow \text{str2num}\left(\text{substr}\left(\text{num2str}\left(K_s \right), 1, 2 \right) \right) \text{ if } K_s > 255 \end{array} \right. \\ K \end{array} \right.$$

$K^T =$

	1	2	3	4	5	6	7	8	9	10	11	12
1	125	156	243	59	34	115	132	174	30	90	81	65

$$S := READ\_BLUE("C\_cod.bmp")$$

$$X := rows(S)$$

$$Y := cols(S)$$

$$M := \left| \begin{array}{l} \text{for } i \in 1.. \frac{X \cdot Y}{8} \\ \quad x \leftarrow \text{floor}\left(\frac{i}{Y} \right) + 1 \\ \quad y \leftarrow \text{mod}(i, Y) + 1 \\ \quad \text{for } s \in 1..\psi \\ \quad \quad \left| \begin{array}{l} x \leftarrow \text{mod}\left(x + B2D\left(\overrightarrow{\left(D2B\left(K_{2 \cdot s-1} \right) \oplus \text{submatrix}\left(D2B(y), 1, 8, 1, 1 \right) \right)} \right), X \right) + 1 \\ y \leftarrow \text{mod}\left(y + B2D\left(\overrightarrow{\left(D2B\left(K_{2 \cdot s} \right) \oplus \text{submatrix}\left(D2B(x), 1, 8, 1, 1 \right) \right)} \right), Y \right) + 1 \end{array} \right. \\ \quad P \leftarrow D2B\left(S_{x,y} \right) \\ \quad Mvec\_bin_i \leftarrow P_1 \\ \text{for } j \in 1.. \frac{rows(Mvec\_bin)}{8} \\ \quad Mvec_j \leftarrow B2D(\text{submatrix}(Mvec\_bin, 8 \cdot j - 7, 8 \cdot j, 1, 1)) \\ vec2str(Mvec) \end{array} \right.$$

Метод розсунення рядка шляхом збільшення пробілів між словами

```
program StegoShift;
type
  StringType = string [$FF];
const
  TempName = '$$$$$$$$.$$$';
  Key1  = $1234;
  Key2  = $4567;
var
  F, G, H     : text;
  Line, Head, Tail, Body : StringType;
  B, I, K, L, N, LenBody, Z : byte;
  C       : char;
  Tab      : array [0..255] of byte;
  Count      : real;
begin
  LowVideo;
  if not (ParamCount in [1..3])
  then
   begin
     WriteLn ('Ожидается ввод 2 или 3 параметров:');
     WriteLn ('2 - вывод стеганограммы в файл;');
     WriteLn ('3 - запись стеганограммы в файл.');
     Exit
   end;
  Assign (F, ParamStr (1));
  Reset (F);
  Count := 0;
  MemW [Dseg : $01FE] := Key1;
  MemW [Dseg : $01FC] := Key2;
(*----------------------ВЫВОД СТЕГАНОГРАММЫ В ФАЙЛ----------------------*)
  if ParamCount = 2
  then
   begin
     if Pos (':', ParamStr (2)) <> 2
     then Assign (G, TempName)
     else Assign (G, Copy (ParamStr (2), 1, 2) + TempName);
     Rewrite (G);
     Z := 0;
     L := 0;
     while not Eof (F)
     do
     begin
       ReadLn (F, Line);
     (* 1. Выделяем тело строки *)
       while (Line <> '')
       and (Line [1] <= ' ')
       do Delete (Line, 1, 1);
       while (Line <> '')
       and (Line [Length (Line)] <= ' ')
       do Delete (Line, Length (Line), 1);
     (* 2. Заполняем таблицу пробелов *)
       FillChar (Tab, SizeOf (Tab), 0);
       LenBody := Length (Line);
       K := 0;
       I := 0;
       while I < LenBody
       do
        begin
         while (I < LenBody) and (Line [I] <> ' ')
         do I := Succ (I);
         if (I < LenBody) and (Line [I] = ' ')
         then
          begin
            K := Succ (K);
```

```
          N := 0;
          while (I < LenBody)
          and (Line [I] = ' ')
          do
           begin
            N := Succ (N);
            I := Succ (I)
           end;
          Tab [K] := N
         end
       end;
      if K > 0
      then K := Pred (K);
     (* 3. Декодируем биты *)
      I := 0;
      while I < K
      do
       begin
        I := Succ (I);
        if Tab [I] > 1
        then
         begin
          Z := Z shr 1;
          if Odd (Tab [I])
          then Z := Z or $80;
          L := Succ (L) mod 8;
          if L = 0
          then
           begin
            C := Chr (Z
            xor Random (256));
            Count := Count + 1;
            Write (C);
            Write (G, C);
            Z := 0
           end
         end
       end;
    end;
   (* 4. Завершаем работу *)
    Close (F);
    Close (G);
    Assign (F, ParamStr (2));
(*$I-*)
    Erase (F);
(*$I+*)
    I := IoResult;
    Rename (G, ParamStr (2));
    if WhereX <> 1
    then WriteLn;
    WriteLn ('Прочитано ', Count : 0 : 0, ' байт стега...');
    Exit
  end;
(*----------------------ЗАПИСЬ СТЕГАНОГРАММЫ В ФАЙЛ--------------------*)
  Assign (G, ParamStr (2));
  if Pos (':', ParamStr (3)) <> 2
  then Assign (H, TempName)
  else Assign (H, Copy (ParamStr (3), 1, 2) + TempName);
  Reset (G);
  Rewrite (H);
  L := 0;
  while not Eof (F)
  do
   begin
   (* 1. мниициализируем таблицу раздвижек *)
    FillChar (Tab, SizeOf (Tab), 0);
```

```
(* 2. Читаем и разделяем строку на части *)
 ReadLn (F, Line);
 I := 0;
 while (I < Length (Line)) and (Line [Succ (I)] <= ' ')
 do I := Succ (I);
 Tab [0] := I;
 (* начало строки *)
 Head := Copy (Line, 1, I);
 I := Length (Line);
 while (I > 0) and (Line [I] <= ' ')
 do I := Pred (I);
 (* конец строки *)
 Tail := Copy (Line, Succ (I), Length (Line) - I);
 (* тело строки *)
 Body := Copy (Line, Succ (Tab [0]), I - Tab [0]);
(* 3. Редуцируем тело строки *)
 LenBody := Length (Body);
 while Pos (' ', Body) > 0
 do Delete (Body, Pos (' ', Body), 1);
 (* число вставляемых пробелов *)
 N := LenBody - Length (Body);
(* 4. Заполняем таблицу раздвижек *)
 K := 0;
 for I := 1 to Length (Body)
 do
  if Body [I] = ' '
  then
   begin
     K := Succ (K);
     Tab [K] := 1
   end;
(* 5. Распределяем значимые (информационные) пробелы *)
 I := 1;
 while I < K
 do
  begin
   if L = 0
   then (* извлекаем очередной байт *)
     begin
      if Eof (G)
      then C := #00
      else Read (G, C);
      Count := Count + 1;
      Z := Ord (C) xor Random (256);
      B := Z and 1;
      L := 1;
      Write (C);
     end;
   if N > Succ (B)
   then (* запас пробелов не исчерпан *)
     begin
      (* кодируем бит в таблице *)
      Tab [I] := Tab [I] + Succ (B);
      (* текущий запас пробелов *)
      N := N - Succ (B);
      (* указываем следующий бит *)
      Z := Z shr 1;
      B := Z and 1;
      (* счётчик записанных битов *)
      L := Succ (L) mod 9
     end;
   I := Succ (I)
  end;
(* 6. Монотонно перераспределяем информационные пробелы *)
 if K > 2
 then (* число пробелов должно возрастать к концу строки *)
  begin
```

```
      I := 0;
      while (Tab [Pred (K)] = 1) and (I < K)
      do
       begin
        Move (Tab [1], Tab [2], K - 2);
        Tab [1] := 1;
        I := Succ (I)
       end
     end;
   (* 7. Распределяем выравнивающие (незначимые) пробелы *)
    while N > 1
    do
     begin
      I := K;
      while (I > 0) and (N > 1)
      do
       begin
        if (Tab [I] > 1) or (I = K)
        then
         begin
          Tab [I] := Tab [I] + 2;
          N := N - 2
         end;
        I := Pred (I)
       end
     end;
    Tab [K] := Tab [K] + N;
   (* 8. Вставляем пробелы в тело строки *)
    N := Length (Body);
    while (N > 0) and (K > 0)
    do
     begin
      while (N > 0) and (Body [N] <> ' ')
      do N := Pred (N);
      if (Tab [K] > 0) and (N > 0)
      then
        for I := 1 to Pred (Tab [K])
        do Insert (' ', Body, N);
      if K > 0
      then K := Pred (K);
      if N > 0
      then N := Pred (N)
     end;
   (* 9. Формируем и записываем строку *)
    Line := Head + Body + Tail;
    WriteLn (H, Line)
   end;
(* Заканчиваем обработку *)
  Close (F);
  Close (G);
  Close (H);
  Assign (F, ParamStr (3));
(*$I-*)
  Erase (F);
(*$I+*)
  I := IoResult;
  Rename (H, ParamStr (3));
  if WhereX <> 1
  then WriteLn;
  if (Count <> 0) and (L <> 0)
  then Count := Count - 1;
  WriteLn ('Записано ', Count : 0 : 0, ' байт стега...');
  Exit
end.
```

Метод зміни порядку проходження маркерів кінця рядка

```
program StegoCR_LF;
type
  StringType = string [$FF];
const
  TempName = '$$$$$$$$.$$$';
  Key1  = $1234;
  Key2  = $4567;
  CR  = #$0D;
  LF  = #$0A;
var
  F, G, H : text;
  X   : StringType;
  I   : integer;
  K, L  : byte;
  C, CC  : char;
  Count  : real;
begin
  LowVideo;
  if not (ParamCount in [1..3])
  then
   begin
     WriteLn ('Ожидается от 1 до 3 параметров:');
     WriteLn ('1 - вывод стеганограммы на экран;');
     WriteLn ('2 - вывод стеганограммы в файл;');
     WriteLn ('3 - запись стеганограммы в файл.');
     Exit
   end;
  Assign (F, ParamStr (1));
  Reset (F);
  Count := 0;
  MemW [Dseg : $01FE] := Key1;
  MemW [Dseg : $01FC] := Key2;
(*----------------------ВЫВОД СТЕГАНОГРАММЫ НА ЭКРАН----------------------*)
  if ParamCount = 1
  then
   begin
     C := #00;
     K := 0;
     L := 0;
     while not Eof (F)
     do
     begin
       Read (F, CC);
       if C = CR
       then
        if CC = LF
        then
         begin
           L := Succ (L) mod 8;
           K := K shr 1;
           C := #00;
           if L = 0
           then
            begin
              Write (Chr (K));
              K := 0;
              L := 0
            end
         end
        else C := CC
       else
        if C = LF
        then
```

```
        if CC = CR
        then
          begin
           L := Succ (L) mod 8;
           K := K shr 1 or $80;
           C := #00;
           if L = 0
           then
            begin
              Write (Chr (K));
              K := 0;
              L := 0
            end
          end
        else C := CC
       else C := CC
    end;
    Close (F);
    if WhereX <> 1
    then WriteLn;
    WriteLn ('Ok!');
    Exit
  end;
(*----------------------ВЫВОД СТЕГАНОГРАММЫ В ФАЙЛ----------------------*)
  if ParamCount = 2
  then
   begin
    if Pos (':', ParamStr (2)) <> 2
    then Assign (G, TempName)
    else Assign (G, Copy (ParamStr (2), 1, 2) + TempName);
    C := #00;
    K := 0;
    L := 0;
    Assign (G, TempName);
    Rewrite (G);
    while not Eof (F)
    do
    begin
     Read (F, CC);
     if C = CR
     then
      if CC = LF
      then
       begin
        L := Succ (L) mod 8;
        K := K shr 1;
        C := #00;
        if L = 0
        then
         begin
          K := K xor Random (256);
          Count := Count + 1;
          Write (G, Chr (K));
          Write (Chr (K));
          K := 0;
          L := 0
         end
       end
      else C := CC
     else
      if C = LF
      then
       if CC = CR
       then
         begin
          L := Succ (L) mod 8;
          K := K shr 1 or $80;
```

```
          C := #00;
          if L = 0
          then
           begin
             K := K
             xor Random (256);
             Count := Count + 1;
             Write (G, Chr (K));
             Write (Chr (K));
             K := 0;
             L := 0
           end
         end
        else C := CC
       else C := CC
    end;
    Close (F);
    Close (G);
    Assign (F, ParamStr (2));
(*$I-*)
    Erase (F);
(*$I+*)
    I := IoResult;
    Rename (G, ParamStr (2));
    if WhereX <> 1
    then WriteLn;
    WriteLn ('Прочитано ', Count : 0 : 0, ' байт стега...');
    Exit
  end;
(*----------------------ЗАПИСЬ СТЕГАНОГРАММЫ В ФАЙЛ---------------------*)
  Assign (G, ParamStr (2));
  if Pos (':', ParamStr (3)) <> 2
  then Assign (H, TempName)
  else Assign (H, Copy (ParamStr (3), 1, 2) + TempName);
  Reset (G);
  Rewrite (H);
  L := 0;
  while not Eof (F)
  do
   begin
    ReadLn (F, X);
    Write (H, X);
    if L = 0
    then
     begin
       Read (G, C);
       Count := Count + 1;
       Write (C);
       K := Ord (C) xor Random (256)
     end;
    if K and 1 = 0
    then Write (H, CR + LF)
    else Write (H, LF + CR);
    K := K shr 1;
    L := Succ (L) mod 8
   end;
  Close (F);
  Close (G);
  Close (H);
  Assign (F, ParamStr (3));
(*$I-*)
  Erase (F);
(*$I+*)
  I := IoResult;
  Rename (H, ParamStr (3));
  if WhereX <> 1
  then WriteLn;
```

```
  if (Count <> 0) and (L <> 0)
  then Count := Count - 1;
  WriteLn ('Записано ', Count : 0 : 0, ' байт стега...');
  Exit
end.
```

Метод хвостових пробілів

```
program StegoBlank;
type
  StringType = string [$FF];
const
  TempName = '$$$$$$$$.$$$';
  Key1  = $1234;
  Key2  = $4567;
  Max  = 240;
var
  F, G, H : text;
  Flag  : boolean;
  X   : StringType;
  I, K, L : byte;
  C   : char;
  Count  : real;
begin
  LowVideo;
  if not (ParamCount in [1..3])
  then
   begin
    WriteLn ('Ожидается от 1 до 3 параметров:');
    WriteLn ('1 - вывод стеганограммы на экран;');
    WriteLn ('2 - вывод стеганограммы в файл;');
    WriteLn ('3 - запись стеганограммы в файл.');
    Exit
   end;
  Assign (F, ParamStr (1));
  Reset (F);
  Count := 0;
  MemW [Dseg : $01FE] := Key1;
  MemW [Dseg : $01FC] := Key2;
(*----------------------ВЫВОД СТЕГАНОГРАММЫ НА ЭКРАН----------------------*)
  if ParamCount = 1
  then
   begin
    Flag := False;
    while not Eof (F)
    do
    begin
     ReadLn (F, X);
     L := Length (X);
     if L < Max
     then
      begin
       while X [L] = ' '
       do L := Pred (L);
       L := Length (X) - L;
       if not Flag
       then K := L
       else Write (Chr (K or L shl 4));
       Flag := not Flag
      end
    end;
    Close (F);
    if WhereX <> 1
    then WriteLn;
    WriteLn ('Ok!');
    Exit
   end;
```

```
(*----------------------ВЫВОД СТЕГАНОГРАММЫ В ФАЙЛ----------------------*)
  if ParamCount = 2
  then
   begin
     if Pos (':', ParamStr (2)) <> 2
     then Assign (G, TempName)
     else Assign (G, Copy (ParamStr (2), 1, 2) + TempName);
     Assign (G, TempName);
     Rewrite (G);
     Flag := False;
     while not Eof (F)
     do
     begin
       ReadLn (F, X);
       L := Length (X);
       if L < Max
       then
        begin
          while X [L] = ' '
          do L := Pred (L);
          L := Length (X) - L;
          if not Flag
          then K := L
          else
           begin
             C := Chr ((K or L shl 4)
              xor Random (256));
             Write (G, C);
             Write (C);
             Count := Count + 1
           end;
          Flag := not Flag
        end
     end;
     Close (F);
     Close (G);
     Assign (F, ParamStr (2));
(*$I-*)
     Erase (F);
(*$I+*)
     I := IoResult;
     Rename (G, ParamStr (2));
     if WhereX <> 1
     then WriteLn;
     if Flag and (Count <> 0)
     then Count := Count - 1;
     WriteLn ('Прочитано ', Count : 0 : 0, ' байт стега...');
     Exit
   end;
(*----------------------ЗАПИСЬ СТЕГАНОГРАММЫ В ФАЙЛ----------------------*)
  Assign (G, ParamStr (2));
  if Pos (':', ParamStr (3)) <> 2
  then Assign (H, TempName)
  else Assign (H, Copy (ParamStr (3), 1, 2) + TempName);
  Reset (G);
  Rewrite (H);
  Flag := False;
  while not Eof (F)
  do
   begin
    ReadLn (F, X);
    while X [Length (X)] = ' '
    do X [0] := Pred (X [0]);
    Write (H, X);
    if Length (X) < Max - 15
    then
```

```
    begin
      if Flag
      then
       for I := 1 to K shr $04
       do Write (H, ' ')
      else
       begin
         Read (G, C);
         Write (C);
         Count := Count + 1;
         K := Ord (C) xor Random (256);
         for I := 1 to K and $0F
         do Write (H, ' ')
       end;
      Flag := not Flag
    end;
   WriteLn (H)
  end;
 Close (F);
 Close (G);
 Close (H);
 Assign (F, ParamStr (3));
(*$I-*)
 Erase (F);
(*$I+*)
 I := IoResult;
 Rename (H, ParamStr (3));
 if WhereX <> 1
 then WriteLn;
 if Flag and (Count <> 0)
 then Count := Count - 1;
 WriteLn ('Записано ', Count : 0 : 0, ' байт стега...');
 Exit
end.
```

Метод знаків однакового накреслення

```
program StegoChange;
type
  StringType = string [$FF];
  Index  = (Rus, Lat);
  SetOfChar = set of char;
const
  TempName = '$$$$$$$$.$$$';
  Max  = 13;
  Key1  = $1234;
  Key2  = $4567;
  Tab : array [Index, 1..22] of char = ('ВЕКМНРСТХаеосАОикпрту'#32,
         'ВЕКМНРСТХаеосАОuknpmy'#00);
var
  F, G, H : text;
  X   : StringType;
  I, J  : integer;
  K, L  : byte;
  C, CC  : char;
  LatSet : SetOfChar;
  CharSet : SetOfChar;
  LargeLat : SetOfChar;
  LargeChar : SetOfChar;
  Count  : real;
begin
  LowVideo;
  if not (ParamCount in [1..3])
  then
   begin
     WriteLn ('Ожидается от 1 до 3 параметров:');
     WriteLn ('1 - вывод стеганограммы на экран;');
```

```
    WriteLn ('2 - вывод стеганограммы в файл;');
    WriteLn ('3 - запись стеганограммы в файл.');
    Exit
   end;
  Assign (F, ParamStr (1));
  Reset (F);
  Count := 0;
  MemW [Dseg : $01FE] := Key1;
  MemW [Dseg : $01FC] := Key2;
  L  := 0;
  LatSet := [];
  CharSet := [];
  for I := 1 to Max
  do
   begin
    LatSet := LatSet + [Tab [Lat, I]];
    CharSet := CharSet + [Tab [Rus, I]] + [Tab [Lat, I]]
   end;
  LargeLat := LatSet;
  LargeChar := CharSet;
  for I := Succ (Max) to 21
  do
   begin
    LargeLat := LargeLat + [Tab [Lat, I]];
    LargeChar := LargeChar + [Tab [Lat, I]] + [Tab [Rus, I]]
   end;
(*----------------------ВЫВОД СТЕГАНОГРАММЫ НА ЭКРАН----------------------*)
  if ParamCount = 1
  then
   begin
     while not Eof (F)
     do
     begin
       ReadLn (F, X);
       for I := 1 to Length (X)
       do
        begin
         J := 1;
         C := X [I];
         if C in LargeChar
         then
          begin
            K := K shl 1;
            if C in LargeLat
            then K := K or 1;
            L := Succ (L) mod 8;
            if L = 0
            then Write (Chr (K))
          end;
        end
     end;
     Close (F);
     if WhereX <> 1
     then WriteLn;
     WriteLn ('Ok!');
     Exit
   end;
(*----------------------ВЫВОД СТЕГАНОГРАММЫ В ФАЙЛ----------------------*)
  if ParamCount = 2
  then
   begin
     if Pos (':', ParamStr (2)) <> 2
     then Assign (G, TempName)
     else Assign (G, Copy (ParamStr (2), 1, 2) + TempName);
     Assign (G, TempName);
     Rewrite (G);
     while not Eof (F)
```

```
    do
    begin
      ReadLn (F, X);
      for I := 1 to Length (X)
      do
       begin
        J := 1;
        C := X [I];
        if C in CharSet
        then
         begin
           K := K shl 1;
           if C in LatSet
           then K := K or 1;
           L := Succ (L) mod 8;
           if L = 0
           then
           begin
            Count := Count + 1;
            K := K xor Random (256);
            Write (Chr (K));
            Write (G, Chr (K))
           end
         end;
       end
    end;
    Close (F);
    Close (G);
    Assign (F, ParamStr (2));
(*$I-*)
    Erase (F);
(*$I+*)
    I := IoResult;
    Rename (G, ParamStr (2));
    if WhereX <> 1
    then WriteLn;
    WriteLn ('Прочитано ', Count : 0 : 0, ' байт стега...');
    Exit
  end;
(*----------------------ЗАПИСЬ СТЕГАНОГРАММЫ В ФАЙЛ--------------------*)
  Assign (G, ParamStr (2));
  if Pos (':', ParamStr (3)) <> 2
  then Assign (H, TempName)
  else Assign (H, Copy (ParamStr (3), 1, 2) + TempName);
  Reset (G);
  Rewrite (H);
  while not Eof (F)
  do
   begin
    ReadLn (F, X);
    for I := 1 to Length (X)
    do
     begin
      J := 1;
      C := X [I];
      while (J <= Max) and (C <> Tab [Rus, J])
           and (C <> Tab [Lat, J])
      do J := Succ (J);
      if J <= Max
      then
        begin
         if L = 0
         then
          begin
           if Eof (G)
           then CC := #00
           else Read (G, CC);
```

```
          Count := Count + 1;
          Write (CC);
          K := Ord (CC) xor Random (256)
         end;
        if K and $80 <> 0
        then X [I] := Tab [Lat, J]
        else X [I] := Tab [Rus, J];
        K := K shl 1;
        L := Succ (L) mod 8
       end
     end;
    WriteLn (H, X)
   end;
  Close (F);
  Close (G);
  Close (H);
  Assign (F, ParamStr (3));
(*$I-*)
  Erase (F);
(*$I+*)
  I := IoResult;
  Rename (H, ParamStr (3));
  if WhereX <> 1
  then WriteLn;
  if (Count <> 0) and (L <> 0)
  then Count := Count - 1;
  WriteLn ('Записано ', Count : 0 : 0, ' байт стега...');
  Exit
end.
```

Метод двійкових нулів

```
program StegoZero;
type
  StringType = string [$FF];
const
  TempName = '$$$$$$$$.$$$';
  Key1  = $1234;
  Key2  = $4567;
var
  F, G, H : text;
  Line  : StringType;
  I, K, L, Z : byte;
  C   : char;
  Count  : real;
begin
  LowVideo;
  if not (ParamCount in [1..3])
  then
   begin
    WriteLn ('Ожидается от 1 до 3 параметров:');
    WriteLn ('1 - вывод стеганограммы на экран;');
    WriteLn ('2 - вывод стеганограммы в файл;');
    WriteLn ('3 - запись стеганограммы в файл.');
    Exit
   end;
  Assign (F, ParamStr (1));
  Reset (F);
  Count := 0;
  MemW [Dseg : $01FE] := Key1;
  MemW [Dseg : $01FC] := Key2;
(*---------------------ВЫВОД СТЕГАНОГРАММЫ НА ЭКРАН---------------------*)
  if ParamCount = 1
  then
   begin
    L := 0;
```

```
    Z := 0;
    while not Eof (F)
    do
    begin
      ReadLn (F, Line);
      while (Line <> '')
      and (Line [1] <= ' ')
      do Delete (Line, 1, 1);
      while (Line <> '')
      and (Line [Length (Line)] <= ' ')
      do Delete (Line, Length (Line), 1);
      while Line <> ''
      do
       begin
        while (Line <> '') and (Line [1] > ' ')
        do Delete (Line, 1, 1);
        if (Length (Line) > 1)
        and ((Copy (Line, 1, 2) = #00' ')
         or (Copy (Line, 1, 2) = ' '))
        then
         begin
          Z := Z shr 1;
          if Line [1] = #00
          then Z := Z or $80;
          L := Succ (L) mod 8;
          if L = 0
          then Write (Chr (Z))
         end;
        while (Line <> '') and (Line [1] <= ' ')
        do Delete (Line, 1, 1)
       end
    end;
    Close (F);
    if WhereX <> 1
    then WriteLn;
    WriteLn ('Ok!');
    Exit
  end;
(*----------------------ВЫВОД СТЕГАНОГРАММЫ В ФАЙЛ----------------------*)
  if ParamCount = 2
  then
   begin
    if Pos (':', ParamStr (2)) <> 2
    then Assign (G, TempName)
    else Assign (G, Copy (ParamStr (2), 1, 2) + TempName);
    Rewrite (G);
    L := 0;
    Z := 0;
    while not Eof (F)
    do
    begin
      ReadLn (F, Line);
      while (Line <> '')
      and (Line [1] <= ' ')
      do Delete (Line, 1, 1);
      while (Line <> '')
      and (Line [Length (Line)] <= ' ')
      do Delete (Line, Length (Line), 1);
      while Line <> ''
      do
       begin
        while (Line <> '') and (Line [1] > ' ')
        do Delete (Line, 1, 1);
        if (Length (Line) > 1)
        and ((Copy (Line, 1, 2) = #00' ')
         or (Copy (Line, 1, 2) = ' '))
        then
```

```
        begin
          Z := Z shr 1;
          if Line [1] = #00
          then Z := Z or $80;
          L := Succ (L) mod 8;
          if L = 0
          then
           begin
             Count := Count + 1;
             C := Chr (Z
             xor Random (256));
             Write (G, C);
             Write (C)
           end
         end;
        while (Line <> '') and (Line [1] <= ' ')
        do Delete (Line, 1, 1)
       end
    end;
    Close (F);
    Close (G);
    Assign (F, ParamStr (2));
(*$I-*)
    Erase (F);
(*$I+*)
    I := IoResult;
    Rename (G, ParamStr (2));
    if WhereX <> 1
    then WriteLn;
    WriteLn ('Прочитано ', Count : 0 : 0, ' байт стега...');
    Exit
  end;
(*-----------------------ЗАПИСЬ СТЕГАНОГРАММЫ В
ФАЙЛ---------------------*)
  Assign (G, ParamStr (2));
  if Pos (':', ParamStr (3)) <> 2
  then Assign (H, TempName)
  else Assign (H, Copy (ParamStr (3), 1, 2) +
TempName);
  Reset (G);
  Rewrite (H);
  L := 0;
  while not Eof (F)
  do
   begin
    ReadLn (F, Line);
    I := 0; (* последний пробельный символ *)
    while (I < Length (Line)) and (Line [Succ (I)] <= ' ')
    do I := Succ (I);
    K := Length (Line); (* последний непробельный
символ *)
    while (K > I) and (Line [K] <= ' ')
    do K := Pred (K);
    while I < K
    do
     begin
      while (I < K) and (Line [Succ (I)] > ' ')
      do I := Succ (I);
      if (K - I > 2) and (Copy (Line, Succ (I), 2) = ' ')
      then
       begin
        if L = 0
        then
         begin
           if Eof (G)
           then C := #00
           else Read (G, C);
           Z := Ord (C) xor Random (256)
         end;
        if Z and 1 = 1
        then Line [Succ (I)] := #00;
        Z := Z shr 1;
        L := Succ (L) mod 8;
        I := I + 2;
        if L = 0
        then
         begin
           Write (C);
           Count := Count + 1
         end
       end;
      while (I < K) and (Line [Succ (I)] = ' ')
      do I := Succ (I)
     end;
    WriteLn (H, Line)
   end;
  Close (F);
  Close (G);
  Close (H);
  Assign (F, ParamStr (3));
(*$I-*)
  Erase (F);
(*$I+*)
  I := IoResult;
  Rename (H, ParamStr (3));
  if WhereX <> 1
  then WriteLn;
  WriteLn ('Записано ', Count : 0 : 0, ' байт
стега...');
  Exit
end.
```

Зміст

НАВЧАЛЬНЕ ВИДАННЯ

Кузнецов Олександр Олександрович
Євсеєв Сергій Петрович
Король Ольга Григорівна

СТЕГАНОГРАФІЯ

Навчальний посібник

Відповідальний за випуск **Пономаренко В. С.**

Відповідальний редактор **Сєдова Л. М.**

Редактор **Семенова І. М.**

Коректор **Мартовицька-Максимова В. А.**

План 2011 р. Поз. № 62-П.
Підп. до друку Формат 60 х 90 1/16. Папір MultiCopy. Друк Riso.
Ум.-друк. арк. 14,5. Обл.-вид. арк. 18,13. Тираж прим. Зам. №

Видавець і виготівник — видавництво ХНЕУ, 61001, м. Харків, пр. Леніна, 9а

Свідоцтво про внесення до Державного реєстру суб'єктів видавничої справи ***Дк № 481 від 13.06.2001 р.***