МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КиївськийНаціональнийуніверситеттехнологій та дизайну
Факультет ринкових, інформаційних та інноваційнихтехнологій
Кафедра інформаційно-комп'ютернихтехнологій
та фундаментальнихдисциплін

Дипломна магістерська робота

на тему:

«ДОСЛІДЖЕННЯ МОЖЛИВОСТЕЙ ЗАСТОСУВАННЯ КРИПТОГРАФІЧНИХ МЕТОДІВ ЗАХИСТУ В ІНФОРМАЦІЙНИХ МЕРЕЖАХ»

Виконав: студент групи МгЧКІ-20
спеціальності 123 Комп'ютерна інженерія
освітньої програми Комп'ютерна інженерія
ГУРОМА Роман
Керівник: к.т.н., доцент
ОДОКІЄНКО Світлана
Рецензент: _____

Черкаси 2021

ЗМІСТ

ВСТУП

Останнім часом активно обговорюється проблема захисту інформаційних мереж. Зокрема звертається увага на відсутність своєчасного втручання в випадках, коли надійність передачі даних та їх вміст знаходиться під загрозою. Втручання людини може бути як несвоєчасним, так і неефективним. Також не слід забувати про людський фактор, халатність, некомпетентність персоналу, який може звести нанівець технічний потенціал навіть самих передових систем захисту інформації. Щоб вберегти непосвячену людину від некоректних дій з інформаційними технологіями можна використати вчення про захист інформації, що називається криптографією.

Криптографія – галузь науки, яка вивчає математичні методи та технології забезпеченняавтентичності, цілісності та конфіденційності інформації. Була зумовлена практичною потребоюпередавати важливу інформацію більш надійним способом. Для математичного аналізу криптографія використовує інструментарій абстрактної алгебри та теорії ймовірностей. Захист досягається шифруванням, тобто перетворенням, які роблять захищені вхідні дані важкорозкриваними за вхідними даними без знання спеціальної ключової інформації ключа. Під ключем розуміється легкозмінна частина криптосистеми, що зберігається в таємниці і визначає, яке шифрувальні перетворення з можливих виконується в даному випадку. Криптосистема родина обраних за допомогою ключа оборотних перетворень, які перетворюють захищений відкритий текст в шифрограму і назад.

Проблема захисту інформації шляхом її перетворення, що виключає її прочитання сторонньою особою, хвилювала людський розум з давніх часів. Історія криптографії ровесниця історії людського мовлення та спілкування. Як приклад, спочатку писемність сама по собі була криптографічного технологією, оскільки в древніх суспільствах нею оволоділи лише певні слої населення. Безліч книг Стародавньої Індії та Стародавнього Єгипту тому приклади.

Різні люди розуміють під шифруванням різні речі. Діти грають в іграшкові шифри й таємні мови. Це, однак, не має нічого спільного зі справжньою криптографією. Справжня криптографія повинна забезпечувати такий рівень

секретності, щоб можна було надійно захистити критичну інформацію від розшифровки великими організаціями такими, як мафія, транснаціональні корпорації і великі держави. Однак зараз, зі становленням інформаційного суспільства, вона стає центральним інструментом для забезпечення конфіденційності [1].

Проблема використання криптографічних методів в даний момент особливо актуальна, тому що:

- розширилося використання комп'ютерних мереж, зокрема мережі Інтернет;
- створення нових електронно обчислювальних пристроїв, нейронних обчислень і систем комунікаційпривело до того, що стала можливою дискредитація криптосистем.

Актуальність роботи. На сучасному етапі розвитку суспільства однією з найбільших цінностей стала інформація. Важливим питанням на сьогоднішній день є захист інформації. Існує декілька напрямків та підходів до захисту інформації, і один із них це криптологія, тобто захист математичними методами. Розвиток електронно-обчислювальної техніки та математичного апарату дає зловмисникам все більше можливостей для дешифрування інформації. З плином часу до алгоритмів шифрування висуваються все більш суворі вимоги, а кожний новий запропонований метод перевіряється все більш детально на велику кількість вразливостей.

Мета завдання. Метою даної магістерської роботи є дослідження можливості застосування криптографічних методів захисту в інформаційних мережах, їх різноманітність, переваги та недоліки, приклади використання.

Об’єкт дослідження. Об`єктом дослідження є криптографічні методи захисту в інформаційних мережах.

Предмет дослідження. Предметом дослідження є практичне використання криптографічних алгоритмів для захисту інформації.

Наукова новизна. Наукова новизна одержаних результатів полягає у використанні криптографічних алгоритмів з відкритим ключем, в поєднанні з

SSL/TLS протоколами, для захисту інформації в мережі.

Апробація результатів магістерської роботи. Основні положення магістерського дослідження доповідались та обговорювались на XVII студентській науковій конференції «Інформаційні інновації сучасного українського суспільства», напрям Інформаційно-комп'ютерний, жовтень 2021 р.

Публікації. За результатами магістерської роботи було опубліковано статтю, яка відображає основні результати роботи.

РОЗДІЛ 1
АНАЛІТИЧНИЙ ОГЛЯД

Для сучасного комп’ютерного співтовариствакриптоатаки стали повсякденністю. Зловмисниками використовуються якметоди соціальної інженерії для отримання необхідної інформації, так і помилки в адмініструванні та програмуванні.

Всі засоби масової інформації останнім часом наповнено повідомленнями про атаки на інформацію, про хакерів і комп'ютернівзломи. Дати визначення цього явищадійсно дуже складно, так як інформацію, особливо в електронному вигляді, представлено безліччю різних форм. За інформацію можна вважати як базу данних, так і окремий файл, повноцінний програмний комплекс, як і одиничний запис в базі даних.Потрібно розуміти, що всі ці об'єкти можуть піддатися і піддаються атакам зловмисників [2].

В процесі зберігання, підтримки і надання доступу до будь-якого інформаційного об'єкту його власник, або уповноважена ним особа, накладає явно або самоочевидний набір політик та правил по роботі з нею. Атакою на інформацію класифікується умисне їх порушення.

В зв’язку з масовим впровадженням електронно обчислювальних машин (комп’ютерів, смартфонів, мережевих інтерфейсів) в усі сфери людської діяльностікількість інформації, котра зберігається в цифровому вигляді виріс в сотні та тисячі разів. І тепер скопіювати за секунду і віднести флешку з базою даних, що містить комерційну таємницю, в рази простіше, ніж копіювати або перезаписувати безліч паперів. А з появою комп'ютерних та телекомунікаційних мереж навіть відсутність фізичного доступу до комп'ютерної системи перестало бути гарантією збереження конфіденційності та безпеки інформації.

Які можуть бути наслідки інформаційних атак? В першу чергу, звісно, більшість будуть зацікавлені економічними втратами:

- До серйозних прямих збитків на ринку може призвести в першу чергу розкриття комерційної інформації;
- Новина в засобах масової інформації про викрадення великого обсягу

інформації частіше за все серйозно впливає на репутацію фірми, в найгірших випадках приводячи до втрат в обсягах торгових операцій, або навіть до закритя, банкротства;

- Скористатися крадіжкою інформації можуть фірми-конкуренти, якщо ті залишилася непоміченеми, для того щоб повністю збанкротувати фірму, нав'язуючи їй фальшиві або завідомо невигодні угоди;
- Підміна даних як на етапі передачі, так і на етапі зберігання в компанії може призвести до величезних фінансових втрат;
- Багаторазово успішні атаки на компанію, котра надає будь-який видпрограмних послуг, знижують довіру до компанії у клієнтів, що позначається на рівні доходів;
- А також комп’ютерними атаками може бути нанесено величезний моральних збиток.

1.1 Криптографія

Криптографія наука про математичні методи забезпечення конфіденційності (неможливості прочитання інформації стороннім) і автентичності (цілісності і справжності авторства, а також неможливості відмови від авторства) інформації.

З широким поширенням писемності криптографія стала формуватися як самостійна наука. Перші криптосистеми зустрічаються вже на початку нашої ери. Бурхливий розвиток криптографічні системи отримали в роки першої і другої світових воєн. Починаючи з післявоєнного часу і по нинішній день, поява обчислювальних засобів прискорило розробку та вдосконалення криптографічних методів [3].

Чому проблема використання криптографічних методів в інформаційних системах стала зараз особливо актуальною?

З одного боку, розширилося використання комп'ютерних мереж, зокрема глобальної мережі Інтернет, по яких передаються великі обсяги інформації державного, військового, комерційного і приватного характеру, не допускає можливість доступу до неї сторонніх осіб.

З іншого боку, поява нових потужних комп'ютерів, технологій мережевих і

нейронних обчислень зробило можливим дискредитацію криптографічних систем ще недавно вважалися практично нерозкривними. Проблемою захисту інформації шляхом її перетворення займається криптологія.

Криптологія розділяється на два напрямки криптографію і криптоаналіз. Мета цих напрямків прямопротилежна.Криптографія займається пошуком і дослідженням математичних методів перетворення інформації. Сфера інтересів криптоаналізу дослідження можливості розшифровки інформації без знання ключів.

Повідомлення, яке ви хочете передати адресату, називається **відкритим текстом**. Для збереження повідомлення в таємниці воно перетворюється криптографічними методами і тільки після цього передається адресату. Перетворене повідомлення будемо називати шифрованим шифртекст, а сам процес перетворення шифрування. Параметр, котрий визначає правило шифрування називається ключем.

Все різноманіття існуючих криптографічних методів можна звести до наступних класів перетворень:

- **Моно і багатоалфавітної підстановки**. Найбільш простий вид перетворень, що полягає в заміні символів вихідного тексту на інші (того ж алфавіту) за більш-менш складним правилом. Для забезпечення високої криптостійкості потрібне використання великих ключів.
- **Перестановки.** Також нескладний метод криптографічного перетворення. Використовується, як правило, в поєднанні з іншими методами.
- **Гаммування**. Цей метод полягає в накладенні на вихідний текст деякої псевдовипадкової послідовності, що генерується на основі ключа.

Блокові шифри представляють собою послідовність (з можливим повторенням і чергуванням) основних методів перетворення, застосовується до блоку (частини) шифрованого тексту. Блокові шифри на практиці зустрічаються частіше, ніж "чисті" перетворення того чи іншого класу в силу їх більш високої криптостійкості. Російський і американський стандарти шифрування засновані саме на цьому класі шифрів [3].

1.2 Історія криптографії

За наукоподібним словом «криптографія» (з давньогрецької буквально - «тайнопис») ховається стародавнє бажання людини сховати важливу інформацію від сторонніх очей. Можна сказати, що сама писемність на самому початку вже була криптографічного системою, так як належала вузькому колу людей, і за допомогою неї вони могли обмінюватися знаннями, недоступними неписьменним. З поширенням писемності виникла потреба в більш складних системах шифрування. З часів стародавніх цивілізацій криптографія вірно служила військовим, чиновникам, купцям і зберігачам релігійних знань.

Найдавнішим свідченням застосування шифру **(близько 4000 р. до н.е.)** вчені вважають давньоєгипетський папірус з перерахуванням монументів часів фараона Аменемхета II. Безіменний автор видозмінив відомі ієрогліфи, але, швидше за все, не для приховування інформації, а для більш сильного впливу на читача [4].

Ще один відомий шифр давньосемітський атбаш, **приблизно 600 р. до н.е**. Тут інформацію заплутували найпростішим способом за допомогою підміни букв алфавіту. Криптограми на атбаш зустрічаються в Біблії.

У Стародавній Спарті користувалися скіталу шифром з циліндра, що обвивається смужками пергаменту. Текст писали в рядок на пергаменті. Після розмотування стрічки текст перетворювався в шифр, прочитати який було можливо, тільки маючи циліндр такого ж діаметру. Можна сказати, що спартанська скитала стала одним з перших криптографічних пристроїв.

У **IV столітті до н.е.** автор військових трактатів Еней Тактик придумав шифрувальний диск, названий згодом його ім'ям. Щоб створити в отвори диска з підписаними поруч з ними буквами послідовно просмикували нитку. Щоб прочитати текст, потрібно було всього лише витягати нитку в зворотній послідовності. Це і становило основний мінус пристрою при наявності часу шифр міг бути розгаданий будь-якою грамотною людиною. Зате, щоб швидко «стерти» інформацію з диска Енея, потрібно було всього лише витягнути нитку або розбити пристрій.

Одним з перших документально зафіксованих шифрів є **шифр Цезаря** (близько 100 р до н.е.). Його принцип був дуже простий: кожна літера вихідного тексту замінювалася на іншу, віддалену від неї за алфавітом на певне число позицій. Знаючи це число, можна було розгадати шифр і дізнатися, які таємниці Цезар передавав своїм генералам.

Шифруванням користувалися багато древніх народів, але особливого успіху в криптографії вже в нашу еру досягли арабські вчені. Високий рівень розвитку математики та лінгвістики дозволив арабам не тільки створювати свої шифри, але і займатися розшифровкою чужих. Це призвело до появи перших наукових робіт з криптоаналіз дешифрування повідомлень без знання ключа. Епоха так званої наївної криптографії, коли шифри були більше схожі на загадки, підійшла до кінця.

Роботи арабських вчених сприяли появі поліалфавітних шифрів, більш стійких до розшифровки, в яких використовувалися відразу кілька алфавітів. Однак люди **Середньовіччя** продовжували користуватися простими шифрами, заснованими на заміну букв іншими буквами або цифрами, неправильному написанні букв і т.д. В середні віки в Європі вважалося, що криптографія була тісно пов'язана з магією і каббалой.

Цікаво, що в **Стародавній Русі** теж були свої способи тайнопису, наприклад літорея, яка ділилася на просту і мудру. У мудрої версії шифру деякі букви замінялися точками, палицями або колами. У простій Літорії, яка ще називалася Тарабарською грамотою, все згодні літери кирилиці розташовувалися в два ряди. Зашифровували лист, замінюючи букви одного ряду буквами іншого.

Ще одним відомим шифром Стародавньої Русі була цифра, коли літери, склади і слова замінялися цифрами. Іноді для ускладнення в шифр додавалися математичні дії, і було непросто розгадати цю загадку: «Десятеріца сугуба і Пятеріца четверіци, одиниця четверіци суто і десятеріца дващі».

В **епоху Відродження** криптографія переживає підйом. Починається період формальної криптографії, пов'язаний з появою формалізованих, більш надійних шифрів. Над деякими загадками вчених Ренесансу криптографи наступних років

билися століттями [4].

Близько **1466 року** італійський вчений Леон Альберті винаходить шифрувальний диск, що складається з двох частин: зовнішньої і внутрішньої. На нерухомому зовнішньому диску був написаний алфавіт і цифри. Внутрішній рухливий диск також містив букви і цифри в іншому порядку і був ключем до шифру. Для шифрування потрібно було знайти потрібну букву тексту на зовнішньому диску і замінити її на букву на внутрішньому, що стоїть під нею. Після цього внутрішній диск зміщувався, і нова буква зашифровувалася вже з нової позиції. Таким чином, шифр Альберті став одним з перших шифрів багатоалфавітної заміни, заснованому на принципу комбінаторики. Крім того, Леон Альберті написав одну з перших наукових робіт по криптографії «Трактат про шифри».

Тут варто згадати таке явище, як **стеганографія**, якому в роботі Альберті також було приділено увагу. Якщо за допомогою шифру намагаються приховати сенс інформації, то стеганографія дозволяє приховати сам факт передачі або зберігання даних. Тобто текст, захований за допомогою цього методу, ви будете вважати за картинку, кулінарний рецепт, список покупок або, наприклад, кросворд. Або взагалі не побачите його, якщо він буде написаний молоком, лимонним соком або з допомогою особливих чорнил. Часто методи стеганографії і криптографії об'єднувалися в одному посланні.

Проривом в криптографії стала книга «Поліграфія» абата Іоганеса Трітемія **1518 року**, що розповідає в тому числі про шифри з поліалфавітною заміною. Найвідомішим шифрувальником XVI століття вважається дипломат і алхімік з Франції Блез де Віженер, який придумав абсолютно стійкий шифр, в якому використовувалося 26 алфавітів, а порядок використання шифру визначався знанням пароля. Можна сказати, що шифр Віженера представляв собою комбінацію декількох вже згадуваних шифрів Цезаря.

Промислова революція не обійшла увагою і криптографію. Близько 1790 року один з батьків засновників США Томас Джефферсон створив дисковий шифр, прозваний пізніше циліндром Джефферсона. Цей прилад, заснований на

роторній системі, дозволив автоматизувати процес шифрування і став першим криптопристроєм Нового часу.

Великий вплив на шифрувальну справу зробив винахід телеграфу. Колишні шифри вмить перестали працювати, при цьому потреба в якісному шифруванні тільки зростала в зв'язку з низкою великих військових конфліктів. У XIX-XX століттях основні імпульси для розвитку криптографії давала саме військова сфера. З **1854року** британські військові застосовують шифр Плейфера, в основі якого шифрування біграм, або пар символів. Цей шифр використовувався до початку Другої світової війни.

У **Другій світовій війні** противники вже використовували мобільні електромеханічні шифратори, шифри яких вважалися нерозкривними. Пристрої були роторними або на цевочних дисках. До перших відносилася знаменита машина

«Енігма», якою користувалися нацисти, до других американська машина М-209.

Принцип роботи «Енігми» полягав в наступному: при кожному натисканні на клавішу з літерою алфавіту в рух приходили один або кілька роторів. Буква змінювалася кілька разів за принципом шифру Цезаря, і в віконці видавався результат. Шифри «Енігми» вважалися найбільш стійкими для злому, так як кількість її комбінацій досягало 15 квадрильйонів. Однак код «Енігми» все ж був розшифрований, спершу польськими криптографами в 1932 році, а потім англійським вченим Аланом Тьюрингом, який створив машину для розшифровки повідомлень «Енігми» під назвою «Бомба». Комплекс з 210 таких машин дозволяв англійцям розшифровувати до 3 тис. Військових сполучень нацистів на добу і вніс великий вклад в перемогу союзників.

Рисунок 1.1 – переносна шифрувальна машина «Енігма»

У 1949 році Клод Шеннон пише роботу «Теорія зв'язку в секретних системах», і криптографія остаточно переходить в сферу математики. До кінця 1960-х роторні шифрувальні системи замінюються більш досконалими блоковими, які припускали обов'язкове застосування цифрових електронних пристроїв. У 1967 році вчений Девід Кан видав популярну книгу «Зломщики кодів», яка викликала великий інтерес до криптографії.

З поширенням комп'ютерів криптографія виходить на новий рівень. Потужності нових пристроїв дозволяють створювати на порядки більш складні шифри. Шифр або код стає мовою спілкування між комп'ютерами, а криптографія стає повноцінною цивільною галуззю. У 1978 році розробляється стандарт шифрування DES, який став основою для багатьох сучасних криптографічних алгоритмів.

Сфера використання криптографії розширюється, при цьому влада різних країн намагаються утримати контроль над використанням шифрів. Розробки криптографів засекречуються, від виробників шифрувальних машин вимагають залишати в продуктах "чорні ходи" для доступу спецслужб.

Паралельно незалежні криптоаналітики розробляють способи шифрування,

якими могли б користуватися всі бажаючі так звану відкриту криптографію. Особливо актуально це стало з розвитком інтернету, де питання конфіденційності інформації встало дуже гостро. Першою криптосистемою з відкритим ключем вважається створений в 1977 році алгоритм RSA, назва якого є акронімом імен творців Ріверст, Шаміра і Адельмана. А в 1991 році американський програміст Філіп Циммерман розробляє популярний пакет PGP з відкритим вихідним кодом для шифрування електронної пошти [4].

Поширення доступного інтернету по всьому світу неможливо уявити без криптографії. З появою месенджерів, соціальних мереж, онлайн-магазинів і сайтів державних послуг передача персональної інформації в мережі відбувається без зупинки і у величезних кількостях. Сьогодні ми стикаємося з криптографією щодня, коли вводимо пароль від поштового сервісу, дізнаємося статус покупки онлайн або робимо грошовий переказ через додаток банку. Криптографія пройшла гігантський шлях від простих шифрів давнини до складних криптосистем. Майбутнє цієї науки твориться на наших очах чергова революція в шифруванні відбудеться з появою квантових суперкомп'ютерів, розробка яких вже ведеться.

1.3 Криптографія і держава

Починаючи з 1970-х років інтерес до криптографії зростає з боку окремих дослідників, бізнесу і приватних осіб. Цьому сприяли в тому числі і публікації у відкритій пресі книга Девіда Кана «Зломщики кодів», готовність наукової (створення осередку Фейстеля, роботи Діффі і Хеллмана, шифрів DES і RSA) і технічної бази (обчислювальної техніки), а також наявність «замовлення» з боку бізнесу вимог до надійної передачі інформації в рамках окремої країни і по всьому світу. Одночасно з цим з'явилося й відвертий спротив з боку держави розвитку відкритої криптографії (цивільної криптографії), що видно на прикладі історії протидії з АНБ. Серед причин негативного ставлення уряду вказують на неприпустимість потрапляння надійних систем шифрування в руки терористів, організованої злочинності або ворожої розвідки [5].

Після зростання суспільного інтересу до криптографії в США в кінці 1970-х

і початку 1980-х років призвело АНБ до низки спроб придушити інтерес суспільства до криптографії. Якщо з компанією IBM вдалося домовитися (в тому числі з питання зниження криптостійкості шифру DES), то наукове співтовариство довелося контролювати через систему грантів Національний науковий фонд США. Представники фонду погодилися направляти роботи по криптографії на перевірку в АНБ і відмовляти у фінансуванні певних наукових напрямів. Також АНБ контролювала і бюро патентів, що дозволяло накласти гриф секретності в тому числі на винаходи цивільних осіб. Так, в 1978 році гриф «секретно», відповідно до закону Invention Secrecy Act про засекречування винаходів, які могли бути використані для вдосконалення техніки військового призначення, отримало винахід «Phaserphone» групи під керівництвом Карла Ніколаї, що дозволяє шифрувати голос. Після того як історія отримала значний розголос в пресі, АНБ довелося відмовитися від спроб засекретити і монополізувати винахід. Також в 1978 році цивільний співробітник АНБ Джозеф Мейер без узгодження з начальством послав в IEEE, членом якого він також був, лист з попередженням, що публікація матеріалів щодо шифрування і криптоаналіз порушує Правила з регулювання міжнародного трафіку озброєнь. Хоча Мейер виступав як приватна особа, лист було розцінено як спроба АНБ припинити цивільні дослідження в області криптографії. Проте його точка зору не знайшла підтримки, але саме обговорення створило рекламу як відкритої криптографії, так і симпозіуму по теорії інформації 1977 року науки, тісно пов'язаної з шифруванням такриптоаналізу завдяки роботам Шеннона.

Після провалів, пов'язаних з листом Мейєра і справи групи Ніколаі, директор АНБ опублікував кілька статей, в яких закликав академічні кола до спільного вирішення проблем, пов'язаних з відкритим вивченням криптографії та національною безпекою. В результаті утворилася деяка структура самоцензури попередньої перевірки наукових публікацій в особливому державному комітеті. У той же час АНБ отримує можливість розподіляти кошти на криптографічні дослідження, «відокремивши» від Національного наукового фонду свій власний, в 2-3 мільйони доларів США. Проте після конфлікту з Леонардо Адлеманом в 1980

році було вирішено, що заявку на фінансування криптографічних досліджень можна подавати як в національний, так і в спеціалізований фонд АНБ.

Законодавчо в США було зроблено обмеження на використання відкритої криптографії. Висувалася вимога навмисне забезпечити ослаблений захист від злому, щоб державні служби при необхідності (в тому числі за рішенням суду) могли прочитати або прослухати зашифровані повідомлення. Однак через декілька інцидентів злому комерційних систем від цього довелося відмовитися, так як заборона на використання сильної криптографії всередині країни став завдавати шкоди економіці. В результаті до кінця 1980-х років в США залишилася єдина заборона на експорт «сильної» криптографії, в результаті якого, а також через розвиток персональної обчислювальної техніки, до початку 1990-х років вся експортована з США криптографія стала «повністю слабкою».

Проте АНБ і ФБР кілька разів піднімали питання про заборону або дозвільний механізм для приватних компаній займатися роботами в області криптографії, але ці ініціативи завжди зустрічали опір суспільства і бізнесу. На даний момент можна сказати, що зараз АНБ відмовилася від усіх претензій і вважає за краще виступати експертною стороною. До цього (а ФБР і до сих пір) кілька разів змінювало свою позицію, пропонуючи різні схеми використання сильної криптографії в бізнесі та приватним особам [5].

У 1991 році законопроект № 266 включив в себе необов'язкові вимоги, які, якби вони були прийняті, змусили б усіх виробників захищеного телекомунікаційного обладнання залишати "чорні ходи", які б дозволили уряду отримувати доступ до незашифрованих повідомлень. Ще до того як законопроект провалився, Філіп Циммерман виклав в Інтернет PGP пакет безкоштовного програмного забезпечення з відкритим кодом для шифрування і електронного підпису повідомлень. Спочатку він планував випустити комерційну версію, але ініціатива уряду щодо просування законопроекту спонукала його випустити програму безкоштовно. У зв'язку з цим проти Циммермана було порушено кримінальну справу за «експорт озброєнь», яке було припинено тільки в 1996 році, коли світ побачила вже 4-а версія програми.

Наступною ініціативою став проект Clipper Chip, Запропонований в 1993 році. Чіп містив сильний, згідно із заявою АНБ, алгоритм шифрування Skipjack, який тим не менше дозволяв третій стороні (тобто уряду США) отримати доступ до закритого ключа і прочитати зашифроване повідомлення. Даний чіп пропонувалося використовувати як основу для захищених телефонів різних виробників. Однак дана ініціатива не була прийнята бізнесом, який вже мав досить сильні і відкриті програми на кшталт PGP. У 1998 році шифр був розсекречений, після чого Біхам, Шамір і Бірюков протягом одного дня справили успішні атаки на варіант шифру з 31 раундом (з 32-х).

Проте ідея депонування ключів поширювалася. У Великобританії її намагалися впровадити кілька років, а у Франції вона почала діяти в 1996 році. Однак, незважаючи на значні зусилля США і, зокрема, її уповноваженого по криптографії Девіда Аарона, багато країн, в тому числі ті, що входять в Організацію економічного співробітництва та розвитку, в цілому відмовилися від цієї ідеї на користь захисту недоторканності приватного життя. Також експерти (наприклад, в доповіді Європейської Комісії «До європейської інфраструктури цифрових підписів та криптографії» COM (97) 503) відзначили наявність безлічі невирішених проблем, пов'язаних зі структурою централізованого депонування ключів, в тому числі: зниження загальної захищеності системи, потенційно високу вартість і потенційну легкість обману з боку користувачів. Останнє легко пояснити на прикладі системи Clipper, коли користувач мав можливість згенерувати неправдиву інформацію для відновлення ключа (разом з коротким хеш-кодом) так, що система працювала без технічної можливості відновити ключ третьою стороною. У грудні 1998 року на засіданні учасниць Вассенаарської угоди США спробували отримати послаблення в правилах експорту для систем з депонуванням ключів, проте сторони не дійшли згоди. Це можна назвати датою остаточної поразки подібних систем на сьогоднішній день. Після цього Франція в січні 1999 року оголосила про відмову від системи депонування ключів. Тайвань, в 1997 році оголосив про плани по створенню такої системи, також відмовився від них в 1998 році. В Іспанії, незважаючи на передбачену можливість депонування

ключів в прийнятому в 1998 році законі про телекомунікації, система так і не запрацювала [5].

Після відмови від технічних засобів доступу до ключів уряди звернулися до ідеї законодавчого регулювання даного питання коли людина сама зобов'язується надати заздалегідь або на вимогу ключ для читання повідомлень. Даний варіант можна назвати «законний доступ». У різних країнах до нього ставляться порізному. ОЕСР залишає за своїми членами свободу у використанні або відмову від даного способу. У липні 1997 року на саміті в Денвері учасники «Великої Вісімки» ідею підтримали. У Малайзії та Сінгапурі за відмову надання ключів слідству людині загрожує кримінальне покарання. У Великобританії та Індії схожі закони розглядаються. В Ірландії прийнятий закон з положеннями про розкриття відкритого тексту, а також з рекомендаціями проти насильницького розкриття ключів. У Бельгії, Нідерландах і США розглядаються пропозиції про розкриття відкритого тексту, але з поправками про необов'язковість свідчення проти самого себе. Деякі країни, такі як Данія, відхилили подібну ініціативу.

У 2000 році США зняли практично всі обмеження на експорт криптографічної продукції, за винятком 7 країн з «терористичними режимами». Ще одним кроком до відкритої криптографії став конкурс AES, в якому брали участь вчені всього світу.

1.4 Категорії інформаційної безпеки

Інформація з точки зору інформаційної безпеки класифікується за наступними категоріями:

- конфіденційність гарантія того, що конкретна інформація доступна тільки тому колу осіб, для кого вона призначена; порушення цієї категорії називається розкраданням або розкриттям інформації;
- цілісність гарантія того, що інформація зараз існує в її початковому вигляді, тобто при її зберіганні або передачі не було проведено несанкціонованих змін; порушення цієї категорії називається фальсифікацією повідомлення;
- автентичність гарантія того, що джерелом інформації є саме та особа, яка заявлено як її автор; порушення цієї категорії також називається фальсифікацією,

але вже автора повідомлення;

- апельованість досить складна категорія, але часто застосовується в електронній комерції гарантія того, що при необхідності можна буде довести, що автором повідомлення є саме заявлений людина, і не може бути ніхто інший; відмінність цієї категорії від попередньої в тому, що при підміні автора, хтось інший намагається заявити, що він автор повідомлення, а при порушенні апельованість сам автор намагається "відхреститися" від своїх слів, підписаних ним одного разу [6].

Відносно інформаційних систем застосовуються інші категорії:

- надійність гарантія того, що система поводиться в нормальному і позаштатному режимах так, як заплановано;
- точність гарантія точного і повного виконання всіх команд;
- контроль доступу гарантія того, що різні групи осіб мають різний доступ до інформаційних об'єктів, і ці обмеження доступу постійно виконуються;
- контрольованість гарантія того, що в будь-який момент може бути проведена повноцінна перевірка будь-якого компонента програмного комплексу;
- контроль ідентифікації гарантія того, що клієнт, підключений в даний момент до системи, є саме тим, за кого себе видає;
- стійкість до навмисних збоїв гарантія того, що при навмисному внесенні помилок в межах заздалегідь обговорених норм система буде вести себе так, як обумовлено заздалегідьо.

Висновок до 1 розділу

Аналітичний огляд криптографії, категорії криптографії, її історія та відносини з державою та суспільством. Огляд показав, що ця тема актуальна і сьогодні. Актуальність полягає в необхідності захисту всіх сфер життя від небажаного доступу до інформації.

РОЗДІЛ 2
МЕТОДИ КРИПТОГРАФІЧНОГО ЗАХИСТУ ІНФОРМАЦІЇ

Залежно від наявності або відсутності ключа кодовані алгоритми діляться на тайнопис і криптографію. Залежно від відповідності ключів шифрування і дешифрування на симетричні і асиметричні. Залежно від типу використовуваних перетворень на підстановочні і перестановочні. Залежно від розміру шифрованого блоку на потокові та блокові шифри.

Відносно криптоалгоритмів існує кілька схем класифікації, кожна з яких заснована на групі характерних ознак. Таким чином, один і той же алгоритм "проходить" відразу за кількома схемами, опиняючись в кожній з них в будь-якії з підгруп [7].

Основною схемою класифікації всіх криптоалгоритмів є наступна:

- **Тайнопис.** Відправник і одержувач призводять над повідомленням перетворення, відомі тільки їм двом. Стороннім особам невідомий сам алгоритм шифрування.
- **Криптографія з ключем.** Алгоритм впливу на дані, що передаються, котрий відомий всім стороннім особам, але він залежить від деякого параметра "ключа", яким володіють тільки відправник і одержувач.
- **Симетричні криптоалгоритми.**Для кодування і розшифровки повідомлення використовується один і той же блок інформації (ключ).
- **Асиметричні криптоалгоритми.** Алгоритм такий, що для шифрування повідомлення використовується один ("відкритий") ключ, відомий всім бажаючим, а для розшифровки інший ("закритий"), що існує тільки в одержувача.

Надалі докладніше розглянемо криптографії з ключем, так як більшість фахівців саме по відношенню до цих криптоалгоритм використовують термін криптографія, що цілком виправдано.

Залежно від характеру впливів, вироблених над даними, алгоритми поділяються на:

- **Перестановки.** Блоки інформації (байти, біти, більші одиниці) не змінюються самі по собі, але змінюється їх порядок проходження, що робить

інформацію недоступною сторонньому спостерігачеві.

- **Символи.** Самі блоки інформації змінюються за законами криптоалгоритма. Переважна більшість сучасних алгоритмів належить цій групі.

Залежно від розміру блоку інформації криптоалгоритми діляться на:

- **Потокові шифри.** Одиницею кодування є один біт. Результат кодування не залежить від минулого раніше вхідного потоку. Схема застосовується в системах передачі потоків інформації, тобто в тих випадках, коли передача інформації починається і закінчується в довільні моменти часу і може випадково перериватися. Найбільш поширеними представителей поточних шифрів є скремблери.
- **Блокові шифри.** Одиницею кодування є блок з декількох байтів (в даний час 4-32). Результат кодування залежить від усіх вихідних байтів цього блоку. Схема застосовується при пакетної передачі інформації і кодування файлів.

2.1 Симетричні криптографічні системи

Симетричні шифрування (також симетричні криптосистеми) спосіб шифрування, в якому застосовується один і той же криптографічний ключдля шифрування і дешифрування. До винаходу асиметричного шифрування єдиним існуючим способом було симетричне шифрування. Обидві сторони мають зберігати ключ алгоритмку в секреті. Ключ алгоритму обирається сторонами ще до початку обміну повідомленнями.

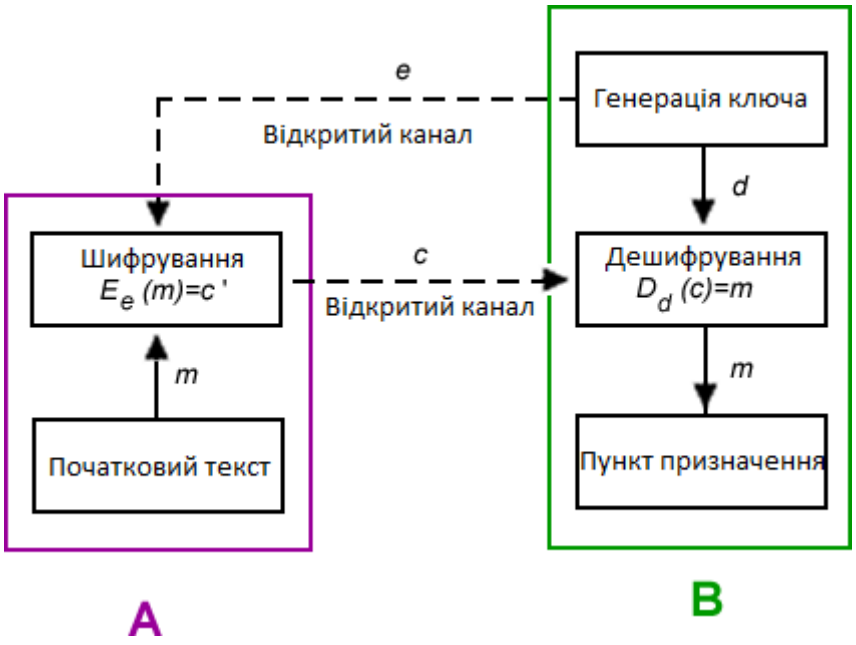

Рисунок 2.1 – Загальна схема семитричної системи шифрування

В даний час симетричні шифри – це поточні та блочні шифри.

Блочний шифр – один із двох видів симетричного шифру. Особливістю блочного шифру є обробка блоку з декількох байт за одну ітерацію (зазвичай 8 або 16). Блочні криптосистеми розбивають текст повідомлення на окремі блоки і потім здійснюють перетворення цих блоків з використанням ключа [8].

Перетворення має використовувати такі принципи:

- Розсіювання (diffusion) тобто зміна будь-якого знака відкритого тексту або ключа впливає на велику кількість знаків шифротекста, котрі приховують статистичні властивості відкритого тексту;
- Перемішування (confusion) використання перетворень, котрі ускладнюють отримання статистичних залежностей між шіфротектстом і відкритим текстом.

До переваг блочних шифрів відносять схожість процедур шифрування і дешифрування, які, як правило, відрізняються лише порядком дій. Це спрощує створення пристроїв шифрування, так як дозволяє використовувати одні і ті ж блоки в ланцюгах шифрування і дешифрування.

Блоковий шифр складається з двох взаємопов'язаних алгоритмів: алгоритм

шифрування *E* і алгоритм дешифрування*E-1*. Вхідними даними служать блок розміром *n* біт і *k*-бітний ключ. На виході виходить *n*-бітний зашифрований блок. Для будь-якого фіксованого ключа функція дешифрування є зворотною до функції шифрування $E^{-1}(E(M)) = M$ для будь-якого блоку *M* і ключа *K*.

Для будь-якого ключа *K, EK* перестановка набору вхідних блоків. Ключ вибирається з *2n!* можливих перестановок.

Розмір блоку *n* це фіксований параметр блокового шифру, зазвичай рівний 64 або 128 бітів, хоча деякі шифри допускають кілька різних значень. Довжина 64 біта була прийнятна до середини 90-х років, потім використовувалася довжина 128 біт і більше. Різні схеми шифрування дозволяють зашифровувати відкритий текст довільної довжини. Кожна має певні характеристики: ймовірність помилки, простота доступу, вразливість до атак. Типовими розміру ключа є 40, 56, 64, 80, 128, 192 і 256 біт. У 2006 році 80-бітний ключ здатний був запобігти атаці грубою силою.

Найпростішим режимом роботи блочного шифру є ECB (Electronic CodeBook (англ.) режим простої заміни або електронної кодової книги), де всі блоки відкритого тексту зашифровуються незалежно один від одного. Однак, при використанні цього режиму статистичні властивості відкритих даних частково зберігаються, так як кожному однаковому блоку даних однозначно відповідає зашифрований блок даних. При великій кількості даних (наприклад, відео або звук) це може призвести до витоку інформації про їх зміст і дати більший простір для криптоаналізу. Видалення статистичних залежностей у відкритому тексті можливо за допомогою попереднього архівування, але воно не вирішує завдання повністю, так як в файлі залишається службова інформація архіватора, що не завжди допустимо [8].

Аналогічна проблема виникає при формуванні та перетворенні «хвостового» блоку даних. Так як дані можуть бути будь-якої довжини, а блоковий шифр вимагає дані кратні довжині блоку, то «хвостовий» блок даних найчастіше доводиться штучно доповнювати до необхідного розміру. При цьому, наприклад, заповнення хвоста однаковими символами полегшує атаку за відомим

відкритому тексту.

Принциповим рішенням описаних проблем є гамування даних і використання контекстно-залежних режимів шифрування.

Поточний шифр це симетричний шифр, в якому кожен символ відкритого тексту перетворюється в символ шифрованого тексту в залежності не тільки від використовуваного ключа, а й від його розташування в потоці відкритого тексту. Поточний шифр реалізує інший підхід до симетричного шифрування, ніж блокові шифри. При блоковому шифруванні відкритий текст розбивається на блоки однакової довжини, при цьому збігаються блоки, при тому що ключі завжди шифрується однаково, при потоковому шифруванні це не так.

У 1949 році Клод Шеннон опублікував роботу, в якій довів абсолютну стійкість шифру Вернама (також відомий, як одноразовий блокнот [one-time pad]). У шифрі Вернама ключ має довжину, рівну довжині самого переданого повідомлення. Ключ використовується в якості гами, і якщо кожен біт ключа вибирається випадково, то криптоаналітику для розтину залишається лише метод грубої сили. Але ключі, які можна порівняти за довжиною з переданими повідомленнями, важко використовувати на практиці. Тому зазвичай застосовують ключ меншої довжини (наприклад, 128 біт). За допомогою нього генерується псевдовипадкова гаммована послідовність. Природно, псевдовипадкова гама може бути використана при атаці на поточний шифр [8].

Типи потокових шифрів:

- При використанні **синхронізованих потокових шифрів** необхідна примусова синхронізація приймального і передавального шифратора (установка ідентичних внутрішніх станів і фази). При втраті бітів або вставці нових бітів відбувається втрата синхронізації і коректне дешифрування стає неможливим. Для синхронізації можуть використовуватися спеціальні марковані послідовності, що вставляються в шифртекст.

- **Самосинхронізовані потокові шифри** використовують спеціально розроблені алгоритми вироблення шифртекста, що дозволяють приймального шифратору отримавши N неспотворених бітів шифртекста автоматично

синхронізуватися з передавальному шифратором. Як наслідок, такі потокові шифри розмножують помилки в каналі зв'язку.

Існує безліч (не менше двох десятків) алгоритмів симетричних шифрів, істотними параметрами яких є:

- стійкість;
- довжина ключа;
- число раундів;
- довжина оброблюваного блоку;
- складність апаратної / програмної реалізації.

Поширені алгоритми:

- DES (Data Encryption Standard, стандарт шифрування даних);
- 3DES (Triple-DES, потрійний DES);
- AES (Advanced Encryption Standard, поліпшений стандарт шифрування);
- RC2 (Шифр Ривеста (Rivest Cipher));
- Blowfish;
- Twofish;
- NUSH;
- IDEA (International Data Encryption Algorithm, міжнародний алгоритм шифрування даних);
- CAST (за ініціалами розробників Carlisle Adams і Stafford Tavares).

Порівняння з асиметричними криптосистемами:

Переваги:

- швидкість (за даними Applied Cryptography на 4 порядки вище);
- простота реалізації (за рахунок більш простих операцій);
- менша необхідна довжина ключа для порівнянної стійкості;
- вивченість (за рахунок більшого віку).

Недоліки:

- складність управління ключами у великій мережі. Значить квадратичне зростання числа пар ключів, які треба генерувати, передавати, зберігати і

знищувати в мережі. Для мережі в 10 абонентів потрібно 45 ключів, для 100 вже 4950, для 1000 499500 і т. д.

- складність обміну ключами. Для застосування необхідно вирішити проблему надійної передачі ключів кожному абоненту, так як потрібен секретний канал для передачі кожного ключа обом сторонам.

Для компенсації недоліків симетричного шифрування в даний час широко застосовується комбінована (гібридна) криптографічний схема, де за допомогою асиметричного шифрування передається сеансовий ключ, який використовується сторонами для обміну даними за допомогою симетричного шифрування [8].

Важливою властивістю симетричних шифрів є **неможливість** їх використання для підтвердження авторства, так як ключ відомий кожній стороні.

2.2 Асиметричні криптографічні системи

Асиметричні криптографічні системи були розроблені в 1970-х рр. Принципова відмінність асиметричної криптосистеми від криптосистеми симетричного шифрування полягає в тому, що для шифрування інформації і її подальшого дешифрування використовуються різні ключі:

- відкритий ключ K використовується для шифрування інформації, обчислюється з секретного ключа до
- секретний ключ k використовується для дешифрування інформації, зашифрованої за допомогою парного йому відкритого ключа K.

Ці ключі розрізняються таким чином, що за допомогою обчислень не можна вивести секретний ключ до з відкритого ключа K. Тому відкритий ключ k може вільно передаватися по каналах зв'язку.

Асиметричні системи називають також двохключовими криптографічними системами, або криптосистемами з відкритим ключем.

Узагальнена схема асиметричної криптосистеми шифрування з відкритим ключем показана на рисунку 2.1.

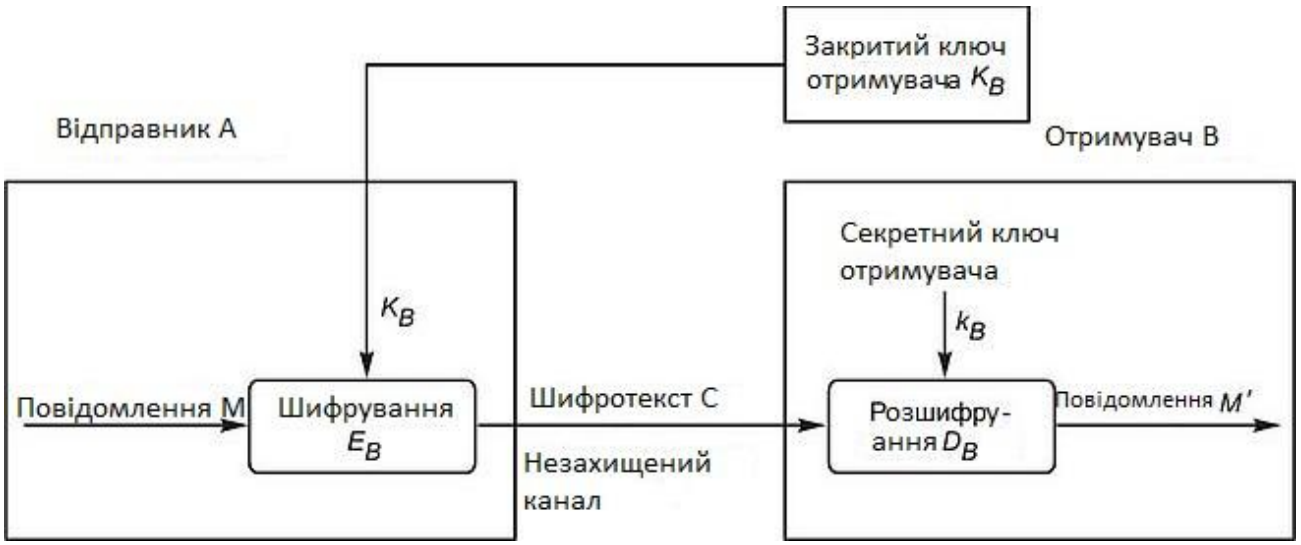

Рисунок 2.2 – Загальна схема асиметричної системи шифрування

Для криптографічного закриття і подальшого дешифрування інформації, що передається використовуються відкритий і секретний ключі одержувача В повідомлення.

Як ключ шифрування повинен використовуватися відкритий ключ одержувача, а в якості ключа дешифрування його секретний ключ.

Секретний і відкритий ключі генеруються попарно. Секретний ключ повинен залишатися у його власника і бути надійно захищений від несанкціонованого доступу (аналогічно ключу шифрування в симетричних алгоритмах). Копія відкритого ключа повинна знаходитися у кожного абонента криптографічної мережі, з яким обмінюється інформацією власник секретного ключа [9].

Процес передачі зашифрованої інформації в асиметричній криптосистемі здійснюється наступним чином.

Підготовчий етап:

- абонент *B* генерує пару ключів: секретний ключ кв і відкритий ключ $K_в$;
- відкритий ключ $K_в$ надсилається абоненту *A* і іншим абонентам (або робиться доступним, наприклад на розділеному ресурсі).

Використання обмін інформацією між абонентами *A* і *B*:

- абонент *А* зашифровує повідомлення за допомогою відкритого ключа $К_в$ абонента *В* і відправляє шифртекст абоненту *В;*

- абонент В розшифровує повідомлення за допомогою свого секретного ключа кв. Ніхто інший (в тому числі абонент *А*) не може розшифрувати дане повідомлення, так як не має секретного ключа абонента *В*. Захист інформації в асиметричній криптосистемі заснований на секретності ключа кв одержувача повідомлення.

Характерні особливості асиметричних криптосистем:

- відкритий ключ $К_в$ і криптограма *С* можуть бути відправлені по незахищених каналах, тобто противнику відомі $К_в$і *С*;

- алгоритми шифрування і дешифрування:

$E_в : M \to C$; $й_в : C \to M$, є відкритими.

У. Діффі і М. Хеллман сформулювали вимоги, виконання яких забезпечує безпеку асиметричної криптосистеми.

1. Обчислення пари ключів ($К_в$, $k_в$) отримувачем В повинно бути простим.

2. Відправник *А*, знаючи відкритий ключ $К_в$і повідомлення *М*, може легко обчислити криптограму

$C = Є_{кв} (M)$.

3. Одержувач *В*, використовуючи секретний ключ кв і криптограму *С*, може легко відновити вихідне повідомлення

$M = П_{кв} (C)$.

4. Противник, знаючи відкритий ключ $К_в$, при спробі обчислити секретний ключ кв наштовхується на непереборну обчислювальну проблему.

5. Противник, знаючи пару ($К_в$, *С*), при спробі обчислити вихідне повідомлення *М* наштовхується на непереборну обчислювальну проблему.

Концепція асиметричних криптографічних систем з відкритим ключем заснована на застосуванні односпрямованих функцій. Односпрямованою функцією називається функція Р (Х), що володіє двома властивостями:

- існує алгоритм обчислення значень функції

$y = P (xy)$;

- не існує ефективного алгоритму звернення (інвертування) функції *У7* (тобто не існує рішення рівняння *Р (Х) = У* відносно *Х*).

Як приклад односпрямованої функції можна вказати цілочисленне множення. Пряма задача обчислення добутку двох дуже великих цілих чисел *Р* і *0*, тобто знаходження значення *N = Р • 3* достатньо нескладне завдання для комп'ютера.

Зворотнє завдання факторизація, або розкладання на множники великого цілого числа, тобто знаходження дільників *Р* і *0* великого цілого числа $A^2 = P?0$, є практично нерозв'язною при досить великих значеннях *N*.

Інший характерний приклад односпрямованої функції це модульна експонента з фіксованими підставою і модулем[9].

Як і в випадку симетричних криптографічних систем, за допомогою асиметричних криптосистем забезпечується не тільки конфіденційність, але також справжність і цілісність переданої інформації. Справжність і цілісність будь-якого повідомлення забезпечується формуванням цифрового підпису цього повідомлення і відправкою в зашифрованому вигляді повідомлення разом з цифровим підписом. Перевірка відповідності підпису отриманого повідомлення після його попереднього дешифрування являє собою перевірку цілісності та автентичності прийнятого повідомлення.

Переваги асиметричних криптографічних систем перед симетричними криптосистемами:

- в асиметричних криптосистемах вирішена складна проблема розподілу ключів між користувачами, так як кожен користувач може згенерувати свою пару ключів сам, а відкриті ключі користувачів можуть вільно публікуватися і поширюватися по мережевим комунікацій;
- зникає квадратична залежність числа ключів від числа користувачів; в асиметричною криптосистемою число використовуваних ключів пов'язано з числом абонентів лінійною залежністю (в системі з *N* користувачів використовуються *2УУ* ключів), а не квадратичні, як в симетричних системах;
- асиметричні криптосистеми дозволяють реалізувати протоколи взаємодії

сторін, які не довіряють один одному, оскільки при використанні асиметричних криптосистем закритий ключ повинен бути відомий тільки його власнику.

Недоліки асиметричних криптосистем:

- на справжній момент немає математичного доказу незворотності використовуваних в асиметричних алгоритмах функцій;
- асиметричне шифрування істотно повільніше симетричного, оскільки при шифруванні і розшифровці використовуються досить ресурсомісткі операції. З цієї ж причини реалізувати апаратний шифратор з асиметричним алгоритмом істотно складніше, ніж реалізувати апаратно симетричний алгоритм;
- необхідність захисту відкритих ключів від підміни.
-

2.3 Метод заміни

Методи шифрування заміною (підстановкою) засновані на тому, що символи вихідного тексту, зазвичай розділені на блоки і записані в одному алфавіті, замінюються одним або декількома символами іншого алфавіту відповідно до прийнятого правилом перетворення.

Одним з важливих підкласів методів заміни є одноалфавітні (або моноалфавитні) підстановки, в яких встановлюється однозначна відповідність між кожним знаком a_i вихідного алфавіту повідомлень A і відповідним знаком ei зашифрованого тексту E. Одноалфавітна підстановка іноді називається також простою заміною, так як є найпростішим шифром заміни[10].

У загальному випадку при одноалфавітній підстановці відбувається однозначна заміна вихідних символів їх еквівалентами з вектора замін (або таблиці замін). При такому методі шифрування ключем є використовувана таблиця замін.

Підстановка може бути задана за допомогою таблиці, наприклад, як показано в таблиці 2.1

Таблиця 2.1 – Таблиця замін для двох шифрів

Відкр. текст	**Шифр 1**	**Шифр 2**	**Відкр. текст**	**Шифр 1**	**Шифр 2**	**Відкр. текст**	**Шифр 1**	**Шифр 2**
А	В	^	М	Т	№	Ч	М	∑
Б	И	@	Н	Ц	#	Ш	У	∇
В	О	)	О	.	-	Щ	Д	γ
Г	А	+	П	Ж	=	Ъ	Э	χ
Д	Щ	<	Р	Г	(	Ы	Н	⊕
Е	П	>	С	Л	?	Ь	Ю	×
Ж	К	∀	Т	Х	%	Э	Ы	ω
З	Б	♦	У	С	⊗	Ю	Ш	$
И	Ъ	*	Ф	Ь	!	Я	Е	Δ
К	пробіл	❤	Х	Ч	№	пробіл	Ф	∞
Л	Р	♠	Ц	З	®	.	Я	♣

У таблиці на рис. 2.3 насправді об'єднані відразу дві таблиці. Одна (шифр 1) визначає заміну російських букв вихідного тексту на інші російські літери, а друга (шифр 2) заміну букв на спеціальні символи. Вихідним алфавітом для обох шифрів будуть великі російські літери (за винятком букв "Е" і "Й"), пробіл і крапка.

Зашифроване повідомлення з використанням будь-якого шифру моноалфавитної підстановки отримують у такий спосіб. Береться черговий знак з вихідного повідомлення. Визначається його позиція в стовпці "Одкр. Текст" таблиці замін. У зашифроване повідомлення вставляється шифрований символ з цієї ж рядки таблиці замін.

Спробуємо зашифрувати повідомлення "вышлите подкрелпение" с використанням цих двох шифрів (рис. 2.4). Для цього беремо першу букву вихідного повідомлення "В". У таблиці на рис. 2.3 в стовпці "Шифр 1" знаходимо для букви "В" замінний символ. Це буде буква "О". Записуємо букву "О" під

літерою "В". Потім розглядаємо другий символ вихідного повідомлення букву "И". Знаходимо цю букву в стовпці "Откр. Текст" і з шпальти "Шифр 1" беремо букву, що стоїть на тому самому рядку, що і буква "И". Таким чином отримуємо другий символ зашифрованого повідомлення букву "Н". Продовжуючи діяти аналогічно, можна зашифровувати все вихідне повідомлення, як в Таблиці 2.2, де перша стрічка – відкрите повідомлення, друга стрічка зашифроване повідомлення зв використанням шифру 1, а третя стрічка – зашифроване повідомлення з використанням шифру 2.

Таблиця 2.2. – Відповідність відкритого повідомлення шифрам 1 та 2

Отриманий таким чином текст має порівняно низький рівень захисту, так як вихідний і зашифрований тексти мають однакові статистичні закономірності. При цьому не має значення, які символи використані для заміни – перемішані символи вихідного алфавіту або знаки, що виглядають таємничо [10].

Зашифроване повідомлення може бути розкрите шляхом так званого частотного криптоаналізу. Для цього можуть бути використані деякі статистичні дані мови, на якому написано повідомлення.

Відомо, що в текстах російською мовою найбільш часто зустрічаються символи О, І. Трохи рідше зустрічаються букви Е, А. З приголосних найчастіші символи Т, Н, Р, С. В розпорядженні криптоаналітиків є спеціальні таблиці частот зустрічальності символів для текстів різних типів наукових, мистецьких тощо.

Криптоаналітика уважно вивчає отриману криптограму, підраховуючи при цьому, які символи скільки разів зустрілися. Спочатку найбільш часто повторюванізнаки зашифрованого повідомлення замінюються, наприклад, буквами О. Далі проводиться спроба визначити місця для букв І, Е, А. Потім підставляються найбільш часто повторювані приголосні. На кожному етапі

оцінюється можливість "поєднання" тих чи інших букв. Наприклад, в російських словах важко знайти чотири поспіль голосні літери, слова в російській мові не починаються з літери И і т.д. Насправді для кожної природної мови (української, російської, англійської і т.д.) існує безліч закономірностей, які допомагають розкрити фахівцям зашифровані противником повідомлення.

Приклади шифрів простої заміни:

Атбаш – шифр простої заміни, використаний для єврейського алфавіту і отримав звідти свою назву. Шифрування відбувається заміною першої літери алфавіту на останню, другий на передостанню (алеф (перша буква) замінюється на тав (останню), бет (друга) замінюється на шин (передостання); з цих поєднань шифр і отримав свою назву). Шифр Атбаш для англійського алфавіту зображено в Таблиці 2.3, де ВА – вихідний алфавіт, а АЗ – алфавіт заміни:

Таблиця 2.3 – Шифр Атбаш

А																								
З																								

Шифр Цезаря один з найдавніших шифрів. При шифруванні кожна буква замінюється іншою, віддаленої від неї в алфавіті на фіксоване число позицій. Шифр названий на честь римського імператора Гая Юлія Цезаря, який використав його для секретного листування. Природним розвитком шифру Цезаря став шифр Віженера. Шифрування з використанням ключа $k = 4$ зображено в таблиці 2.4.

Таблиця 2.4 – Шифрування з використанням ключа $k = 4$

А																								
З																								

Сучасним прикладом шифру Цезаря є ROT13. Він зрушує кожен символ англійського алфавіту на 13 позицій. Використовується в інтернет-форумах, як засіб для приховування спойлерів, основних думок, рішень загадок і образливих матеріалів від випадкового погляду.

Шифр з використанням кодового слова є одним з найпростіших як в реалізації, так і в розшифровуванні. Ідея полягає в тому, що вибирається кодове слово, яке пишеться попереду, потім виписуються інші літери алфавіту в своєму порядку. Шифр з використанням кодового слова WORD наведено в Таблиці 2.5.

Таблиця 2.5 – Шифр з використанням кодового слова WORD

A																								
Z																								

Як ми бачимо, при використанні короткого кодового слова ми отримуємо дуже просту заміну. Ми можемо використовувати слово з повторюваними літерами в якості кодового слова, але тільки в тому випадку, якщо приберемо з кодового слова зайві літери, інакше це призведе до неоднозначності розшифровки, тобто двом різним буквах вихідного алфавіту буде відповідати одна і та ж буква шифрованого тексту.

За традицією, зашифрований текст пишуть блоками (інша назва «групи») по 5 символів, не враховуючи пунктуацію і прогалини. Це допомагає уникнути помилок при передачі шифрованого повідомлення і дозволяє приховати кордону слів в початковому тексті. Блок містить 5 символів, так як раніше їх було зручно передавати по телеграфу [10].

Головний недолік цього методу шифрування це те, що останні букви алфавіту (які мають низькі коефіцієнти при частотному аналізі) мають тенденцію залишатися в кінці. Більш захищений спосіб побудувати алфавіт заміни полягає в тому, щоб виконати колоночного переміщення (переміщення стовпців) в алфавіті, використовуючи ключове слово, але це нечасто робиться.

Незважаючи на те, що число можливих ключів є дуже великим ($26! = 2^{88.4}$), цей вид шифру може бути легко зламаним. За умови, що повідомлення має достатню довжину, криптоаналітик може припустити значення деяких найпоширеніших букв виходячи з аналізу частотного розподілу символів в зашифрованому тексті. Це дозволяє формувати окремі слова, які можуть бути попередньо використані, для подальшого отримання більш повного вирішення Згідно віддалі унікальності англійської мови 27.6 букв від зашифрованого тексту має бути достатньо, щоб зламати шифр простої заміни. На практиці зазвичай достатньо 50 символів для злому, хоча деякі шифротексти можуть бути зламані і з меншою кількістю символів, якщо знайдені будь-які нестандартні структури. Але при рівномірному розподілі символів в тексті можуть знадобитися куди довші шифротексти для злому.

Відстань унікальності термін, який використовується в криптографії, що звертається до довжини оригінального шифротекста, якого повинно бути достатньо для злому шифру.

2.4 Метод перестановки

Суть методів перестановки полягає в поділі вихідного тексту на блоки фіксованої довжини і подальшій перестановці символів всередині кожного блоку за певним алгоритмом.

Перестановки виходять за рахунок різниці шляхів запису вихідної інформації і шляхів зчитування зашифрованої інформації в межах геометричної фігури. Прикладом найпростішої перестановки є запис блоку вихідної інформації в матрицю по рядках, а зчитування по стовпцях. Заповнення рядків матриці і зчитування зашифрованої інформації по стовпчиках може здаватися ключем. Крипостійкість методу залежить від довжини блоку (розмірності матриці). Так для блоку довжиною 64 символи (розмірність матриці 8 х 8) можливі 1,6 х 109 комбінацій ключа. Для блоку довжиною 256 символів (матриця розмірністю 16 х 16) число можливих ключів досягає 1,4 х 1026. Рішення завдання перебору ключів в останньому випадку навіть для сучасних ЕОМ представляє істотну складність.

Перестановки використовуються також в методі, заснованому на застосуванні маршрутів Гамільтона. Цей метод реалізується шляхом виконання наступних кроків.

1. Вихідна інформація розбивається на блоки. Якщо довжина шифрованої інформації не кратна довжині блоку, то на вільні місця останнього блоку поміщаються спеціальні службові символи-заповнювачі (наприклад *).

2. Символами блоку заповнюється таблиця, в якій для кожного порядкового номера символу в блоці відводиться цілком певне місце (рис. 2.5).

3. Зчитування символів з таблиці здійснюється по одному з маршрутів. Збільшення числа маршрутів підвищує Крипостійкість шифру. Маршрути вибираються або послідовно, або їх черговість задається ключем *К*.

4. Зашифрована послідовність символів розбивається на блоки фіксованої довжини *L*. Величина *L* може відрізнятися від довжини блоків, на які розбивається вихідна інформація на кроці 1.

Дешифрування проводиться в зворотному порядку. Відповідно до ключем вибирається маршрут і заповнюється таблиця згідно з цим маршрутом [11].

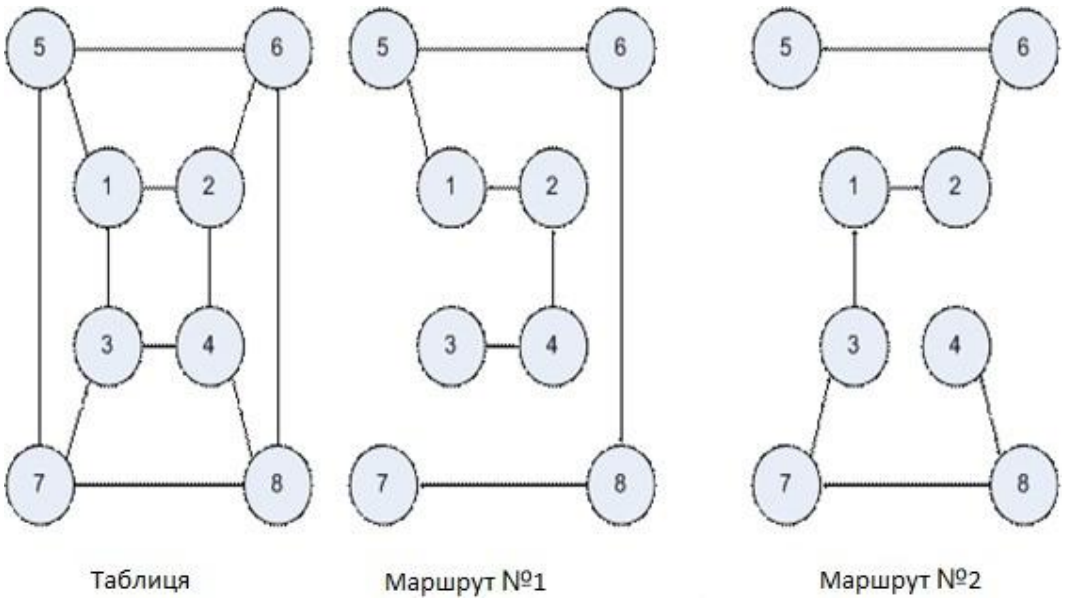

Рисунок 2.3 – Варіант 8-елементної таблиці та маршрутів Гамільтона

З таблиці символи зчитуються в порядку проходження номерів елементів. Нижче наводиться приклад шифрування інформації з використанням маршрутів

Гамільтона.

Нехай потрібно зашифрувати вихідний текст Т0 = <МЕТОДИ_ПЕРЕСТАНОВКИ>. Ключ і довжина зашифрованих блоків відповідно рівні*: K = <2, 1, 1>, L = 4*. Для шифрування використовуються таблиця і два маршрути. Для заданих умов маршрути з заповненими матрицями мають вигляд, показаний на рис. 2.4.

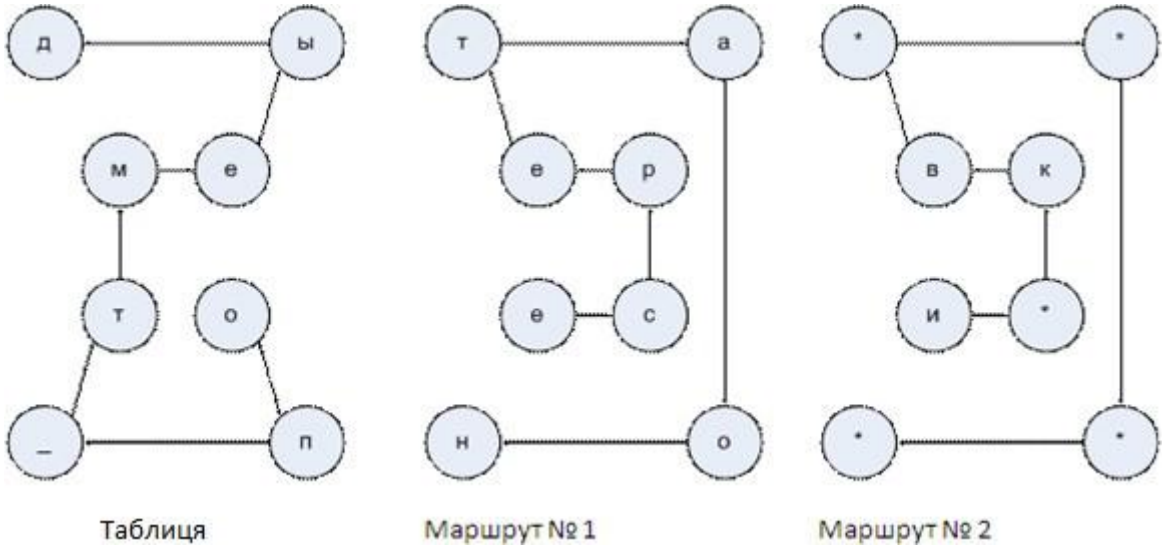

Рисунок 2.4 – Приклад шифрування за допомогою маршрутів Гамільтона

Крок 1. Вихідний текст розбивається на три блоки:

Б1 = <МЕТОДЫ_П>; Б2 = <ЕРЕСТАНО>;

Б3 = <ВКИ*****>.

Крок 2. Заповнюються три матриці з маршрутами 2, 1, 1 (рис. 2.5).

Крок 3. Отримання шифротексту шляхом розстановки символів в відовідності маршрутам [11].

Т1 = <ОП_ТМЕЫДЕСРЕТАОНИ*КВ****>.

Крок 4. Розбиття на блоки шифртексту.

Т1 = <ОП_Т МЕЫД ЕСРЕ ТАОН И*КВ ****>.

У практиці велике значення має використання спеціальних апаратних схем, що реалізують метод перестановок (рис. 2.5).

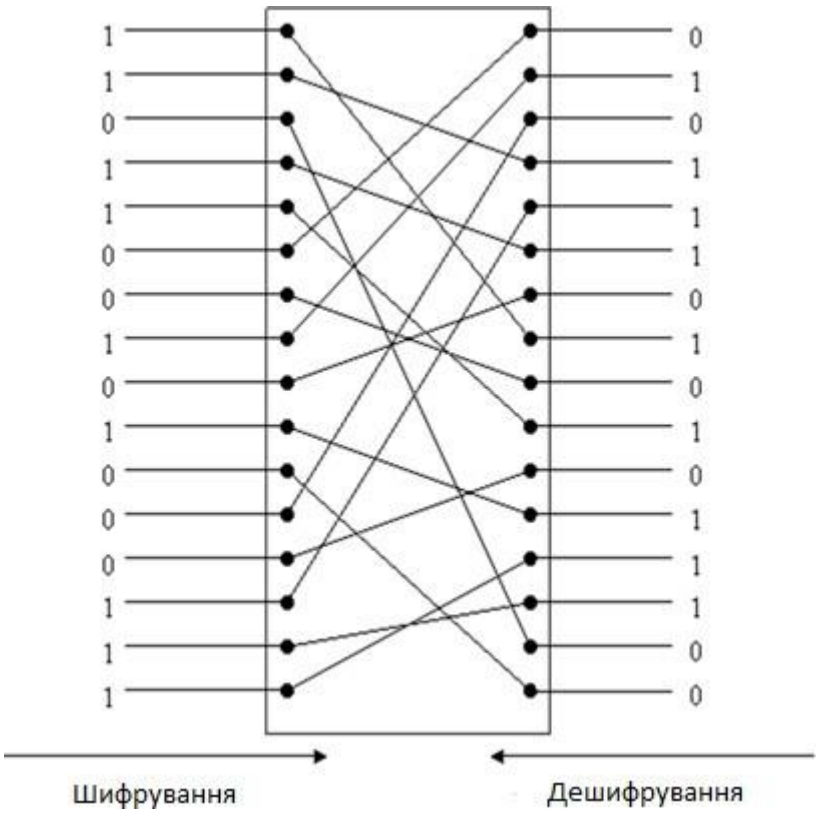

Рисунок 2.5 – Схема перестановок

Паралельний двійковий код блоку вихідної інформації (наприклад, два байта) подаються на схему. За рахунок внутрішньої комутації в схемі здійснюється перестановка біт в межах блоку. Для дешифрування блоку інформації входи і виходи схеми міняються місцями.

Методи перестановок просто реалізуються, але мають два суттєвих недоліки. По-перше, вони допускають розкриття шифртекста за допомогою статистичної обробки. По-друге, якщо вихідний текст розбивається на блоки довжиною *K*символів, то криптоаналітику для розкриття шифру досить направити в систему шифрування *K-1* блок тестової інформації, в яких всі символи за винятком одного однакові.

2.5 Криптографічні атаки

При слові «криптографія» дехто згадує свій пароль WiFi, зелений замочок поруч з адресою улюбленого сайту і то, як важко залізти в чужу пошту. Інші згадують низку вразливостей останніх років з промовистими абревіатурами

(DROWN, FREAK, POODLE ...), стильними логотипами і попередженням терміново оновити браузер.

Криптографія охоплює все це, але суть в іншому. Суть в тонкій грані між простим і складним. Деякі речі просто зробити, але складно повернути назад: наприклад, розбити яйце. Інші речі легко зробити, але важко повернути назад, коли відсутня маленька важлива вирішальна частина: наприклад, відкрити замкнені двері, коли «вирішальна частина» є ключем. Криптографія вивчає ці ситуації і способи їх практичного використання.

Багато з атак засновані на декількох загальних принципах, а нескінченні сторінки формул часто зводяться до простих для розуміння ідей. Існує декілька базових стратегій криптографічних атак, а саме: брутфорс, частотний аналіз, інтерполяція, попереднє обчислення та кроспротоколи/пониження.

Простий брутфорс. Схема шифрування складається з двох частин:

1) функція шифрування, яка приймає повідомлення (відкритий текст) в поєднанні з ключем, а потім створює зашифроване повідомлення шифротекст;

2) функція дешифрування, яка приймає шифротекст і ключ і створює відкритий текст. І шифрування, і дешифрування повинні бути легко обраховуються з ключем і важко без нього.

Припустимо, що ми бачимо шифротекст і намагаємося розшифрувати його без будь-якої додаткової інформації (це називається атакою «тільки шифротекст»). Якщо ми якимось чарівним чином знайдемо правильний ключ, то можемо легко перевірити, що він дійсно правильний, якщо результат є розумним повідомленням.

Зверніть увагу, що тут два неявних припущення. По-перше, що ми знаємо, як виконати розшифровку, тобто, як працює криптосистема. Це стандартне припущення при обговоренні криптографії. Приховування деталей реалізації шифру від зловмисників може здатися додатковим заходом безпеки, але як тільки зловмисник дізнається ці деталі, дана додаткова безпека непомітно і необоротно втрачена. Такий принцип Керчхоффса: потрапляння системи в руки ворога не повинно завдавати незручностей.

По-друге, ми припускаємо, що правильний ключ є єдиним ключем, який приведе до розумної розшифровки. Це також розумне припущення; воно виконується, якщо шифротекст набагато довше ключа і добре читається. Як правило, так і буває в реальному світі, за винятком величезних непрактичних ключів або інших махінацій, які краще залишити в стороні.

З огляду на вищевикладене, виникає стратегія: перевірити кожен можливий ключ. Це називається брутфорсом, і така атака гарантовано працює проти всіх практичних шифрів в кінцевому підсумку. Наприклад, брутфорса досить, щоб зламати шифр Цезаря, древній шифр, де ключем є одна буква з алфавіту, що має на увазі трохи більше 20 можливих ключів.

На жаль для криптоаналітиків, збільшення розміру ключа добре захищає від брутфорса. У міру зростання розміру ключа кількість можливих ключів збільшується експоненціально. З сучасними розмірами ключів простий брутфорс абсолютно не практичний. Щоб зрозуміти, що ми маємо на увазі, візьмемо найшвидший відомий суперкомп'ютер на середину 2019 року: Summit від IBM, з піковою продуктивністю близько 1017 операцій в секунду. Сьогодні типова довжина ключа становить 128 біт, що означає 2128 можливих комбінацій. Для перебору всіх ключів суперкомп'ютера Summit потрібен час, який приблизно в 7800 разів перевищує вік Всесвіту. На рисунку 2.6 приведено приклад брутфорсу за допомогою утиліти Brutecms.

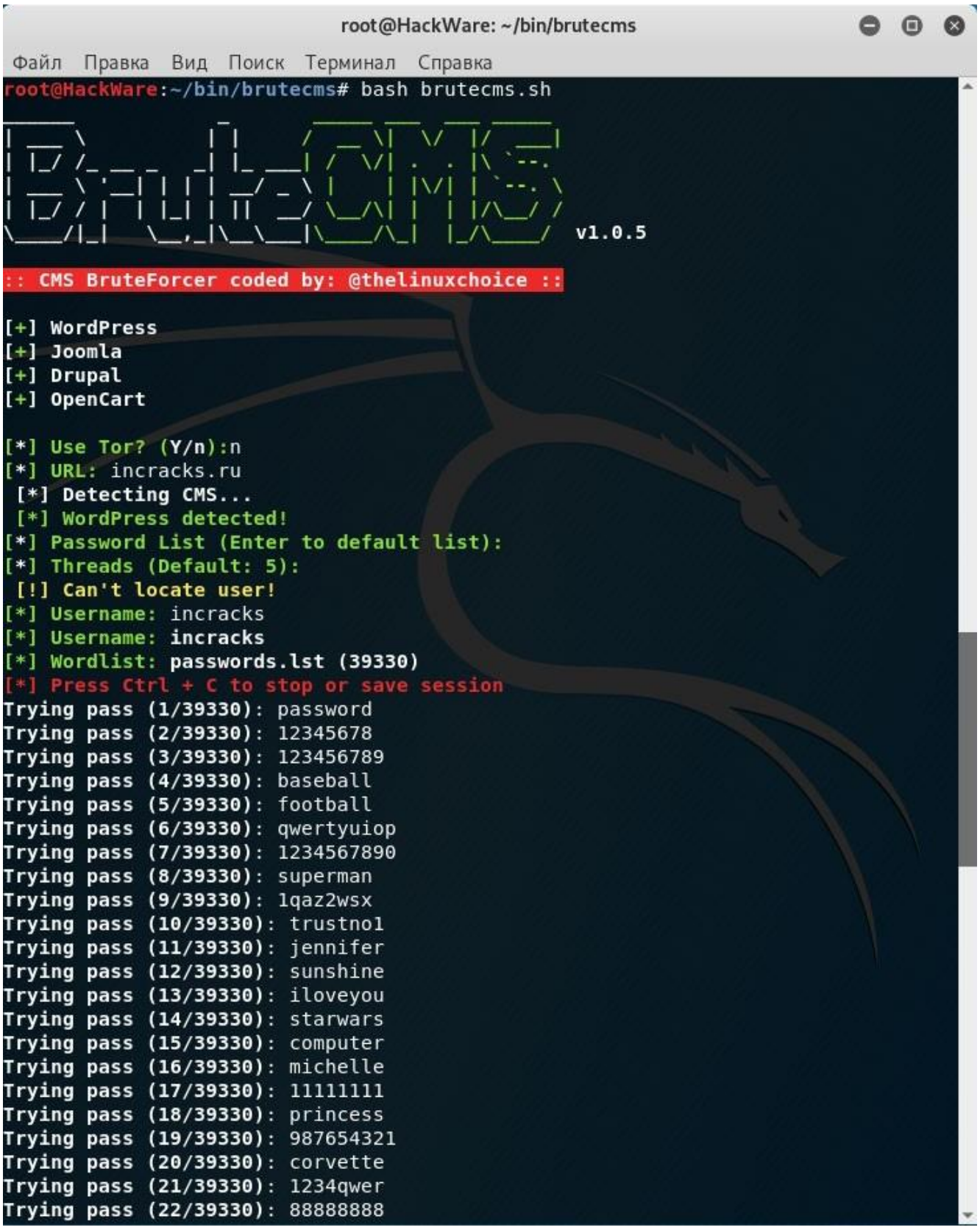

Рисунок 2.6 – Перебір простих паролів через Brutecms

Частотний аналіз. Більшість текстів це не тарабарщина. Наприклад, в англомовних текстах багато букв 'e' і артиклів 'the'; в двійкових файлах багато

нульових байтів як заповнювач між фрагментами інформації. Частотний аналіз - будь-яка атака, яка використовує цей факт.

Канонічним прикладом шифру, уразливого для цієї атаки, є простий шифр підстановки. У цьому шифрі ключ являє собою таблицю з заміною всіх букв. Наприклад, 'g' замінюється на 'h', 'o' на j, Тому слово 'go' перетворюється в 'hj'. Цей шифр важко піддається простому Брутфорсу, так як існує дуже багато можливих таблиць підстановки. Якщо вас цікавить математика, ефективна довжина ключа становить близько 88 біт: це log2(26!). Але частотний аналіз зазвичай швидко справляється із завданням.

Розглянемо наступний шифротекст, оброблений простим шифром підстановки:

XDYLY ALY UGLY XDWNKE WN DYAJYN ANF YALXD DGLAXWG XDAN ALY FLYAUX GR WN OGQL ZDWBGEGZDO

Оскільки Y зустрічається часто, в тому числі в кінці багатьох слів, ми можемо попередньо припустити, що це буква e:

XDeLe ALe UGLe XDWNKE WN DeAJeN ANF eALXD DGLAXWG XDAN ALe FLeAUX GR WN OGQL ZDWBGEGZDO

Пара XD повторюється на початку декількох слів. Зокрема, поєднання XDeLe явно передбачає слово these або there, тому продовжуємо:

theLe ALe UGLe thWNKE WN heAJeN ANF eALth DGLAtWG thAN ALe FLeAUt GR WN OGQL ZDWBGEGZDO

Далі припустимо, що L відповідає r, A a і так далі. Ймовірно, доведеться зробити кілька спроб, але в порівнянні з повним брутфорсом ця атака відновлює вихідний текст в найкоротші терміни:

there are more things in heaven and earth horatio than are dreamt of in your philosophy

Для деяких рішення таких «криптограми» захоплююче хобі.

Ідея частотного аналізу більш фундаментальна, ніж здається на перший погляд. І він застосовується до набагато більш складних шифрів. Протягом всієї історії різні конструкції шифрів намагалися протистояти такій атаці за допомогою

«поліалфавітних підстановки». Тут в процесі шифрування таблиця заміни букв змінюється складними, але передбачуваними способами, які залежать від ключа. Всі ці шифри свого часу вважалися важкими для злому; і все ж скромний частотний аналіз в підсумку всії їх здолав.

Найамбіційнішим поліалфавітних шифром в історії і, напевно, найвідомішим, був шифр «Енігми» у Другій світовій війні. Він був відносно складним в порівнянні з попередниками, але в результаті довгої і наполегливої роботи британські криптоаналітики зламали його за допомогою частотного аналізу. Їм довелося порівнювати відомі пари відкритих і зашифрованих текстів (так звана «атака на основі відкритих текстів») і навіть провокуючи користувачів «Енігми» на шифрування певних повідомлень з аналізом результату («атака на основі підібраного відкритого тексту»). Але це не полегшило долю переможених армій ворогів і потоплених підводних човнів.

Після цього тріумфу частотний аналіз зник з історії криптоанализа. Шифри сучасної цифрової епохи створені для роботи з бітами, а не літерами. Що ще більш важливо, ці шифри розроблені з похмурим розумінням того, що пізніше отримало назву закон Шнайера: будь-хто може створити алгоритм шифрування, який сам не зможе зламати. Недостатньо, щоб шифрувальна система здавалася складною: щоб довести свою цінність, вона повинна пройти безжалісний огляд безпеки від багатьох криптоаналітиків, які зроблять все можливе, щоб зламати шифр.

Попередні обчислення. Атаки проти конкретного шифру піддаються безжалісному аналізу витрат та вигоди. Якщо співвідношення сприятливо, атака не відбудеться. Але атаки, які діють відразу проти багатьох потенційних жертв, майже завжди окупаються, і в цьому випадку найкраща практика проектування припустити, що вони почалися з першого дня. У нас є по суті криптографічний версія закону Мерфі: «Все, що реально може зламати систему, зламає систему».

Найпростіший приклад криптосистеми, вразливою до атаки з попередніми обчисленнями, шифр з постійним алгоритмом без використання ключа. Так було у випадку з шифром Цезаря, який просто зрушує кожну букву алфавіту на три

букви вперед (таблиця закільцевана, тому остання буква в алфавіті шифрується третьою). Тут знову проявляється принцип Керхгоффа: як тільки система зламана, вона зламана назавжди.

Концепція проста. Навіть початківець розробник криптосистем, швидше за все, усвідомлює загрозу і відповідним чином підготувався. Якщо подивитися на еволюцію криптографії, такі атаки були недоречні для більшості шифрів, починаючи з перших поліпшених версій шифру Цезаря, аж до занепаду поліалфавітних шифрів. Такі атаки повернулися тільки з настанням сучасної ери криптографії.

Це повернення викликано двома факторами. По-перше, нарешті, з'явилися досить складні криптосистеми, де можливість експлуатації після злому була очевидною. По-друге, криптографія набула такого широкого поширення, що мільйони непрофесіоналів кожен день приймали рішення, де і які частини криптографії використовувати повторно. Минув деякий час, перш ніж експерти усвідомили, що виникли ризики і підняли тривогу [12].

Інтерполяція досліджує відомі пари відкритого і зашифрованого текстів, отримані в результаті застосування одного і того ж ключа. З кожної пари витягуються окремі спостереження, які дозволяють зробити загальний висновок про ключі. Всі ці висновки розпливчасті і здаються марними, поки раптово не досягнуть критичної маси і не приведуть до єдиного можливого висновку: яким би неймовірним він не був, він повинен бути істинним. Після цього або розкривається ключ, або процес розшифровки стає настільки відпрацьованим, що його можна тиражувати.

Проілюструємо на простому прикладі, як працює інтерполяція. Припустимо, що ми хочемо прочитати інформацію в смартфоні нашого знайомого, Максима. Він шифрує кожне число в своєму смартфоні за допомогою простої криптосистеми, про яку дізнався з рекламного оголошення на сайті «Криптографія для дітей». Система працює наступним чином: Максим вибирає два числа, які йому подобаються: M і N. З цього моменту, щоб зашифрувати будь-яке число x, він обчислює $Mx + N$. Наприклад, якщо Боб вибрав $M = 3$ і $N = 4$, то

цифра 2 зашифрують як 3 * 2 + 4 = 10.

Припустимо, 28 грудня ми помітили, що Максим щось пише на своєму смартфоні. Коли він закінчить, ми непомітно розблокуємо його смартфон і подивимося на запис:

Дата: 235/520 Привіт,

Сьогодні був плодовитий день. Через 64 дня у мене зустріч з Оленою, яка живе в квартирі 843. Я дійсно думаю, що вона може бути 26!

Оскільки ми маємо серйозний намір простежити за Максимом на його побаченні (в цьому сценарії нам по 15 років), то критично важливо дізнатися дату, а також адресуОлени. На щастя, ми помічаємо, що криптосистема Максима вразлива для атаки інтерполяції. Ми можемо і не знати *M* і *N*, але ми знаємо сьогоднішню дату, тому у нас є дві пари «відкритий текст шифротекст». А саме, ми знаємо, що 12 шифрується в 235, а 27 в 520. Що і запишемо:

*M * 12 + N = 235*

*M * 27 + N = 520*

Оскільки нам 15 років, ми вже знаємо про систему двох рівнянь з двома невідомими, чого в даній ситуації досить для знаходження *M* і *N* без особливих проблем. Кожна пара «відкритий текст-шифротекст» накладає обмеження на ключ Максима і двох обмежень разом досить, щоб повністю відновити ключ. У нашому прикладі відповідь *M = 19* і *N = 7* (при *19x + 7 = 26x = 1*, так що 26 в щоденнику відповідає слову 'the one', тобто «та сама»).

Інтерполяційні атаки, звичайно, не обмежуються такими простими прикладами. Кожна криптосистема, яка зводиться до добре зрозумілому математичного об'єкту і списку параметрів, наражається на ризик інтерполяційної атаки чим більше зрозумілий об'єкт, тим вище ризик.

Новачки часто скаржаться, що криптографія це «мистецтво проектування якомога більше потворних речей». Ймовірно, багато в чому винні атаки інтерполяції. Максим може або використовувати елегантний математичний дизайн, або зберегти конфіденційність побачення з Оленою але на жаль, зазвичай не можна отримати і те, і інше.

Кросс-протокол або атака на пониження є формою криптографічної атаки на комп'ютерну систему або комунікаційний протокол, що змушує його відмовитися від режиму високої якості роботи (наприклад, зашифрованих з'єднань) на користь більш старої, більш низької версії режиму роботи (наприклад, відкритий текст), який зазвичай надається для забезпечення сумісності зі старими системами. Приклад такої вразливості був виявлений в OpenSSL, який дозволяв зловмиснику погоджувати використання більш низькою версії TLS між клієнтом і сервером. Це один з найбільш поширених типів атак на більш ранню версію. Інший приклад перехоплення веб-трафіку і перенаправлення користувача з безпечної HTTPS-версії веб-сайту на незашифровану HTTP-версію.

Атаки на більш ранню версію часто реалізуються як частина атаки «людина посередині» (MITM) і можуть використовуватися як спосіб включення криптографічної атаки, яка в іншому випадку була б неможлива. Атаки переходу на більш ранню версію були постійною проблемою з сімейством протоколів SSL / TLS; Приклади таких атак включають атаку POODLE. Атаки переходу на більш ранню версію протоколу TLS приймають різні форми. Дослідники класифікували атаки на більш ранню версію по чотирьом різним векторам, що представляє собою основу для міркувань про атаки на більш ранню версію наступним чином:

1. Протокол *елемент*, який призначений:
    - алгоритм;
    - версія;
    - шар.
2. Тип *уразливості*, при якій можлива атака:
    - реалізація;
    - дизайн;
    - модель довіри;
3. Використаний *метод*:
    - падіння;
    - модифікація;
    - ін’єкція.

4. Рівень збитків, що нанесла атака:
   - зломана безпека;
   - ослаблена безпека.

Є кілька недавніх пропозицій, які використовують концепцію попереднього знання, щоб дозволити клієнтам TLS (наприклад, веб-браузерам) захищати конфіденційні доменні імена від певних типів атак на більш ранню версію, які використовують підтримку клієнтами застарілих версій або які не рекомендують шифрувальних наборів (наприклад, тих , які не підтримують пряму секретність або автентифікації шифрування), такі як POODLE, фрагментація ClientHello і варіант атак DROWN (також відомого як «спеціальне утоплення») на попередньої версії. Видалення зворотної сумісності часто є єдиним способом запобігти атакам на більш ранню версію. Однак іноді клієнт і сервер можуть розпізнавати один одного як оновлені, що не дозволяє їм. Наприклад, якщо і веб-сервер, і призначений для користувача агент реалізують HTTP Strict Transport Security, і призначений для користувача агент знає це про сервер (або раніше звертався до нього через HTTPS, або тому, що він знаходиться в «списку попереднього завантаження HSTS»), то для користувача агент відмовиться отримати доступ до сайту через звичайний HTTP, навіть якщо зловмисний маршрутизатор представить його і сервер один одному як тих, котрі підтримують HTTPS.

2.6 Стеганографія

Стеганографія відома ще з часів Геродота. У Стародавній Греції послання писалися гострими паличками на дощечках, покритих воском. В одній з історій Демерат хотів послати в Спарту повідомлення про загрозу нападу Ксеркса. Тоді він зіскоблив віск з дощечки, написав послання безпосередньо на дереві, потім знову покрив її воском. В результаті дошка виглядала невикористаною і без проблем пройшла огляд сотників. Ще один, досить несподіваний спосіб приховування інформації або умовних знаків татуювання на голові голеного посланця. Коли в V столітті до н.е. тиран Гіст, перебуваючи під наглядом царя Дарія в Сузах, повинен був послати секретне повідомлення своєму родичеві в анатолийский місто Мілет, він поголив наголо свого раба і витатуював послання

на його голові. Коли волосся знову відросло, раб відправився в шлях. Так Геродот описує один з перших випадків застосування в стародавньому світі стеганографії мистецтва прихованого повідомлення [13].

Стеганографія це наука про приховану передачу інформації шляхом збереження в таємниці самого факту її передачі. На відміну від іншого напрямку захисту інформації, криптографії, яка приховує вміст секретного повідомлення,стеганографія приховує саме його існування.

Стеганографія та криптографія не замінюють, а доповнюють одне одного. Якщо файл захищається методами стеганографії, це знижує ймовірність виявлення наявності вбудованої в зображення інформації. А якщо ця інформація зашифрована (криптографія), то вона отримує ще один, додатковий рівень захисту.

Цифрове зображення з вбудованою в нього секретною інформацією назвемостегосистемою. Розглянемо її основні складові:

- Контейнер саме зображення, яке необхідно захистити. Контейнер, вяке не вбудовано повідомлення, називається порожнім або контейнером-оригіналом. Контейнер, з вбудованим повідомленням називається контейнером-результатом абозаповненим;
- Приховане повідомлення файл або повідомлення, що вбудовується в контейнер,що дозволяє ідентифікувати автора цифрового зображення;
- Стегоключ (або просто ключ) секретний ключ, необхідний для приховуванняінформації. Залежно від кількості рівнів захисту може бути від одного додекількох ключів.
- Стеганографічний канал (стегоканал) канал передачі стего.

При створенні стегосистеми необхідно врахувати наступні моменти:

- викрадачеві відомі методи стеганографії і способи побудови стегосистем;
- методи приховування вбудованого повідомлення повинні забезпечувати автентичності іцілісність вихідного цифрового зображення;
- єдина інформація, яка повинна залишитися невідомою викрадачеві відкрито поширюваного файлу це стегоключа, за допомогою якого автор зможе

показати вміст вбудованого повідомлення і довести своє авторство [13];

- якщо викрадач дізнається про існування прихованого повідомлення, це не повиннодозволити йому отримати повідомлення до тих пір, поки ключ йому не відомий;
- викрадач повинен бути позбавлений технічних або інших переваг врозпізнаванні прихованого повідомлення.

Загальний вигляд стегосистем показано на рисунку 2.7.

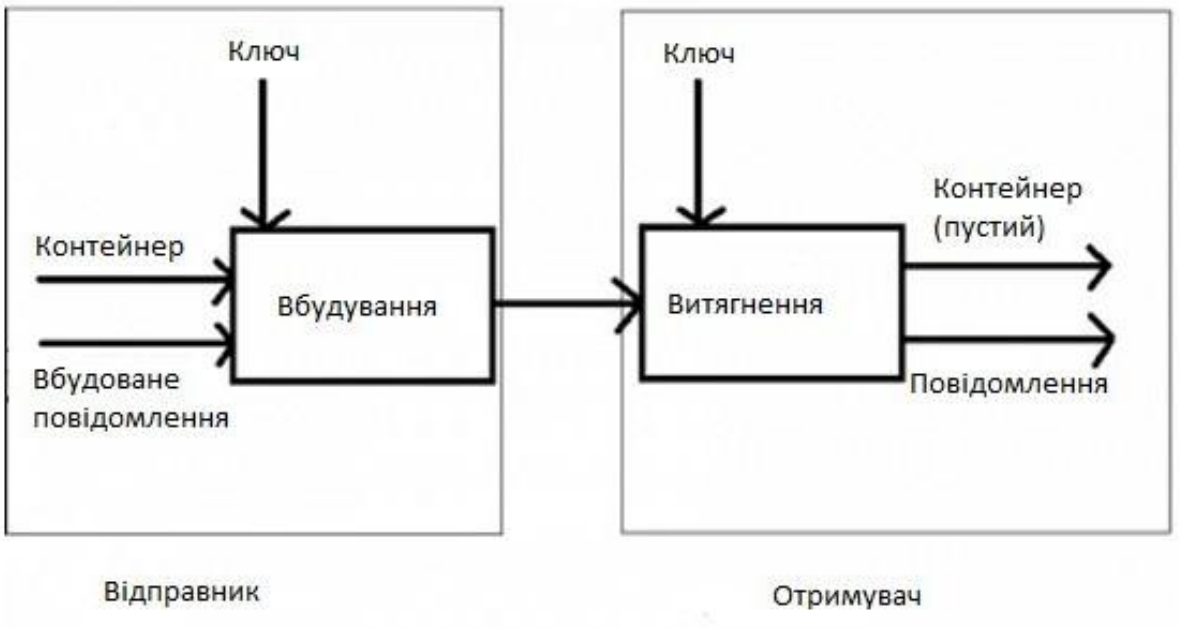

Рисунок 2.7 – Загальна схема стегосистем

Найбільш поширеним, але менш стійким є **метод заміни найменших значимих бітів (LSB-метод)**. Він полягає у використанні похибки дискретизації, яка завжди існує в оцифрованих зображеннях. Дана похибка дорівнює найменшому значущому розряду числа, визначає величину колірної складової елемента зображення (пікселя). Тому модифікація молодших бітів в більшості випадків не викликає значної трансформації зображення і не виявляється візуально.

Іншим популярним методом вбудовування повідомлень є **використання особливостей форматів даних, що використовують стиснення з втратою даних** (наприклад, JPEG-формат). Цей метод (на відміну від LSB-методу) більш стійкий до геометричних перетворень і виявлення каналу передачі, так як є

можливість в широкому діапазоні варіювати якість стисненого зображення, що робить неможливим визначення походження спотворення [13].

Розміщені в Інтернеті цифрові зображення зазвичай бувають збережені в **кольоровій моделі RGB**. Графічні файли цієї цифрової моделі кодують кожну точку малюнка трьома байтами. Кожна така точка складається з адитивних складових: червоного, зеленого і синього. При зміні кожного з трьох найменш значущих біт інтенсивність даної точки змінюється менше ніж на 1%. Це дозволяє приховувати в стандартному графічному зображенні об'ємом 800 Кбайт близько 100 Кбайт інформації (при перегляді зображення це непомітно).

Каналом R (червоний) людське око вловлює 7 біт з 8, по G (зелений) 8 біт з 8; по каналу B (синій) 4 біт з 8. Можна зробити висновки про те, що людське око найменш сприйнятливе до синього і червоного кольорів. Ця перевага дозволяє вбудовувати інформації в область синього і червоного кольорів.

Найбільш затребуваний для захисту авторських прав метод стеганографії – це вбудовування **цифрових водяних знаків**, налаштований на вбудовування в мультимедійний файл прихованих маркерів, стійких до різних атак. ЦВЗ повинні бути надійними, стійкими до спотворень файлу (масштабування, обертання, компресія з втратами та ін.) та мати невеликий обсяг [13].

Для відновлення водяного знака із захищеного файлу при відомих параметрах синтезу (зокрема, просторової несучої) досить виконати двовимірне перетворення Фур'є.

Прихований маркер (мітка) це вузькополосний сигнал в широкому діапазоні частот маркованого зображення. Така мітка створюється за допомогою модифікацій двох різних алгоритмів: приховування інформації шляхом фазової модуляції інформаційного сигналу з псевдовипадковою послідовністю чисел або ж поділом наявного діапазону частот на кілька каналів (передача проводиться між цими каналами).

Щодо вихідного зображення мітка це певний додатковий шум, але за рахунок того, що шум в сигналі присутній завжди, його невелике зростання за рахуноквпровадження мітки не дає помітних на око спотворень. Крім того, мітка

розсіюється по всьому вихідному зобрежнні, в результаті чого стає більш стійкою до вирізання.

Декілька слів про існуючі програми для стеганографії, які працюють з різними типами файлів, розміщених в Інтернеті.

- gifshuffle 2.0. Ця програма використовується для приховування повідомлень взображеннях формату GIF, перетасовуючи colourmap, і залишаючи зображення явнонезмінним. Gifshuffle працює з усіма GIF-форматами, включаючи зображення зпрозорістю і анімацією і, що важливо, передбачає стиснення і кодуванняприхованого повідомлення. Працює на платформах Windows.
- Hide and Seek 5.0. Ця стегопрограмма приховує будь-які дані в GIFзображенні. Дані шифруються за допомогою blowfish-алгоритму. Працює наплатформах Windows.
- F5-програма використовується для приховування файлів в BMP, GIF і JPEGзображеннях. Приховувані дані можна захистити паролем. Працює наплатформах Windows.
- Steghide 0.5.1. Це стеганографічна програма, яка приховує біти файлуданих в деяких з найменш істотних бітів іншого файлу такимспособом, що існування файлу даних невидимо і не може бути доведено.Програма використовує blowfish-кодування, MD5 hashing passphrases на blowfish-ключі, і псевдовипадковий розподіл прихованих бітів в контейнерних даних.Працює на платформах Linux, UNIX, Solaris і.т.п.
- bmpPacker. Це утиліта для шифрування / дешифрування інформації, якапоміщає зашифрований файл всередині bmp-файлу (картинки). bmpPackerпідтримує алгоритми Blowfish, Twofish і Rijndael. Працює на платформі Windows.
- SecurEngine Professional 1.0. Це безкоштовна програма, що дозволяєприховувати різні файли в графічних зображеннях або текстових фалах. Усі приховувані дані попередньо шифруються. Працює на платформі Windows.
- appendX v 4.0. Це стеганографічний інструмент, який додаєдані в кінець до інших файлів (подібно JPEG або PNG), щоб приховатиінформацію. Написаний

на мові PERL.

- Masker 7.0. Ця програма дозволяє приховувати повідомлення в зображеннях,причому підтримується величезна кількість форматів, серед яких є як форматипрямого кодування, так і стискаючі (JPEG).
- І, нарешті, Steganos Security Suite, одна з найбільш відомих на поточниймомент програм.

2.7 Хеш-функції

На сьогоднішниій деньмайжежоденсервіс криптографії не може обійтись без використання хеш-функцій.Це функції, які призначені для «стиснення» будь якогонабору даних або повідомлення, котрі записані, як правило, в двійковому алфавіті, в деяку бітову комбінацію фіксованої довжини, котру називаютьзгорткою. Хеш-функції мають велику кількість застосування, при тестуванні логічних пристроїв, перевірці цілісності записів в базах даних, при проведенні статистичних експериментів та при побудові алгоритмів швидкого пошуку. Рівномірність розподілу їх значень при випадковому виборі значень аргументу є основною вимогою до хеш-функцій.

Криптографічною хеш-функцією можна назвати будь-яку хеш-функцію, яка є крипостійкою, тобто задовольняє специфічні для криптографічних додатків вимоги. Хеш-функції у криптографії застосовуються для вирішення наступних завдань:

- автентифікація джерела даних;
- побудови систем контролю цілісності даних при їх передачі або зберіганні.

Хеш-функцією називають будь-яку функцію$h:X \to Y$, легко обчислювану та таку, котра для будь-якого повідомлення M значення $h(M) = H$ (згортка) має фіксовану бітову довжину. X безліч всіх повідомлень, Y безліч двійкових векторів фіксованої довжини.

Зазвичай, хеш-функції будують на основі так званих однокрокових стискаючих функцій $y = f(x1, x2)$ двох змінних, де $x1$, $x2$ і y виконавчі вектори

довжини *m*, *n* і *n* відповідно, причому *n* довжина згортки, а *m* довжина блоку повідомлення. Для отримання значення *h (M)* повідомлення спочатку розбивається на блоки довжиною*m* (але, якщо довжина повідомлення не кратна *m* то останній блок певним чином доповнюється до повного), а потім до отриманих блоків*M1, M2, .., MN* застосовують наступну послідовну процедуру обчислення згортки:

$$H_o = v,$$
$$H_i = f(M_i, H_{i-1}),\ i = 1,.., N,$$
$$h(M) = H_N$$

Тут *V* певна константа, часто називається ініціалізуючим вектором. Її обераютьз різних міркувань і вона може являти собою секретну константу чи набір випадкових даних (наприклад, вибірку дати і часу).При такому підході властивості хеш-функції будуть повністю визначатися властивостями однокрокового стискання функції [14].

Виділяють два основних види криптографічних хеш-функцій –безключаові і ключові. Ключові хеш-функції ще називають кодами автентифікації повідомлень. Вони надають можливість без додаткових засобів гарантувати як цілісність даних в системах,де користувачі довіряють один одному так і достовірність джерела даних.

Безключовимихеш-функціями називають коди виявлення помилок. Вони дають можливість за допомогою додаткових засобів (наприклад, шифрування) гарантувати цілісність даних. Ці хеш-функції можуть застосовуватися в системах, в яких користувачі довіряють один-одному, так і не довіряють.

До хеш-функцій основною вимогою є рівномірність розподілу їх значень при випадковому виборі значень аргументу. Для криптографічних хеш-функцій також важливо, щоб значення функції сильно змінювалосяпри щонайменшій зміні аргументу. Це можна назвати лавинним ефектом.

До основних функцій хешування зазвичай ставлять такі вимоги:

- відсутність можливостімодифікації;

- відсутність можливостіфабрикації.

Перша вимога означаєвисоку складність підбору для заданого повідомлення з відомим значенням згортки іншого повідомлення з правильним значенням згортки.

Друге високу складність підбору повідомлення з правильним значенням згортки.

До безключової функцій ставлять наступні вимоги:

- односпрямованість;
- стійкість до знаходження другого прообразу;
- стійкість до колізій.

Під односпрямованістю мається на увазівисока складність знаходження повідомлення за заданим значенням згортки. Слід зауважити, що на даний момент немає використовуваних хеш-функцій, де була би доведена односпрямовність.

Під стійкістю до знаходження другого прообразу розуміють складність знаходження другого повідомлення з аналогічним значенням згортки для заданого повідомлення з відомим значенням згортки.

Під стійкістю до колізій розуміють складність знаходження пари повідомлень з аналогічними значеннями згортки. Частіше за всеу криптоаналітиків служить першим сигналом старіння алгоритму і необхідності його швидкої замінисаме знаходження способу побудови колізій.

Популярні хеш-алгоритми:

- CRC16/32 контрольна сума (не є криптографічним перетворенням).
- MD2/4/5/6 -творіння одного з авторів алгоритму RSA, Рона Райвеста.
- MD5 колись був дуже популярним, але перші передумови злому з'явилися ще в кінці 1990-х, і в даний момент його популярність швидкозгасає.
- MD6 досить цікавий алгоритмз конструктивної точки зору. Був висунутий на конкурс SHA-3, але автори не встигли довести його до робочого прототипу, та цей алгоритм відсутній в списку кандидатів, які пройшли у другий раунд.
- SHA широко поширені зараз алгоритми. Відбувається активний перехід

від SHA-1 до стандартів версії SHA-2. SHA-2 загальна назва групи алгоритмів SHA224, SHA256, SHA384 і SHA512. SHA224 та SHA384 – в цілому є аналогами для SHA256 та SHA512 відповідно, але лише після самого розрахунку згортки частина даних в ній відкидається. Використовувати їх є сенс лише для забезпечення сумісності з застарілим обладнанням.

Висновок до 2 розділу

В цьому розділі було розглянуто базові методи криптографічного захисту інформації, які бувають класифікації шифрів, в чому їх основні відмінності, розібрали головні переваги та недоліки.

Хоч і існують абсолютно стійкі системи шифрування, однак вони дуже не зручні і вимагають великих витрат при використанні. Жодна із широко використовуваних на практиці систем шифрування не є абсолютно стійкою. На даний момент одні з найкращих показників стійкості мають симетричні компоновані системи шифрування, де використовуються по черзі алгоритми перестановки та заміни.

РОЗДІЛ 3
ПРАКТИЧНЕ ВИКОРИСТАННЯ КРИПТОГРАФІЇ В ІНФОРМАЦІЙНИХ МЕРЕЖАХ

3.1 Протоколи SSL/TLS та їх використання на веб-сервері

Все частіше і частіше згадуються в останній час протоколи TLS і SSL, більш поширеним стає використання цифрових сертифікатів, та навіть було створеноорганізації, котрі готовінадавати цифрові сертифікати всім бажаючим безкоштовно, для того, щоб гарантувати шифрування трафіку між сторінками в інтернеті таклієнтським браузером. Необхідно це, звісно, для безпеки, щоб в мережі ніхто не міг отримати доступ до даних, які передаються від сервераклієнту тав зворотньому напрямку.

SSL Secure Socket Layer, дослівно рівень захищених сокетів. TLS Transport Layer Security, дослівно безпека транспортного рівня. SSL є технологією, що виникла першою, TLS було створено пізніше та він взяв за основу специфікації SSL 3.0, котрі були розроблені компанією Netscape CommunicationsCorporations. Проте, дані протоколи мають одне і те ж завдання – забезпечити захищену передачу даних між двома комп'ютерами в інформаційній мережі. Таку передачу використовують для різних сервісів, таких як : сайти, електронна пошта, обмін повідомленнями та інших.

Захищена передача даних забезпечується за допомогою шифрування і автентифікаціїпереданої інформації. В цілому, ці протоколи, TLS і SSL, працюють однаково, суттєвих відмінностей немає. TLS, якщо так можна сказати, є наступником SSL, однак вони можуть використовуватисяодночасно, причому навіть на одному і тому ж сервері. Така підтримка необхідна для того, щоб можна було забезпечити роботу як з новими клієнтами (пристроями та браузерами), так і з застарілими, які TLS не підтримують. Хронологія виникнення цих протоколів виглядає ось так:

- SSL 1.0 –не було опубліковано;
- SSL 2.0 лютий 1995 року;
- SSL 3.0 1996 рік;

- TLS 1.0 січень 1999 року;
- TLS 1.1 квітень 2006 року;
- TLS 1.2 серпень 2008 року;
- TLS 1.3 – серпень 2018 року.

Принцип роботи SSL і TLS, як вже було сказано, аналогічний. Поверх протоколу TCP/IP встановлюється зашифрований канал, всередині якого передаються дані через прикладний протокол HTTP, FTP, і так далі. Ось як це можна уявити графічно:

Рисунок 3.1 – Протокол SSL/TLS

Прикладний протокол «загортається» у TLS / SSL, а той в свою чергу в TCP/IP. В цілому, дані по прикладному протоколу передаються по TCP/IP, але вони зашифровані. І дешифрувати дані, що передаються може лише та машина, якою було встановлене з’єднання. Для всіх інших, хто отримає передані пакети, ця інформація не матиме сенсу, якщо вони не зможуть її розшифрувати [15].

Установка з'єднання відбувається в кілька етапів:

1) Клієнт встановлює з'єднання з сервером і запитує захищене підключення за одним з протоколів. Це може забезпечуватися або встановленням з'єднанням на порт, який спочатку призначений для роботи з SSL/TLS, наприклад, 443, або додатковим запитом від клієнта на встановлення захищеного з'єднання після встановлення звичайного;

2) При встановленні з'єднання клієнт надає список алгоритмів

шифрування, які йому доступні. Сервер звіряє отриманий список зі списком алгоритмів, які доступні самому серверу, і вибирає найбільш надійний алгоритм, після чого повідомляє клієнту, який алгоритм буде використовуватись;

3) Сервер надсилає клієнту свій цифровий сертифікат, підписаний підтверджуючимцентром, і відкритий ключ самого сервера;

4) Клієнт може зв'язатися з сервером довіреного центру сертифікації, котрий підписав сертифікат сервера, та перевірити, чи валідовано сертифікат сервера. Але може і не зв'язуватися. В операційній системі зазвичай вже встановлені кореневі сертифікати авторизації, з якими звіряють підписи серверних сертифікатів, наприклад, браузери.

5) Генерується сеансовий ключ для захищеного з'єднання. Це робиться в такий спосіб:

- клієнт генерує випадкову цифрову послідовність;
- клієнт шифрує її за допомогою відкритого ключа сервера і посилає результат на сервер;
- сервер дешифрує отриману послідовність за допомогою закритого ключа.

Оглядаючись на те, що алгоритм шифрування є асиметричним, розшифрувати послідовність може лише сервер. При використанні асиметричного шифрування використовується два ключі –публічний та приватний. Публічним ключем повідомлення шифрується, а приватним дешифрується. Дешифрувати повідомлення, маючи лише публічний ключ, не можна.

Таким чином встановлюється зашифроване з'єднання. Дані, що передаються по ньому, шифруються і дешифруються до тих пір, поки не буде розірвано з’єднання.

Ранішо було згадано про кореневий сертифікат. Це сертифікат авторизаціонного центру, підпис яким підтверджує те, щопідписаний сертифікат, є саме тим, котрий належить до віповідного сервісу. В самому сертифікаті зазвичай міститься ряд інформаційних полів, в яких вказано інформацію про ім'я серверу, якому надано сертифікат, а також терміни дії цього сертифікату. Якщо термін дії сертифікату скінчився, він визнається недійсним.

Для отримання підписаного серверного сертифікату необхідно згенерувати запит на підпис (Certificate Sign Request) і відправити цей запит авторизаційному центру, котрий поверне підписаний сертифікат, що встановлюється безпосередньо на сервері, трохи нижче подивимося, як це можна зробити самостійно. Спочатку генерується ключ для шифрування, потім на підставі цього ключа генерується запит на підпис, тобто сам CSR-файл [15].

Клієнтський сертифікат може бути згенеровано як для використання в пристроях, так і для використання користувачами. Зазвичай, такі сертифікати використовуються при двосторонній верифікації, коли клієнт верифікує, що сервер дійсно той, за кого себе видає, і сервер у відповідь робить аналогічну процедуру. Ця взаємодія називається двосторонньою автентифікацією (mutual authentication). Двостороння автентифікація дозволяє не лише підвищити рівень безпеки в порівнянні з односторонньою, а також може служити заміною автентифікації з використанням логіна і пароля.

Розглянемо на на практиці, як відбуваються дії, пов'язані з генерацією сертифікатів, з самого початку, і при цьому на власному прикладі.

Перше, що необхідно зробити це згенерувати кореневий сертифікат. Кореневий сертифікат підписується самим собою. А потім вже цим сертифікатом будуть підписуватися інші сертифікати.

```
$ openssl genrsa -out root.key 2048
Generating RSA private key, 2048 bit long modulus
.....+++
.....+++ e is 65537 (0x10001)

$ openssl req -new -key root.key -out root.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
```

There are quite a few fields but you can leave some blank For some fields there will be a default value,

If you enter '.', the field will be left blank.

Country Name (2 letter code) [AU]:UA

State or Province Name (full name) [Some-State]:N/A Locality Name (eg, city) []:Kyiv

Organization Name (eg, company) [Internet Widgits Pty Ltd]:My Company

Organizational Unit Name (eg, section) []:IT Service

Common Name (e.g. server FQDN or YOUR name) []:My Company Root Certificate

Email Address []:it@mycompany.com

Please enter the following 'extra' attributes to be sent with your certificate request

A challenge password []:

An optional company name []:My Company

$ openssl x509 -req -days 3650 -in root.csr -signkey root.key -out root.pem

Signature ok

subject=/C=RU/ST=N/A/L=Saint-Petersburg/O=My Company/OU=IT Service/CN=My Company Root Certificate/emailAddress=it@mycompany.com Getting Private key

Таким чином було згенеровано спочатку сам приватний ключ, потім запит на підпис, а потім своїим ключем підписали свій же запит і отримали власний цифровий сертифікат, виданий на 10 років. Пароль (challenge password) не обов'язково вводити при генерації сертифікату.

Приватний ключ **обов’язково** необхідно зберігати в надійному місці, отримавши його зловмисним зможе підписати від вашого імені будь-який сертифікат. А отриманий кореневий сертифікат можна використовувати для ідентифікації того, що сертифікат, наприклад, сервера підписано саме нами, а не кимось іншим. Саме такі дії виконують авторизовані центри при генерації

власних сертифікатів. Після процесу створення кореневого сертифіката можна приступити до генерації сертифіката сервера.

Вміст сертифіката можна переглянути таким чином:

$ openssl x509 -noout -issuer -enddate -in root.pem

issuer= /C=UA/ST=N/A/L=Kyiv/O=My Company/OU=IT Service/CN=My Company Root Certificate/emailAddress=admin@mycompany.com

notAfter=Jan 22 11:49:41 2025 GMT

Для підпису сертифіката для сервера необхідно виконати наступні дії:

1) згенерувати ключ;

2) згенерувати запит на підпис;

3) підписати самостійно або надіслати CSR-файл в авторизаційний центр.

Серверний сертифікат може містити ланцюжок сертифікатів, якими підписано сертифікат сервера, але її можна також зберігати в окремому файлі. В принципі, виглядає все приблизно так само, як і при генерації кореневого сертифіката [15].

$ openssl genrsa -out server.key 2048

Generating RSA private key, 2048 bit long modulus

.....

.....+++

.....+++ e is 65537 (0x10001)

$ openssl req -new -key server.key -out server.csr

You are about to be asked to enter information that will be incorporated

into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value, If you enter '.', the field will be left blank.

Country Name (2 letter code) [AU]:UA

State or Province Name (full name) [Some-State]:N/A Locality Name (eg, city) []:Kyiv

Organization Name (eg, company) [Internet Widgits Pty Ltd]:My Company

Organizational Unit Name (eg, section) []:IT Service

Common Name (e.g. server FQDN or YOUR name) []:www.mycompany.com Email Address []:admin@mycompany.com

Please enter the following 'extra' attributes to be sent with your certificate request

A challenge password []:

An optional company name []:

$ openssl x509 -req -in server.csr -CA root.pem -CAkey root.key CAcreateserial -out server.pem -days 365

Signature ok

subject=/C=UA/ST=N/A/L=Kyiv/O=My Company/OU=IT Service/CN=www.mycompany.com/emailAddress=admin@mycompany.com Getting CA Private Key

$ openssl x509 -noout -issuer -subject -enddate -in server.pem issuer= /C=UA/ST=N/A/L=Kyiv/O=My Company/OU=IT Service/CN=My Company Root Certificate/emailAddress=it@mycompany.com

subject= /C=UA/ST=N/A/L=Kyiv/O=My Company/OU=IT Service/CN=www.mycompany.com/emailAddress=admin@mycompany.com notAfter=Jan 25 12:22:32 2016 GMT

Таким чином сертифікат сервера було підписано та ми будемо знати, якою саме організацією видано цей сертифікат. Після підпису готовий сертифікат можна використовувати за призначенням, наприклад, встановити на веб-сервер.

Для встановлення SSL/TLS-сертифікату на веб-сервер nginx треба виконати кілька простих кроків:

a. Скопіювати файли .key і .pem на сервер. У різних операційних системах

ключі і сертифікати можуть зберігатися в різних директоріях. BUbuntu, наприклад, це директорія /etc/ssl/certs для сертифікатів і

/etc/ssl/private для ключів. У RedHat це /etc/pki/tls/certs та

/etc/pki/tls/private.

b. Прописати необхідні налаштування в конфігураційний файл для хоста. Ось

як це приблизно має виглядати:

3) Створити віртуальний хост, який буде прослуховувати 443 порт. Конфігурація буде виглядати приблизно так:

</FilesMatch>

<Directory /usr/lib/cgi-bin>

SSLOptions +StdEnvVars

</Directory>

BrowserMatch "MSIE [2-6]" \ nokeepalive ssl-unclean-shutdown \ downgrade-1.0 force-response-1.0

BrowserMatch "MSIE [17-9]" ssl-unclean-shutdown

</VirtualHost>

</IfModule>

4) Перезапустити веб-сервер Apache.

Клієнтський сертифікат створюється приблизно так само, як серверний.

$ openssl genrsa -out client.key 2048

Generating RSA private key, 2048 bit long modulus

.....+++

.....+++ e is 65537 (0x10001)

$ openssl req -new -key client.key -out client.csr

You are about to be asked to enter information that will be incorporated

into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank For some fields there will be a default value,

If you enter '.', the field will be left blank.

Country Name (2 letter code) [AU]:UA

State or Province Name (full name) [Some-State]:Kyiv Locality Name (eg, city) []:^C

kisha@kisha-desktop:~/Temp/certs/CA$ openssl req -new -key client.key -out client.csr

You are about to be asked to enter information that will be incorporated

1) into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank For some fields there will be a default value,

If you enter '.', the field will be left blank.

Country Name (2 letter code) [AU]:UA

State or Province Name (full name) [Some-State]:N/A Locality Name (eg, city) []:Kyiv

Organization Name (eg, company) [Internet Widgits Pty Ltd]:My Company

Organizational Unit Name (eg, section) []:Engineering

Common Name (e.g. server FQDN or YOUR name) []:Mikhail Kornieiev Email Address []:kornieiev@mycompany.com

Please enter the following 'extra' attributes to be sent with your certificate request

A challenge password []:

An optional company name []:

$ openssl x509 -req -in client.csr -CA root.pem -CAkey root.key CAcreateserial -out client.pem -days 365

Signature ok

subject=/C=UA/ST=N/A/L=Kyiv/O=My Company/OU=Engineering/CN=Ivan Ivanov/emailAddress=kornieiev@mycompany.com

Getting CA Private Key

$ openssl x509 -noout -issuer -subject -enddate -in client.pem issuer= /C=UA/ST=N/A/L=Kyiv/O=My Company/OU=IT Service/CN=My Company Root Certificate/emailAddress=it@mycompany.com

subject= /C=UA/ST=N/A/L=Kyiv/O=My Company/OU=Engineering/CN=Ivan Ivanov/emailAddress=kornieiv@mycompany.com

notAfter=Oct 25 13:17:15 2020 GMT

Отже, тепер можна використовувати клієнтський сертифікат для роботи з нашим сервером [15].

При використанні SSL/TLS одним з основних методів злому є метод MITM (Man In The Middle), «людина посередині». Цей метод грунтується на використанні серверного сертифіката і ключа на якомусь вузлі, котрий буде прослуховувати трафік і дешифрувати інформацію, якою обмінюються клієнт і сервер. Для організації прослуховування можна використовувати, наприклад, програму sslsniff. Тому кореневий сертифікат і ключ зазвичай бажано зберігати на пристрої, котрий не підключено до мережі, для підписання підключати запити на підпис на флешці, підписувати і так само відключати. Та, звісно, робити резервні копії.

3.2 Протокол SSH

SecureShell (SSH) широко використовуваний протокол транспортного рівня для захисту з'єднань між клієнтами і серверами. Це базовий протокол захищеного доступу до інфраструктури. Нижче наведено короткий опис рукостискання, яке відбувається перед встановленням безпечного каналу між клієнтом і сервером і перед початком повного шифрування трафіку.

Рукостискання починається з того, що обидві сторони посилають один одному рядок з номером версії. У цій частині рукостискання не відбувається нічого особливого, але слід зазначити, що більшість щодо сучасних клієнтів і серверів підтримують лише SSH 2.0 через недоліки в дизайні версії 1.0.

В процесі обміну ключами (іноді званого KEX) сторони обмінюються загальнодоступною інформацією і виводять секрет, яким користуються клієнтом і сервером. Цей секрет неможливо виявити або отримати із загальнодоступної інформації.

Обмін ключами починається з того, що обидві сторони посилають один одному повідомлення SSH_MSG_KEX_INIT зі списком підтримуваних криптографічних примітивів і їх переважним порядком.Криптографічні примітиви повинні встановити будівельні блоки, які будуть використовуватися для обміну ключами, а потім повного шифрування даних.

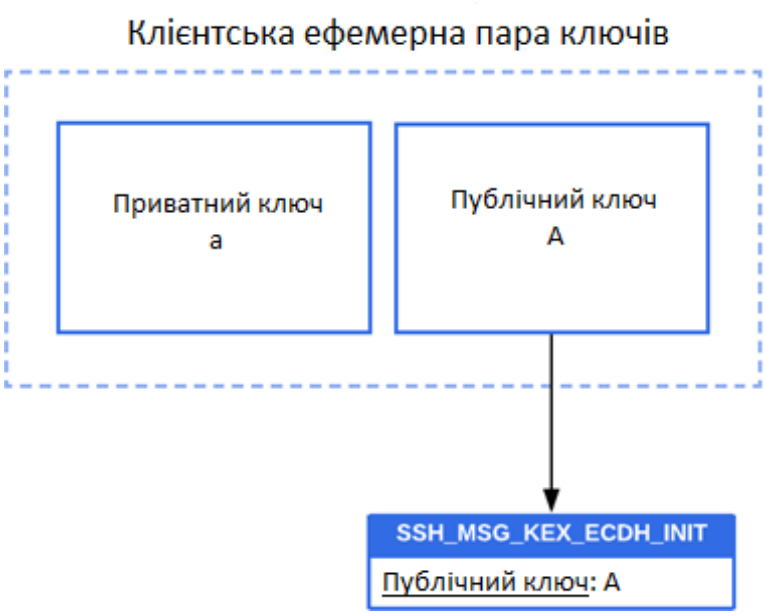

Рисунок 3.2 – Генерація повідомлення ініціалізації обміну ключами

Оскільки обидві сторони використовують один і той же алгоритм для вибору криптографічних примітивів зі списку підтримуваних, після ініціалізації можна негайно почати обмін ключами. За приклад візьмемо протокол еліптичних кривих Діффі-Хеллмана (EllipticCurveDiffie-Hellman, ECDH), так що обмін ключами починається з того, що клієнт генерує ефемерну пару ключів (закритий і пов'язаний з ним відкритий ключ) і відправляє серверу свій відкритий ключ у повідомленні SSH_MSG_KEX_ECDH_INIT.

Варто підкреслити, що ця ключова пара ефемерна: вона використовується тільки для обміну ключами, а потім буде видалена. Це надзвичайно ускладнює проведення класу атак, де зловмисник пасивно записує зашифрований трафік з надією вкрасти закритий ключ коли-небудь в майбутньому. Дуже важко вкрасти те, чого більше не існує. Це властивість називається прямою секретністю (forward secrecy).

Сервер чекає повідомлення SSH_MSG_KEX_ECDH_INIT, а при отриманні генерує власну ефемерну пару ключів. За допомогою відкритого ключа клієнта і власної пари ключів сервер може згенерувати загальний секрет K[16].

Потім сервер генерує дещо, що можна назвати хешем обміну H, і підписує його, генеруючи підписаний хеш HS. Хеш обміну і його підпис служать декільком цілям:

- оскільки хеш обміну включає загальний секрет, він доводить, що інша сторона змогла створити загальний секрет;
- цикл підпису / перевірки хеша і підписи обміну дозволяють клієнту перевірити, що сервер володіє закритим ключем хоста, і тому клієнт підключений до правильного сервера (якщо клієнт може довіряти відповідному відкритому ключу);
- за рахунок підпису хеша замість підпису вхідних даних, розмір підписаних даних істотно зменшується і призводить до більш швидкого рукостискання.

Хеш обміну генерується шляхом взяття хеша (SHA256, SHA384 або SHA512, в залежності від алгоритму обміну ключами) наступних полів:

- Меджік M. Версія клієнта, версія сервера, повідомлення клієнта SSH_MSG_KEXINIT, повідомлення сервера SSH_MSG_KEXINIT;
- Відкритий ключ (або сертифікат) хоста сервера HPub. Це значення (і відповідний йому закритий ключ HPriv) зазвичай генерується під час ініціалізації процесу, а не для кожного рукостискання;
- Клієнтський відкритий ключ A;
- Серверний відкритий ключ B;
- Загальний секрет K.

З цією інформацією сервер може сконструювати повідомлення SSH_MSG_KEX_ECDH_REPLY, застосувавши недовговічний відкритий ключа сервера B, відкритий ключ хоста сервера HPub і підпис на хеше обміну HS.

Як тільки клієнт отримав від сервера SSH_MSG_KEX_ECDH_REPLY, у нього є все необхідне для обчислення секрету K і хеша обміну H.

В останній частині обміну ключами клієнт отримує відкритий ключ хоста (або сертифікат) з SSH_MSG_KEX_ECDH_REPLY і перевіряє підпис хеша обміну HS, що підтверджує право власності на закритий ключ хоста. Щоб запобігти атакам типу «людина в посередині» (MitM), після перевірки підпису відкритий ключ хоста (або сертифікат) перевіряється по локальній базі відомих хостів; якщо цей ключ (або сертифікат) не є довіреною, з'єднання розривається.

The authenticity of host 10.10.10.10 (10.10.10.10)' can't be established.

ECDSA key fingerprint is SHA256: pnPn3SxExHtVGNdzbV0cRzUrtNhqZv +Pwdq/qGQPZO3. Are you sure you want to continue connecting (yes/no)?

SSH-клієнт пропонує додати ключ хоста в локальну базу відомих хостів.

BOpenSSH це зазвичай ~/.ssh/known_hosts.

На цьому етапі обидві сторони погодили криптографічні примітиви, обмінялися секретами і згенерували матеріал ключів для обраних примітивів. Тепер між клієнтом і сервером може бути встановлений безпечний канал, який здатний забезпечить конфіденційність і цілісність.

Ось як рукостискання SSH встановлює безпечне з'єднання між клієнтами і серверами.

3.3 Електронний підпис

Електронний цифровий підпис це криптографічний механізм, який використовується для перевірки автентичності та цілісності цифрових даних. Ми можемо розглядати його як цифрову версію звичайних рукописних підписів, але з більш високим рівнем складності і безпеки.Висловлюючись простими словами, ми можемо описати електронний підпис як код прикріплений до повідомлення або документа. Після його генерації він виступає в якості доказу того, що повідомлення не було підроблено протягом свого шляху від відправника до одержувача.

Концепція захисту комунікаційних каналів зв'язку з використанням криптографії бере свій початок ще з давніх часів, а системи з цифровим підписом з'явилися тільки в 1970-х роках завдяки розвитку криптографії з відкритим ключем.

Першим кроком є хешування повідомлення або цифрових даних. Це робиться шляхом обробки інформації за допомогою алгоритму хешування для генерації безпосередньо самого хеша (дайджест повідомлення). Повідомлення можуть значно відрізнятися за своїм розміром, проте після їх хеширования все хеші будуть володіти однаковою довжиною. Це одна із самих основних властивостей хеш-функції.

Проте, хешування даних не є обов'язковою умовою для створення електронного підпису, оскільки замість цього можна використовувати приватний ключ для того, щоб підписати повідомлення. Але якщо мова йде про криптовалюту, дані завжди хешуются, оскільки робота з дайджестами фіксованої довжини спрощує весь процес обробки інформації [17].

Після хешування даних відправник повідомлення повинен підписати його, і саме в цей момент вступає в гру криптографія з відкритим ключем. Існує кілька видів алгоритму цифрового підпису, кожен з яких має свій унікальний механізм. Але хешуване повідомлення в будь-якому випадку буде підписано приватним ключем, а одержувач потім зможе перевірити його справжність за допомогою відповідного публічного ключа (наданого підписуючою особою).

Іншими словами, якщо приватний ключ не включений при створенні підпису, одержувач повідомлення не зможе використовувати відповідний публічний ключ для перевірки його дійсності. Оскільки публічний і приватний ключі генеруються відправником повідомлення, тільки публічний використовується спільно з одержувачем.

Варто відзначити, що цифрові підписи безпосередньо взаємопов'язані з вмістом кожного повідомлення. Таким чином, на відміну від рукописних підписів, які як правило однакові незалежно від контексту документа, кожне повідомлення з цифровим підписом буде мати зовсім іншим цифровим кодом підпису.

Давайте розглянемо це на прикладі для того, щоб краще проілюструвати весь процес до останнього кроку перевірки вмісту. Уявіть, що абонент А пише повідомлення абоненту Б, хешує його, а потім об'єднує хеш зі своїм приватний ключем для створення цифрового підпису. В даному випадку підпис виступає в якості унікального цифрового ідентифікатора конкретно цього повідомлення.

Коли абонент Б отримує повідомлення, він може перевірити достовірність цифрового підпису за допомогою публічного ключа наданого абонентом А. Таким чином, абонент Б може переконатися в тому, що підпис був створений саме абонентом А, оскільки тільки абонент А є володарем відповідного приватного ключа (принаймні так повинно бути).

З цієї причини для абонента А вкрай важливо зберігати свій приватний ключ в секреті. Якщо інша людина заволодіє її приватним ключем, він зможе створювати цифрові підписи і здійснювати операції від його імені.

В основному цифрові підписи призначені для досягнення трьох результатів: цілісності даних, автентичності і неотрекаємості.

- Цілісність даних. Абонент Б може упевнитися, що повідомлення абонента А не змінювався протягом свого шляху. Наслідком будь-яких змін в повідомленні буде генерація зовсім інший підпису.
- Автентифікація. Поки приватний ключ абонента А зберігається в секреті, абонент Б може використовувати публічний ключ абонента А, щоб підтвердити факт того, що цифрові підписи були створені саме абонентом А і ніким іншим.

- Невідрікаємість. Після того, як підпис було згенеровано, абонент А не зможе заперечувати своє ставлення до нього в майбутньому, тільки в разі, якщо його приватний ключ був якимось чином скомпрометований.

Електронні підписи можуть застосовуватися до різних видів цифрових документів і сертифікатів. Таким чином, у них є кілька напрямків, деякі з найбільш поширених варіантів використання включають в себе:

- інформаційні технології, підвищення безпеки систем інтернет-комунікації;
- фінанси, аудит, звіти про витрати, кредитні договори і багато інших фінансових документів;
- юридичні питання, а саме використання в усіх видах ділових контрактів і юридичних угодах, включаючи урядові документи.
- охорона здоров'я та запобігання шахрайства з рецептами і медичними записами.
- блокчейн, де система електронних підписів гарантує, що тільки законні власники криптовалюти можуть підписати транзакцію для подальшого переказу коштів (за винятком випадків, коли приватний ключ власника було скомпрометовано).

Основні проблеми з якими може зіткнутися дана технологія залежить принаймні від трьох складових:

- якість алгоритмів використовуваних для генерації цифрового підпису має вкрай важливе значення. Це включає в себе вибір надійних хеш-функцій та криптографічних систем;
- якщо алгоритми працюють правильно, а інтеграція технології цифрових підписів пройшла не зовсім успішно, система швидше за все буде мати певною кількістю недоліків;
- у разі витоку приватних ключів або з якоїсь причини вони були скомпрометовані, властивості автентифікації і невідрікаємості будуть визнані недійсними. Для користувачів криптовалюти втрата свого приватного ключа може привести до значних фінансових втрат.

3.4 Цифровий водяний знак

Цифровий водяний знак (ЦВЗ) це технологія для відвернення викрадень або використання цифрових зображень, аудіо та відео без дозволу власника. ЦВЗ є впровадження цифрового підпису в дані. Існує два класи цифрових водяних знаків видимі і невидимі.

Відомий водяний знак найкраще використовувати для даних, зоровий образ яких не псується при додаванні цифрового підпису. Перевагою таких водяних знаків є те, що дані захищені авторським правом і їх повноцінне використання стає неможливим без видалення цифрового водяного знаку. Подібні заходи захисту значно спрощують суперечки з авторського права, оскільки наявність або факт видалення водяних знаків можна легко виявити.

Невидимий водяний знак використовується, коли зовнішній вигляд даних не може бути змінений. Невидимий ЦВЗ це спеціальна мітка, вбудована в цифровий контент, званий контейнером, з метою захисту авторських прав і підтвердження цілісності документа. Перевага такого типу водяних знаків полягає в тому, що їх не можна легко виявити. Потенційні порушники можуть використовувати дані, не підозрюючи, що вони містять маркування власника.

В даний час розроблено багато різних методів вбудовування невидимих ЦВЗ в зображення, наприклад, метод LSB (Last Significant Bit), метод псевдовипадкового інтервалу, метод псевдовипадкової перестановки (вибору), метод блочного приховування. Найбільш поширеним, але і найменш стійким до спотворень є метод заміни бітів молодших розрядів, що представляють яскравість / колір пікселя, або так званий LSB-метод. Суть методу LSB полягає в заміні молодших значущих бітів в байтах зображення (контейнері), що відповідають за кодування кольору, на біти приховуваного повідомлення. Основними перевагами даного методу є: 1) той факт, що людське око в більшості випадків не здатний помітити зміни в молодших бітах; 2) простота самого методу; 3) можливість приховувати в відносно невеликих зображеннях досить великі обсяги інформації. Основним недоліком методу LSB є його висока чутливість до спотворень контейнера. В роботі також позначається питання про визначення автентичності

документів, будь то фізичні документи (гроші, цінні і конфіденційні документи) або електронні документи (скановані копії, фотографії, електронний друкований текст). Це викликано збільшенням обсягу документообігу між організаціями, а також розвитком технологій обміну документами. Впроваджувані водяні знаки можна використовувати як для підтвердження дійсності документа, так і для вбудовування в них певної інформації.

В даний час при формуванні ЦВЗ застосовується принцип вбудовування мітки, яка представляє собою вузькосмуговий сигнал, в широкому діапазоні частот, що маркується зображення. Цей метод реалізується за допомогою двох різних алгоритмів. В одному алгоритмі інформація передається за допомогою фазової модуляції «несучої», що представляє собою псевдовипадкову послідовність чисел. В іншому весь діапазон частот ділиться на кілька субдіапазонів і передача проводиться між цими субдіапазонами. Відносно маркується зображення мітку можна розглядати як певний додатковий шум. Але так як в зображенні завжди присутній шум, то його незначне зростання за рахунок додавання мітки не призводить до помітного для зору збільшення спотворень. Крім того, сигнал, що представляє мітку, поширюється по всьому зображенню, завдяки чому досягається стійкість до обрізку зображення. Паралельний алгоритм навчання нейронної мережі з машиною опорних векторів в якості вихідного шару дозволяє автоматизувати процедуру формування тренувальних наборів при створенні систем розпізнавання зображень.

На рисунку 3.3 зображено фази життєвого циклу цифрового водяного знаку.

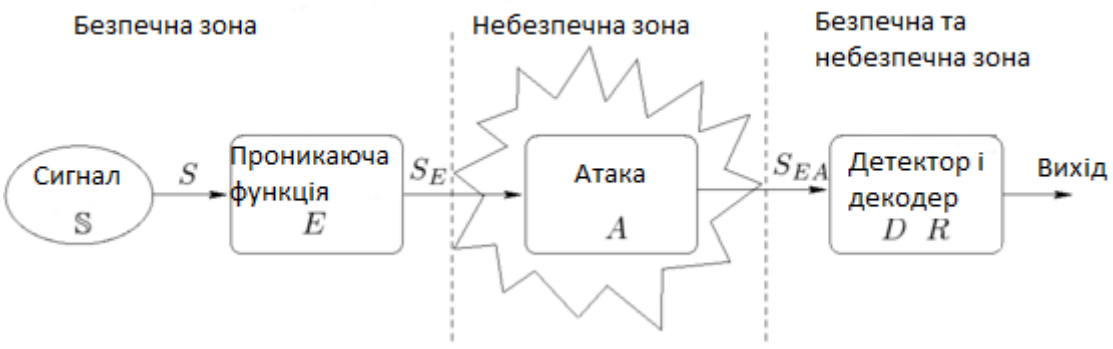

Рисунок 3.3 – Життєвий цикл цифрового водяного знаку

Найважливіше застосування ЦВЗ знайшли в системах захисту від копіювання, які прагнуть запобігти або утримати від несанкціонованого копіювання цифрових даних. Застосовують ЦВЗ в стеганографії (спосіб передачі або зберігання інформації з урахуванням збереження в таємниці самого факту такої передачі або зберігання), коли сторони обмінюються секретними повідомленнями, впровадженими в цифровий сигнал. Використовується як засіб захисту документів з фотографіями паспортів, водійських посвідчень, кредитних карт з фотографіями. Хоча деякі формати цифрових даних можуть також нести в собі додаткову інформацію, яка називається метаданими, ЦВЗ відрізняються тим, що інформація

«зашита» прямо в сигнал. Об'єкти мультимедіа в цьому випадку будуть представляти собою контейнери (носії) даних. Основна перевага полягає в наявності умовною залежності між подією підміни об'єкта ідентифікації та наявності елемента захисту прихованого водяного знака. Підміна об'єкта ідентифікації призведе до висновку про підробку всього документа [18].

Видимі прозорі водяні знаки менш вивчені. Їх виявлення є вельми важливим завданням.

3.5 Приклад стартапу з використанням криптографії

Данний стартап намагається зробити Інтернет більш безпечним, шляхом популяризації шифрування з відкритим ключем. Майже кожен день можна прочитати про черговий гучний злом. Останніми прикладами є злом конфіденційної ділової переписки Sony (в результаті вона була оприлюднена), а також злом Apple iCloud (в результаті приватні фотографії знаменитостей наповнили інтернет).

Хакери стають все більш витонченими, розкриваючи ресурси і системи безпеки, розроблені для застарілих загроз, настільки ж легко, як банку шпротів. Через це багато людей (втім цілком виправдано) починають сумніватися в довірі до технологічних компаній, що зберігають і обробляють приватну інформацію. Все це відбувається, незважаючи на те, що технологія, яка вирішує цю проблему

вже давно існує, і це, як ви вже напевно розумієте, криптографія з відкритим ключем. Шифрування з відкритим ключем було винайдено математиками і програмістами в 1970-х. Сьогодні важко переоцінити значення цього винаходу.

Ключовою перевагою шифрування з відкритим ключем є незалежність від постачальників інформаційних та телекомунікаційних послуг. До них відносяться поштові послуги, месенджери, соціальні мережі, пошукові системи, інтернетпровайдери, оператори стільникового зв'язку, політики, юридичні угоди і багато багато іншого. Використання цієї технології вимагає лише довіри до математики.

Чому ж ця технологія не використовується повсюдно? В тій чи іншій мірі різні форми криптографії використовуються майже в кожному популярному інтернет-сервісі, проте злом і компрометація даних продовжується. В основному це відбувається через недосконаліть вбудованих криптографічних протоколів, спеціально внесених програмних помилок, помилок співробітників, виробничої економії, правових обмежень і невірних управлінських рішень.

Ідеальним рішенням було б використання чистої технології, щоб не потрібно було сумніватися в сторонніх постачальниках. Сьогодні таким чином технологію використовують тільки найбільш технічно підковані користувачі мережі. Журналістка Fusion Кашмір Хілл, наприклад, твітнула свій відкритий ключ. Проте для відправки зашифрованого повідомлення необхідно використовувати програмні засоби, які зазвичай дуже складні і громіздкі для простих користувачів. В результаті технологія залишається притулком невеликого кола технічно просунутих ентузіастів.

Основна ідея стартапу полягає в тому, щоб зробити технологію доступною для звичайних користувачів. В основі стартапу є база даних.

Крім банальної відправки повідомлень криптографія з відкритим ключем може використовуватися в широкому спектрі завдань, починаючи від файлообміну і закінчуючи верифікацією програмного забезпечення і контролем модифікації вихідного коду. Цілком можливо використання її для автентичності на веб-сайтах, що може зробити непотрібним як звичайний парольний захист, так

і двухфакторную авторизацію, яка зараз широко впроваджується.

Крім своїх основних функцій зберігання данний стартап може включати в себе набір нативних додатків для різних платформ.

Висновок до 3 розділу

Ключовим принципом данного стартапу є те, що вам не потрібно довіряти сервісу, так як все необхідне програмне забезпечення з відкритим вихідним кодом, що дає можливості як для незалежної перевірки, так і для розробки форків, яким власне і є данний стартап. Так що все, що вимагає хоч найменшої довіри, піддається перевірці. Виходячи з усього цього зрозуміло, що щоб не сталося з самою платформою це ніяк не вплине на безпеку всього, що використовується стартапом.

Шифрування з відкритим ключем було в довгому ув'язненні в нішевих технологічних співтовариствах. Надто довго.

ЗАГАЛЬНІ ВИСНОВКИ

Метою дипломної магістерської роботи є дослідження можливості застосування криптографічних методів захисту в інформаційних мережах. В результаті вирішення поставлених задач, було досягнуто таких результатів:

1. Проведено аналітичний огляд криптографії, категорій криптографії, її історії та відносин з державою і суспільством. Огляд показав, що дана тема є актуальною на сьогоднішній день. Актуальність полягає у необхідності захисту всіх сфер життєдіяльності від небажаного доступу до інформації.

2. Досліджено базові методи криптографічного захисту інформації, такі як симетричні та асиметричні шифри, методи заміни та перестановки.

Перевагами симетричних криптосистем є швидкість (на три порядки вище асиметричних), простота реалізації, менша необхідна довжина ключа, вивченість. До недоліків можна віднести складність управління ключами у великій мережі, складність обміну ключами.

Перевагами асиметричних криптографічних систем перед симетричними крипто-системами є вирішена проблема складного розподілу ключів між користувачами, відсутність квадратичної залежності, можливість застосування протоколів взаємодії сторін, що не довіряють одне одному. До недоліків можна віднести те, що нема математичного доказу незворотності, набагато менша швидкість в порівнянні з симетричним шифруванням та необхідність захисту відкритих ключів від підміни.

3. Розглянуто принципи роботи криптографічних атак та методики захисту від них.

4. Також було розглянуто стеганографію, та можливість її використання в поєднанні з криптографією.

5. Проведено дослідження з налаштування захисту веб-серверів Nginx та Apache та створено клієнтські сертифікати TLS/SSL, для встановлення захищенного HTTPSз’єднання між клієнтом та сервером.

Робота показує, що протягом усієї своєї історії людству необхідно шифрування тієї чи іншої інформації. З такої потреби виросла ціла наука

криптографія. Раніше криптографія служила тільки інтересам держави, але з появою інтернету її методи стали цікавити і приватних осіб. На сьогоднішній день криптографія широко використовується не лише хакерами, але і борцями за свободу інформації, фінансовим сектором, воєнними структурами та простими користувачами, охочими захистити свої дані в мережі. Актуальність криптографії не згасне в найближчі століття.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Методи та засоби синтезу операцій перестановок, керованих інформацією, для комп'ютерних криптографічних систем. URL:https://www.slideshare.net/sitecdtu/i-75720925

2. Технологія захисту інформації. Проблеми захисту інформації в сучасних іс. Основні види сучасних комп'ютерних злочинів. Засоби захисту інформації.

URL: https://shag.com.ua/tema-tehnologiya-zahistu-informaciyi-problemi-zahistu-informac.

3. Вступ до криптології.URL: https://zavantag.com/docs/2307/index331503. html.

4. Криптография. История шифровального дела.

URL: https://news.rambler.ru/other/44425644-kriptografiya-istoriya-shifrovalnogodela/.

5. Історія криптографії.

URL: znaimo.com.ua/Історія_криптографії

6. Методи та засоби захисту інформації в мережі Інтернет.

URL: infopedia.su/8xb03c.html

7. Класифікація сучасних криптографічних методів. URL: https://lickeys.ru/uk/zhestkijj-disk/klassifikaciya-sovremennyhkriptograficheskih-metodov/

8. Симетричні криптосистеми – це.

URL: http://jak.bono.odessa.ua/articles/simetrichni-kriptosistemi-ce.php

9. Сучасні алгоритми шифрування. Методи шифрування даних. URL: https://polarize.ru/uk/grafika/sovremennye-algoritmy-shifrovaniya-metodyshifrovaniya-dannyh/

10. Шифрування методом заміни.

URL: http://ni.biz.ua/11/11_4/11_46236_shifrovanie-metodom-zameni-podstanovki.html

11. Методи перестановки.

URL: http://ni.biz.ua/3/3_3/3_37326_metodi- perestanovki.html

12. Криптографические атаки: объяснение для смятённых умов.

URL: https://habr.com/ru/post/462437/

13. Стеганографічний алгоритм захисту даних з використанням файлів зображень. URL: http://www.economy.nayka.com.ua/?op=1&z=5584

14. Розрахунок хеш функцій. Хеш-функції: поняття та основи. Що це таке. URL: https://utalux.ru/uk/raschet-hesh-funkcii-hesh-funkcii-ponyatie-i-osnovychto-eto-takoe/

15. Основи SSL. Основи SSL Двусторонне ssl з'єднання з платіжною системою.

URL:codoschool.ru/uk/mts-bank/osnovy-ssl-osnovy-ssl-dv...ie-splatezhnoi.html

16. Рукопожатие SSH простыми словами. URL: https://habr.com/ru/company/dcmiran/blog/474654/

17. Что такое цифровая подпись?

URL: https://academy.binance.com/ru/articles/what-is-a-digital-signature

18. Цифровий водяний знак.

URL: http://kolosok.lviv.ua/index/cifrovij/uk/vodanoj-cifrovij-vodanij-znak