Інформаційна безпека

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ ДЕРЖАВНИЙ ПОЛІТЕХНІЧНИЙ КОЛЕДЖ

Для спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» спеціалізації 5.151.2 «Обслуговування пристроїв електрозв'язку

Методичний посібник
для самостійної роботи студентів

з навчальної дисципліни «Інформаційна безпека»

2019 р.

Методичний посібник для самостійної роботи з дисципліни «Інформаційна безпека» студентів спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» спеціалізації 5.151.2 «Обслуговування пристроїв електрозв'язку»

/ Укладач: Слобожанюк Р.І., викладач вищої категорії– ХДПК, 2019. – 62 с.

Методичний посібник розглянутий та рекомендований цикловою комісією

«Транспорту та електрозв'язку»

(назва циклової комісії)

Протокол № _____ від "___"_____2019 р.

Голова циклової комісії _____ (Руденко А.І.)

(підпис) (прізвище та ініціали)

Схвалено методичною радою Харківського державного політехнічного коледжу

Протокол № _____ від "____"_____20____р.

Голова методичної ради _____ (Величко В.О.)

(Підпис) (прізвище та ініціали)

ВСТУП

Навчальний час, відведений на вивчення дисципліни, визначається навчальним планом спеціальності. Розподіл навчального часу за видами занять у тому числі на самостійну роботу , визначається робочою навчальною програмою.

Самостійна робота (СР) - це одна з складових навчального процесу. Вона дає можливість студентам підвищити свої знання та самостійно мислити.

Методичний посібник до самостійної роботи студентів з дисципліни призначений для допомоги студентам при роботі з матеріалом, який виділено робочою програмою на самостійне вивчення

Методичний посібник містить питання та короткий зміст тем, винесених на самостійне вивчення та літературу, якою студент має користуватися при вивченні кожної теми.

О'БЄМ ДИСЦИПЛІНИ ТА ВИДИ НАЧАЛЬНОЇ РОБОТИ

Вид навчальної роботи	Всього годин	Семестр 7
Всього годин на дисципліну	108	
Аудиторні заняття :	64	
лекції	58	
Практичні заняття	6	
Самостійна робота	44	
Форма поточного контролю		диференційований залік

№ п/п	**Форма занять**	**Обсяг годин**	**Зміст занять**
1.	Самостійна робота		Структура криптосистеми
2.	Самостійна робота		Прості методи криптографічного перетворення: шифри перестановки, шифри простої заміни, шифри складної заміни
3.	Самостійна робота		Практичне застосовування шифрів перестановки в системах зв'язку
4.	Самостійна робота		Задачі, що вирішуються криптографічними методами.
5.	Самостійна робота		Вимоги до реалізації сучасних криптоалгоритмів.
6.	Самостійна робота		Стандарт шифрування ГОСТ 28147-89
7.	Самостійна робота		Основні схеми шифрування в асиметричній системі шифрування
8.	Самостійна робота		Електронний підпис (ЕП). Історія виникнення . Призначення і застосування ЕП

9.	Самостійна робота		Використання механізму електронного цифрового підпису Алгоритм цифрового підпису
10.	Самостійна робота		Проблема автентифікації даних та електронний цифровий підпис.
11.	Самостійна робота		Технічний захист інформації. Основні терміни та визначення
12.	Самостійна робота		Захист інформації в оптоволоконних мережах
Разом			

Тема 1: Структура криптосистеми (2 години)

1 Вимоги до криптографічних систем

Процес криптографічного закриття даних може здійснюватися як програмно, так і апаратно. Апаратна реалізація відрізняється суттєво більшою вартістю, однак їй властиві й переваги: висока продуктивність, простота, захищеність і т.д.. Програмна реалізація більш практична, допускає відому гнучкість у використанні.

Для сучасних криптографічних систем захисту інформації сформульовані наступні загальноприйняті вимоги:

зашифроване повідомлення повинне піддаватися читанню тільки при наявності ключа;

число операції, необхідних для визначення використаного ключа шифрування по фрагменту шифрованого повідомлення й відповідного йому відкритого тексту, повинне бути не менше загального числа можливих ключів;

число операцій, необхідних для розшифровування інформації шляхом перебору всіляких ключів, повинне мати строгу нижню оцінку й виходити за межі можливостей сучасних комп'ютерів (з урахуванням можливості використання мережних обчислень) або вимагати неприйнятно високих витрат на ці обчислення;

знання алгоритму шифрування не повинне впливати на надійність захисту;

незначна зміна ключа повинна приводити до істотної зміни виду зашифрованого повідомлення навіть при шифруванні того самого вихідного тексту;

незначна зміна вихідного тексту повинне приводити до істотної зміни виду зашифрованого повідомлення навіть при використанні того самого ключа;

структурні елементи алгоритму шифрування повинні бути незмінними;

додаткові біти, що уводяться в повідомлення в процесі шифрування, повинні бути повністю й надійно сховані в шифрованому тексті;

довжина шифрованого тексту не повинна перевершувати довжину вихідного тексту;

не повинне бути простих і легко встановлюваних залежностей між ключами, послідовно використовуваними в процесі шифрування;

будь-який ключ із множини можливих повинен забезпечувати надійний захист інформації;

алгоритм повинен допускати як програмну, так і апаратну реалізацію, при цьому зміна довжини ключа не повинне вести до якісного погіршення алгоритму шифрування.

Не слід забувати й про такі банальні речі, як гроші. Іншими словами - скільки буде коштувати збиток, який понесе власник інформації, при несанкціонованім її використанні? чи Коштує ця інформація тих вкладень, які необхідно зробити для закриття інформації?

2 Модель криптографічної системи

Криптографічна система (криптосистема) – система секретного зв’язку, в якій зміст інформації, що передається, утаємничується за допомогою криптографічних перетворень; при цьому сам факт передачі інформації не приховується.

Криптографічні перетворення визначаються певним параметром, який називається ключ. Зазвичай ключ є буквеною або числовою послідовністю. Кожне криптографічне перетворення однозначно визначається ключем і описується певним криптографічним алгоритмом.

Криптографічними перетвореннями є:

Шифрування– процес перетворення вихідного тексту (P) в зашифрований текст (C) за допомогою шифруючої функції (E) з секретним ключем шифрування (Ke) у відповідності з обраним алгоритмом шифрування: $C=E_{Ke}(P)$.

Розшифрування – обернений шифруванню процес перетворення зашифрованого тексту (C) в вихідний текст (P) за допомогою функції розшифрування (D) з секретним ключем розшифрування (Kd) у відповідності з обраним алгоритмом шифрування: $P=D_{Kd}(C)$.

Сімейство обернених перетворень защифрування і розшифрування називають **шифром**.

Алгоритми шифрування і розшифрування можуть відрізнятись, відповідно можуть розрізнятись і ключі шифрування і розшифрування.

В загальному випадку криптосистема має наступну структуру (Рис.2.1):

Рисунок 1 - Структура криптосистеми

Робота криптосистеми може бути описана наступним чином:

З джерела ключів вибирається ключі (шифрування K_e і розшифрування K_d) і відправляються по надійним каналам передаючій і приймаючій стороні.

До вихідного (або відкритого) повідомлення p, що призначене для передачі, застосовується алгоритм шифрування E_{Ke}, внаслідок чого отримується зашифроване повідомлення $C=E_{Ke}(p)$.

Зашифроване повідомлення пересилається по каналу для обміну повідомленнями, який не вважається надійним (тобто зашифроване повідомлення може бути перехоплене порушником), приймаючій стороні.

На приймаючій стороні до зашифрованого повідомлення C застосовується обернене перетворення для отримання вихідного повідомлення $p=D_{Kd}(C)$.

В класичній криптографії для шифрування і розшифрування використовувався один і той самий ключ: $K_e=K_d=K$. Такі криптосистеми отримали назву криптосистем з секретним ключом (secret key cryptosystems); сьогодні їх також називають симетричним криптосистемами (symmetric cryptosystems).

Зауважимо, що алгоритми шифрування і розшифрування E і D відкриті, і секретність вихідного тексту p в даному шифротекста C залежить від таємності ключа K.

Для сучасної криптографії революційним стало усвідомлення того факту, що ключі шифрування і розшифрування можуть не співпадати. Такі криптосистеми отримали назву асиметричних криптосистем (asymmetric cryptosystems).

Оскільки зашифроване повідомлення передається через канал, доступний противнику, можливе його перехоплення і «прочитання» особою, яка не має ключа. В цьому випадку говорять про **дешифрування** повідомлення.

Таким чином, терміни «розшифрування» і «дешифрування» слід розрізняти: при розшифровуванні ключ вважається відомим, тоді як при дешифруванні ключ невідомий.

Розшифрування здійснюється так само просто, як і шифрування. Дешифрування є значно складнішою задачею.

Рівень складності цієї задачі і визначає здатність протистояти спробам противника заволодіти інформацією, яка захищається. В зв'язку з цим говорять про криптографічну стійкість шифру, розрізняючи більш і менш стійкі.

Наука, що вивчає методи дешифрування, називається **криптоаналізом**. Галузь знань, що об'єднує криптографію і крипто аналіз, називають **криптологією**.

Тема 2: Прості методи криптографічного перетворення. (4 години)

ПЛАН

1 Етапи розвитку криптографічних систем
2 Шифри перестановки. Прилад Сцитала. Магічні квадрати.
3 Шифри простої заміни
4 Шифри складної заміни

Ключові слова: *шифр Сцитала, криптостійкість, ключ шифру*

1 Етапи розвитку криптографічних систем

В ході розвитку криптосистем виділяють такі етапи (Таблиця 1).

Таблиця 1 -- **Етапи розвитку криптосистем**

Етап	Час	Особливості
I	V-IV тт. до н.е.	Виникнення перших шифрувальних пристроїв(Сцитала, табличка Енея і т.п.)
II	I ст. до н.е.	Початок застосування шифрованого зв'язку в системі органів влади (шифр Цезаря і т.п.)
III	XV-XVI ст. (Відродження)	Розвиток наукових методів криптографії і крипто аналізу (метод частотного аналізу, винайдення поліалфавітних шифрів, поворотної решітки)
IV	XVII-XVIII ст. (ера «чорних кабінетів»)	Використання номенклаторів як основного засобу шифрування
	XIX ст.	Винайдення гаміювання
	XX ст.	Винайдення колісних шифраторів (Енігма)
	кінець XX ст.	Народження «нової криптографії»

Перші системи шифрування з'явились одночасно з письменністю у V-IV тт. до н.е. Відомо, що вже в той час в Спарті використовувався один з перших шифрувальних пристроїв – **Сцитала**. Це був жезл циліндричної форми, на який намотувалась стрічка пергаменту. Вздовж осі циліндру записувався текст, призначений для передачі. Після запису стрічка знімалась з жезлу і передавалась адресату, який мав таку саму Сциталу.

Цікаво, що був відомий і метод дешифрації такого шифру, винайдення якого приписується Арістотелю. Пропонувалось заточити на конус довгий брус і, обернувши його стрічкою, починати зсувати її по конусу від малого діаметру до найбільшого. Там, де діаметр конусу співпадав з діаметром Сцитали, букви тексту утворювали слова. Далі залишалось тільки виготовити циліндр потрібного діаметру.

Іншим шифрувальним засобом часів Спарти була **табличка Енея**. На невеличкій горизонтальній табличці розташовувався алфавіт, а по її боковим сторонам розташовувались виїмки для намотування нитки. При шифруванні нитка закріплювалась з однієї із сторін

таблички і намотувалась на неї. На нитці робились відмітки (наприклад, вузлики) в місцях, що знаходились напроти букв даного тексту. Після шифрування нитка змотувалась і доставлялась адресату. Цей шифр був досить надійним: історії не відомі факти його дешифрування.

Подібні шифрові пристосування з невеличкими змінами проіснували до епохи розквіту Риму. Для управління імперією шифрований зв'язок в системі органів влади був життєво необхідним. Особливу роль відіграв шифр, запропонований Юлієм Цезарем: кожна буква повідомлення замінялась буквою алфавіту, зсунутою на три позиції (замість A – D, B – E, … , Z –A).

За часів Цезаря до XIV-XV століть криптографія розвивалась, але практика шифрування зберігалась у глибокій таємниці. Так, в часи хрестових походів шифрувальники Папи Римського після року роботи підлягали фізичному знищенню. Тому про цей період майже нічого невідомо.

В епоху Відродження почали розвиватись наукові методи криптографії і крипто аналізу. Зокрема для дешифрування шифрів став використовуватись

метод частотного аналізу зустрічає мості букв, для шифрування почали використовуватись **шифри багатозначної заміни** (омофони), отримали поширення **багатоалфавітні шифри** (шифр Тритемія, шифр Віженера), в якості засобу шифрування почала використовуватись **поворотна решітка.**

Поворотна решітка представляла собою лист з твердого матеріалу з прорізями. Накладаючи таку решітку на лист паперу, можна записувати у вирізи таємне повідомлення. Після цього, знявши решітку, треба було заповнити вільні місця, що залишились, маскуючим текстом.

Подібним стеганографічним методом маскування повідомлень користувались багато історичних осіб, зокрема, кардинал Рішіл'є у Франції. Він використовував в якості решітки прямокутник розміру 7x10 (Рис.2.2) з прорізами в позиціях (1,8), (2,9), (3,6), (4,5), (4,6), (5,1), (5,6), (5,7), (5,9), (6,2), (6,10), (7,9), (7,10).

Для довгих повідомлень прямокутник застосовувався кілька разів.

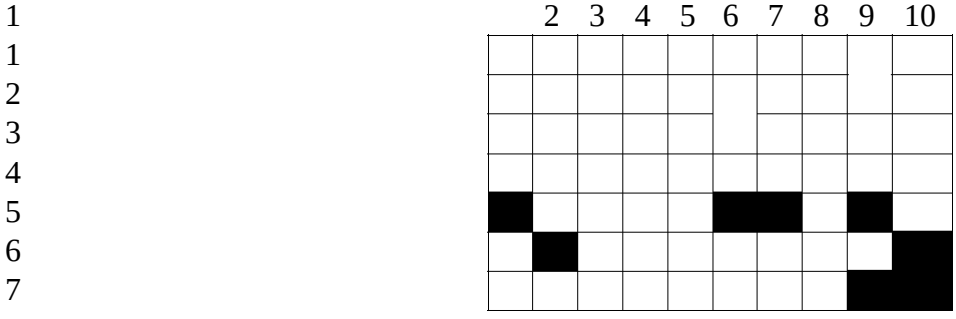


Рисунок 1 - Решітка Рішіл'є

В історії криптографії XVII-XVIII ст. називають ерою «чорних кабінетів». В цей період в багатьох державах Європи з'явились дешифрувальні підрозділи, які і називались «чорними кабінетами». Перший з них був створений у Франції за ініціативою кардинала Рішіл'є.

Характерним для цього періоду є широке розповсюдження шифрів, які називались **номенклаторами**. Номерклатори поєднували в собі просту заміну код. В якості прикладу номенклатура наведемо «малий шифр», що використовувався в наполеонівській армії (Таблиця 2).

Таблиця 2 - **«Малий шифр», що використовувався в наполеонівській армії**

A-15 AR-25 AL-39	J-87 JAI-123	R-169 RA-146 RE-126 RI-148
	K-?	S-167 SA-171 SE-177 SI-134
B-37 BU-3 BO-35 BI-29		
C-6 CA-32 CE-20	L-96 LU-103 LE-117	SO-168 SU-174
D-23 DE-52	LA-106	T-176 TI-145 TO-157
E-53 ES-82 ET-50 EN-68	M-114 MA-107	U-138
F-55 FA-69 FE-58 FO-71	N-115 NE-94 NI-116	V-164 VE-132 VI-161 VO-175
G-81 GA-51	O-90 OT-153	W, X, Y-?
H-85 HI-77	P-137 PO-152	Z-166
I-119 IS-122	Q-173 QUE-136	

Приклад застосування цього шифру наведений в Таблиці 2

Таблиця 3 - **Приклад шифрування «малим шифром» слова NAPOLEON**

N	A	P	O	L	E	O	N
115	15	137	90	96	53	90	115
або							
N	A	PO	LE	O	N		
115	15	152	117	90	115		

В простих нуменклаторах код складався з кількох десятків слів або фраз з двобуквеними кодовими позначеннями. З часом списки слів в номерклаторах збільшились до двох-трьох тисяч.

В цілому XVII-XVIII ст. не дали для криптографії нових ідей. Ера «чорних кабінетів» закінчилась в 40их роках XIX ст.

Нові ідеї з'явились в середині XIX ст. Цей період пов'язаний з виникненням стійкого способу ускладнення числових кодів – **гаміювання**. Він полягав у перешифровуванні закодованого повідомлення за допомогою деякого ключового числа, яке називалось **гамою.** Шифрування за допомогою гами полягало у сумуванні всіх кодованих груп повідомлення з одним і тим самим ключовим числом.

Приклад 1. Результат накладення гами 6413 на кодований текст (одиниці перенесення, що з'являються при додаванні між кодовими групами, опускаються):

3425 7102 8139

6413 6413 6413

9838 3515 4552

Приклад 2. Результат оберненої операції «зняття гами» 6413:

9838 3515 4552

6413 6413 6413

3425 7102 8139

Перша світова війна стала поворотним пунктом в розвитку криптографії в зв'язку зі зростанням обсягів шифропереписки. Криптографія стала широким полем діяльності. Проте нових наукових ідей в цей час не з'явилося.

Майже половина XX ст. була пов'язана з використанням **колісних шифраторів.** Їх різні конструкції були запатентовані майже одночасно в різних країнах (Голандії, Німеччині, Швеції). Найбільш відомою шифромашиною цих часів була «Енігма», яка у довоєнний період і під час другої світової війни використовувалась в германській армії. Принцип її роботи такий. На екрані оператору показувалася буква, якою шифрувалася відповідна літера на клавіатурі. Те, якою буде зашифрована літера, залежало від початкової конфігурації коліс. Суть в тому, що існувало понад сто трильйонів можливих комбінацій коліс, і з часом набору тексту колеса зсувалися самі, так що шифр змінювався протягом усього повідомлення. Все Енігми були ідентичними, так що при однаковому початковому положенні коліс на двох різних машинах і текст виходив однаковий. У німецького командування були Енігми і список положень коліс на кожен день, так що вони могли з легкістю розшифровувати

повідомлення один одного, але вороги не повідомляючи положень послання прочитати не могли.

В другій половині XX ст. в зв'язку з розвитком елементної бази обчислювальної техніки з'явились **електронні шифратори**. Сьогодні для шифрування даних найбільш широко застосовують три види шифраторів:

апаратні, програмно-апаратні та програмні. Їх основна відмінність полягає не тільки в способі реалізації шифрування і ступеня надійності захисту даних, але в ціні. Найдешевші пристрої шифрування - програмні, потім йдуть програмно-апаратні засоби і, нарешті, найдорожчі - апаратні.

В 70-их роках народилась «нова криптографія» - **криптографія з відкритим ключем.** Криптографія почала широко використовуватись не тільки у військовій, дипломатичній, державній сферах, а також в комерційній,

банківській та інших сферах.

Шифри перестановки

Простий метод криптографічного перетворювання, який містить правило переставляння літер у відкритому тексті. Шифри перестановки мають невелику криптостійкість, тому їх не використовують без додаткових перетворювань. Шифр перестановки здійснює перетворювання переставляння літер у відкритому тексті. Типовим прикладом шифру перестановки є **шифр Сцитала.**

Зазвичай відкритий текст розбивається на відрізки однакової довжини і кожен відрізок шифрується незалежно. Нехай, приміром, довжина відрізків дорівнює n та – взаємнооднозначне відбиття множини {1, 2, ..., n} в собі. Тоді шифр перестановки впроваджується у такий спосіб: відрізок відкритого тексту x x_n ... 1 перетворюється на відрізок шифрованого тексту)1( ) (... xδ xδ n .

Оберемо ціле додатне число, скажімо, 5; розташуємо числа від 1 до 5 у дворядковий запис, в якому другий рядок – довільне переставляння чисел верхнього рядка:

Цю конструкцію називають перестановкою, а число 5 – степенем перестановки. Зашифруємо фразу

«ДО БУЛАВИ ТРЕБА ГОЛОВИ».

У цій фразі 19 літер. Доповнимо її довільною літерою (наприклад Ь) до найближчого числа, кратного до 5, тобто 20.

Випишемо цю доповнену фразу без пропусків, водночас розбивши її на п'ятизнакові

групи:

ДОБУЛ АВИТР ЕБАГО ЛОВИЬ

Літери кожної групи переставимо відповідно до зазначеного дворядкового запису за таким правилом: перша літера ставиться на третє місце, друга – на друге, третя – на п'яте, четверта – на перше і п'ята – на четверте. Здобутий текст виписуємо без пропусків:

УОДЛБТВАРИГБЕОАИОЛЬВ

При розшифровуванні текст розбивається на групи по п'ять літер, які переставляються у зворотному порядку: перша – на четверте місце, друга – на друге, третя – на перше, четверта – на п'яте і п'ята – на третє.

Ключем шифру є обране число 5 і порядок розташування.

Прилад Сцитала

Одним з перших фізичних приладів, які зреалізовують шифр перестановки, є так званий прилад Сцитала.

Його було винайдено у давній, "варварській", Спарті за часів Лікурга (V ст. до н. е.). Рим швидко скористався цим приладом. Для зашифровування тексту використовувався циліндр заздалегідь обумовленого діаметра. На циліндр намотувався тонкий ремінець з пергамента, і текст виписувався порядково вздовж осі циліндра. Потім ремінець змотувався й доправлявся одержувачеві повідомлення. Останній намотував його на циліндр того самого ж діаметра і зчитував текст по осі циліндра. У цьому прикладі ключем шифру є діаметр циліндра та його довжина, котрі, власне кажучи, породжують дворядковий запис, аналогічний до того, що його наведено вище.

Шифр Сцитала зреалізовує один з варіантів сучасного так званого шифру маршрутної перестановки. Зміст цього шифру полягає в такому. Відкритий текст виписується в прямокутну таблицю з n рядків та m стовпців. Припускається, що довжина тексту t_{nm} (у противному разі ділянка тексту, що залишилася, шифрується окремо за тим самим шифром). Якщо t є строго менше за nm, то порожні клітинки, що залишилися, заповнюються довільним набором літер абетки. Шифртекст виписується за цією таблицею заздалегідь обумовленим "маршрутом" – шляхом, що він проходить одноразово через усі клітинки таблиці. Ключем шифру є числа n та m і окреслений маршрут. У такому трактуванні шифр Сцитала набуває описаного нижче вигляду.

Нехай m – кількість обвитків ремінця на циліндрі; n – кількість літер, розташованих на одному обвитку. Тоді відкритий текст, виписаний порядково в зазначену таблицю, шифрується шляхом послідовного зчитування літер за стовпцями. Оскільки маршрут є відомий і незмінний, то ключем шифру є числа m та mn, зумовлені діаметром циліндра й довжиною ремінця. При перехопленні повідомлення (ремінця) єдиним секретним ключем є діаметр.

Винахід дешифрувального пристрою – Антисцитала – приписують великому Аристотелю. Він запропонував використовувати конусоподібний "спис", на який намотувався перехоплюваний ремінець; цей ремінець пересувався віссю доти, аж допоки не з'являвся осмислений текст. В часи середньовіччя європейська криптографія набула сумнівного розголосу, який відлунює й дотепер.

Криптографію стали ототожнювати з чорною магією, з певною формою окультизму, астрологією, алхімією, єврейською каббалою. До зашифровування інформації 21 долучалися містичні сили. Приміром, рекомендувалося використовувати так звані "магічні квадрати".

"Магічні квадрати"

"Магічними квадратами" називають квадратні таблиці з вписаними в їхні клітинки послідовними натуральними числами, розпочинаючи від 1, що вони дають у сумі по кожному стовпцю, кожному рядку і кожній діагоналі одне й те саме число. Кількість "магічних квадратів" швидко зростає зі збільшенням розміру квадрата.

Кількість "магічних квадратів" 44 становить 880, а кількість "магічних квадратів" 55 – близько 250 000.

Уперше ці квадрати виникли в Китаї, де їм було надавано певної "магічної" сили.

Наведемо приклад: у квадрат розміром 44 вписуються цифри від 1 до 16. Зашифровування за "магічним квадратом" здійснюється у такий спосіб.

Приміром, треба зашифрувати фразу: «ПРИЛІТАЮ СЬОГОДНІ».

Літери цієї фрази вписуються послідовно до квадрата відповідно до записаних в них чисел, а в порожні клітинки (якщо такі є) проставляють довільні літери

І	**И**	**Р**	**О**
І	10 Ь	1 1 О	8 Ю
С	**Т**	**А**	**Г**
Л	**Н**	**Д**	**П**

Після цього зашифрований текст записується вже в рядок: ИРОІЬОЮСТАГЛНДП

При розшифровуванні текст вписується до квадрата – й відкритий текст читається в

послідовності чисел "магічного квадрата".

Даний шифр – звичайний шифр перестановки, але вважалося, що особливої стійкості йому надає чаклунство "магічного квадрата".

II Шифри простої заміни

Приклад

Розглянемо приклад використання системи шифрування Цезаря стосовно українського алфавіту при k = 3 і m = 36, внаслідок чого отримаємо такі одноалфавітні підставляння:

а	б	в	г	д	е
є	ж	з	и	і	ї
й	к	л	м	н	о
п	р	с	т	у	ф
х	ц	ч	ш	щ	ю
я	ь	_	.	,	'

⇒

.	,	'	а	б	в
г	д	е	є	ж	з
и	і	ї	й	к	л
м	н	о	п	р	с
т	у	ф	х	ц	ч
ш	щ	ю	я	ь	_

Спробуємо зашифрувати вхідний текст "Все йде, все минає, і краю немає." Результати його шифрування матимуть такий вигляд:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
В	с	е	_	й	д	е	,	_	в	с	е	_	м	и	н	а	є	,	_	і	_	к	р	а	ю	_	н	е	м	а	є	.
'	о	в	ю	и	б	в	ь	ю	'	о	в	ю	й	є	к	.	є	ь	ю	ж	ю	і	н	.	ч	ю	к	в	й	.	є	я

тобто, отримаємо на виході такий зашифрований текст:

'овюибвью'овюйєк.єьюжюін.чюквй.єя

Однак, при k = 5 отримаємо зовсім інший зашифрований текст.

Перевагою системи шифрування Цезаря є простота шифрування вхідного повідомлення та дешифрування зашифрованого тексту.

Приклад

Шифрування одноабетковою підстановкою

Нехай зашифровується повідомлення українською мовою й при цьому заміні підлягає кожна літера повідомлення. Формально в цьому разі шифр заміни можна описати в такий спосіб. Для кожної літери *а* вихідної абетки будується певна множина символів M_a, аби множини M_a та $M_в$ попарно не перетинались за $а \neq в$, тобто будь-які дві різні множини не містили однакових елементів. Множина M_a називається множиною шифропозначань для літери *а*.

Таблиця 1 є ключем шифру заміни. Знаючи її, можна здійснювати як зашифровування, так і розшифровування.

Таблиця 1 - Зашифровування літер відкритого повідомлення за допомогою будь-якого символу з множини M_j

а	*б*	*в*	…	*я*
M_a	$M_б$	$M_в$	…	$M_я$

При зашифровуванні кожна літера *а* відкритого повідомлення, розпочинаючи з першої, замінюється на будь-який символ з множини M_a . Якщо в повідомленні міститься кілька літер *а*, то кожна з них замінюється на будь-який символ з M_a. За рахунок цього за допомогою одного ключа (табл. 5.2) можна дістати різноманітні варианти зашифрованого повідомлення для одного й того самого відкритого повідомлення.

Наприклад, якщо ключем є таблиця 2, то повідомлення «мені відомі шифри заміни» може бути зашифровано, кожним з трьох способів, поданих в цій таблиці.

Таблиця 2 - Зашифровування літер відкритого повідомлення за допомогою одного з трьох числових символів множини M_j

а	*б*	*в*	*г*	*д*	*е*	*є*	*ж*	*з*	*и*	*і*	*к*	*л*	*м*	*н*	*о*	*п*
21	37	14	22	01	24	74	62	73	46	65	23	12	08	27	53	35
40	26	63	47	31	83	17	88	30	02	34	91	72	32	77	68	60
10	03	71	82	15	70	42	11	55	90	92	69	38	61	54	09	84

р	*с*	*т*	*у*	*ф*	*х*	*ц*	*ч*	*ш*	*щ*	*ь*	*ї*	*ю*	*я*
04	20	13	59	25	75	43	19	29	06	48	36	28	16
44	52	39	07	49	33	85	58	80	50	56	78	64	41
45	89	67	93	76	18	51	87	66	81	79	86	05	57

м	**е**	**н**	**і**	**в**	**і**	**д**	**о**	**м**	**і**	**ш**	**и**	**ф**	**р**	**и**	**з**	**а**	**м**	**і**	**н**	**и**
08	24	54	65	14	34	31	53	32	92	09	61	89	29	90	30	40	08	65	27	46
32	83	77	34	71	65	01	68	61	92	68	08	20	66	90	73	40	61	34	77	02
61	70	27	34	63	92	15	09	08	65	68	32	20	80	02	55	10	32	92	54	90

Оскільки множини M_a, $M_б$, $M_в$, …, $M_я$ попарно не перетинаються, то за кожним символом шифрованого повідомлення можна однозначно визначити, якій множині він належить й, отже, яку саме літеру відкритого повідомлення він замінює. Тому розшифровування є можливе й відкрите повідомлення визначається у єдиний спосіб.

Приклад:

Розглянемо 30-літерну абетку української мови:

АБВГДЕЄЖЗИІКЛМНОПРСТУФХЦЧШЩЬЮЯ

У цій абетці відсутні літери Ї, Й, що практично не обмежує можливостей за складання відкритих повідомлень українською мовою.

В абетці будь-якої природної мови літери слідують одна за одною у певному порядку. Це дає можливість надати кожній літері абетки її природний порядковий номер.

Занумеруємо літери української абетки відповідно до табл. 3.

Таблиця 3 - Відповідність поміж українською абеткою та множиною цілих

А	Б	В	Г	Д	Е	Є	Ж	З	И	І	К	Л	М	Н
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ю	Я
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30

Якщо у відкритому повідомленні кожну літеру замінити на її природний порядковий номер у розглядуваній абетці, то перетворення числового повідомлення на літерне дозволяє однозначно відновити вихідне відкрите повідомлення.

Наприклад, числове повідомлення 1 13 22 1 3 11 20 перетвориться на літерне повідомлення: АЛФАВІТ.

Аналогічно даному прикладу виконайте своє завдання

Якщо процес шифрування продовжувати необмежено, то в якому циклі вперше з'явиться вихідне слово?

Афінна система підставлянь Цезаря

У системі шифрування Цезаря використовувалися тільки адитивні властивості множини цілих чисел Zm . Проте символи множини Zm можна також помножити за модулем m. Застосовуючи одночасно операції додавання та множення за модулем m над елементами множини цілих чисел Zm , можна отримати систему підставляння, яку називають афінною системою підставляння Цезаря.

Перетворення в такій системі мають таке визначення:

$$E_{a,\,b}\,(t) = at + b \pmod{m}$$

У наведеному вище перетворенні

де a, b – цілі числа, $0 \le a, b \le m$, НСД(a, m) = 1.

Буква початкового тексту, яка відповідає числу t, замінюється на букву, що відповідає числовому значенню (a·t + b) за модулем m. Необхідно відзначити, що перетворення Ea,b(t) є взаємно однозначним відображенням на множині цілих чисел Zm ′ тільки в тому випадку, якщо найбільший спільний дільник чисел a і m, що позначається як НСД(a, m), дорівнює одиниці, тобто a і m повинні бути взаємно простими числами. Наприклад, якщо у англійському алфавіті m = 26 та прийнявши a = 3 і b = 5, очевидно НСД(3, 26) = 1, то отримуємо таку відповідність між числовими кодами букв:

t	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
$p=a\cdot t+b$	5	8	11	14	17	20	23	26	29	32	35	38	41	44	47	50	53	56	59	62	65	68	71	74	77	80
$p \bmod m$	5	8	11	14	17	20	23	0	3	6	9	12	15	18	21	24	1	4	7	10	13	16	19	22	25	2

Перетворюючи числа в букви англійського алфавіту, отримуємо таку відповідність для букв початкового тексту і зашифрованого тексту:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
a	b	c	d	e	f	q	h	i	j	k	l	м	n	o	p	q	r	s	t	u	v	w	x	y	z
5	8	11	14	17	20	23	0	3	6	9	12	15	18	21	24	1	4	7	10	13	16	19	22	25	2
f	i	l	o	r	u	x	a	d	q	j	м	p	s	v	y	b	e	h	k	n	q	т	w	z	c

Наприклад, початкове повідомлення encryption перетвориться в такий зашифрований текст: rslezykdvs.

Приклад

Розглянемо приклад використання афінної системи підставлянь Цезаря стосовно українського алфавіту. Тут, при m = 36, a = 7 і b = 13 НСД (7, 36) = 1, тобто отримуємо таку відповідність між числовими кодами символів:

0	1	2	3	4	5	
6	7	8	9	10	11	
12	13	14	15	16	17	
18	19	20	21	22	23	
24	25	26	27	28	29	
30	31	32	33	34	35	630

⇒

13	20	27	34	5	12	
19	26	33	4	11	18	
25	32	3	10	17	24	
31	2	9	16	23	30	
1	8	15	22	29	0	
7	14	21	28	35	6	630

Число 630 є контрольною сумою чисел, які знаходяться у кожній із таблиць. Перетворюючи числа в букви українського алфавіту, отримуємо таку відповідність для букв початкового тексту і зашифрованого тексту:

а	б	в	г	д	е
є	ж	з	и	і	ї
й	к	л	м	н	о
п	р	с	т	у	ф
х	ц	ч	ш	щ	ю
я	ь	_	.	,	'

⇒

к	с	ш	,	е	й
р	ч	.	д	ї	п
ц	_	г	і	о	х
ь	в	и	н	ф	я
б	з	м	у	ю	а
ж	л	т	щ	'	є

Спробуємо зашифрувати вхідний текст "Все йде, все минає, і краю немає." (див. прикл. 1). Результати його шифрування матимуть такий вигляд:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
В	с	е	_	й	д	е	,	_	в	с	е	_	м	и	н	а	є	,	_	і	_	к	р	а	ю	_	н	е	м	а	є	.
ш	и	й	т	ц	е	й	'	т	ш	и	й	т	і	д	о	к	р	'	т	ї	т	_	в	к	а	т	о	й	і	к	р	щ

тобто, отримаємо на виході такий зашифрований текст:

шийтцей'тшийтідокр'тїт_вкатойікрщ

Перевагою афінної системи підставлянь Цезаря над розглянутими раніше полягає у зручності управління ключами – ключі шифрування та дешифрування подаються в компактній формі у вигляді пари чисел (a, b). Недоліки афінної системи аналогічні недолікам системи шифрування Цезаря. Тому афінна система підставлянь Цезаря хоча і використовувалася на практиці декілька століть назад, проте сьогодні її застосовують переважно як ілюстративні приклади основних криптологічних положень різних систем шифрування.

Шифрування за допомогою таблиці Трисемуса

Для здобуття шифру заміни Трисемуса зазвичай використовувалися таблиці для записування літер абетки й ключове слово. У таблицю спочатку вписувалося по рядках ключове слово, причому повторювані літери відкидалися. Потім ця таблиця доповнювалася літерами абетки, які не увійшли до неї одна за одною.

Для української абетки таблиця Трисемуса може мати розмір 4×8.

Приклад.

Оберемо за ключ слово БАНДЕРОЛЬ.

Розглянемо шифрувальну таблицю з ключем, поданим у таблиці:

Шифрувальна таблиця з ключевим словом

Б	**А**	**Н**	**Д**	**Е**	**Р**	**О**	**Л**
Ь	В	Г	Є	Ж	З	И	І
Ї	Й	К	М	П	С	Т	У
Ф	Х	Ц	Ч	Ш	Щ	Ю	Я

Як і в разі полібіанського квадрата, при шифруванні відшукують у цій таблиці чергову літеру відкритого тексту і записують у шифртекст літеру, розташовану нижче за неї в тім самім стовпці. Якщо літера тексту міститься в нижньому рядку таблиці, тоді для шифртексту беруть першу верхню літеру з того ж самого стовпця.

Наприклад, при шифруванні за допомогою цієї таблиці повідомлення

ДІЛОМАЙСТРАВЕЛИЧАЄ

дістаємо шифртекст

ЄУІИЧВХЩЮЗВЙЖІТДВМ

Такі табличні шифри називаються ***монограмними***, тому що шифрування виконується за однією літерою.

Шифрування за допомогою біграмного шифру Плейфейра

Основою шифру Плейфейра є шифрувальна таблиця з випадково розташованими літерами абетки вихідних повідомлень.

У цілому структура таблиці системи шифрування Плейфейра є цілком аналогічна до структури таблиці Трисемуса. Тому для пояснення процедур зашифровування й розшифровування в системі Плейфейра скористаємося таблицею

Шифрувальна таблиця з ключевим словом

Б	**А**	**Н**	**Д**	**Е**	**Р**	**О**	**Л**
Ь	В	Г	Є	Ж	З	И	І
Ї	Й	К	М	П	С	Т	У
Ф	Х	Ц	Ч	Ш	Щ	Ю	Я

Процедура зашифровування включає такі вимоги:

Відкритий текст вихідного повідомлення розбивається на пари літер (біграми). Текст повинен мати парну кількість літер і в ньому не повинно бути біграм, котрі містили б дві

однакові літери. Якщо цих вимог не додержано, то текст змодифіковується навіть через незначні орфографічні помилки.

Послідовність біграм відкритого тексту перетворюється за допомогою шифрувальної таблиці на послідовність біграм шифртексту за такими правилами:

якщо дві літери відкритого тексту не потрапляють в один рядок чи стовпець (як, наприклад, літери А та И у таблиці- Шифрувальна таблиця з ключовим словом), тоді відшукують літери в кутах прямокутника, визначуваного даною парою літер. (У нашому прикладі це літери АИОВ. Пара літер АИ відбивається в пару ОВ.) Послідовність літер у біграмі шифртексту має бути дзеркально розташована стосовно послідовності літер у біграмі відкритого тексту;

- якщо обидві літери біграми відкритого тексту належать до одного стовпця таблиці, то за літери шифртексту вважаються літери, котрі розміщено під ними (наприклад, біграма НК дає біграму шифртексту ГЦ). Якщо при цьому літера відкритого тексту перебуває в нижньому рядку, то для шифртексту береться відповідна літера з верхнього рядка того самого стовпця (наприклад, біграма ВХ дає біграму шифртексту ЙА);

- якщо обидві літери біграми відкритого тексту належать до одного рядка таблиці, то за літери шифртексту вважаються літери, котрі розміщено праворуч від них (наприклад, біграма НО дає біграму шифртексту ДЛ). Якщо при цьому літера відкритого тексту перебуває в крайньому правому стовпці, то для шифру беруть відповідну літеру з лівого стовпця в тому самому рядку (наприклад, біграма ХЯ дає біграму шифртексту ЦФ).

Зашифруємо текст:

ВСЯКИЙ СВОГО ЩАСТЯ КОВАЛЬ

Розбиття цього тексту на біграми дає

ВС ЯК ИЙ СВ ОГ ОЩ АС ТЯ КО ВА Л Ь

Дана послідовність біграм відкритого тексту перетворюється за допомогою шифрувальної таблиці (Шифрувальна таблиця з ключовим словом) на таку послідовність біграм шифртексту:

ЗЙ ЦУ ВТ ЙЗ НИ РЮ РЙ УЮ ТН ЙВ БІ

При розшифровуванні застосовується зворотний порядок дій.

3 ШИФРИ СКЛАДНОЇ ЗАМІНИ

Шифрування за допомогою таблиці Віженера

Приклад

Нехай за ключове обране слово *АМБРОЗІЯ*. Необхідно зашифрувати повідомлення *ПРИЛІТАЮ СЬОМОГО*.

Випишемо відкрите повідомлення в рядок і запишемо під ним ключове слово з повторенням. В третій рядок будемо виписувати літери шифртексту, визначувані з таблиці Віженера

Повідомлення	П	Р	И	Л	І	Т	А	Ю	С	Ь	О	М	О	Г	О
Ключ	А	М	Б	Р	О	З	І	Я	А	М	Б	Р	О	З	І

Шифртекст	П	В	І	Б	Ш	Ь	І	Ь	С	Й	П	В	В	Ї	Ш

Шифрування за допомогою подвійного квадрата

1854 року англієць Чарльз Уїтстон розробив новий метод шифрування біграмами, котрий називають подвійним квадратом. Своєї назви цей шифр дістав за аналогією з полібіанським квадратом. Шифр Уїтстона відкрив новий етап в історії розвинення криптографії. На відміну від полібіанського, шифр "подвійний квадрат" використовує одразу дві таблиці (рис.1), розміщувані на одній горизонталі, а шифрування здійснюється біграмами, як у шифрі Плейфейра. Ці не надто складні модифікації призвели до з′являння на світ якісно нової криптографічної системи ручного шифрування. Шифр "подвійний квадрат" виявився вельми надійним та зручним і застосовувався в Німеччині навіть у роки другої світової війни.

Пояснімо процедуру шифрування цим шифром на прикладі. Нехай є дві таблиці з довільно розташованими в них українськими абетками. Перед зашифровуванням повідомлення розбивають на біграми. Кожна біграма шифрується окремо. Першу літеру біграми відшукують у лівій таблиці, а другу літеру – у правій таблиці. Потім подумки будують прямокутник у такий спосіб, аби літери біграми містилися в його протилежних вершинах. Інші дві вершини цього прямокутника надають літери біграми шифртексту.

Ж	**Щ**	**Н**	**Ю**	**Р**
И	**Т**	**Ь**	**Ц**	**Б**
Я	**М**	**Е**	**Л**	**С**
В	**І**	**П**	**Ч**	**А**
Х	**Д**	**У**	**О**	**К**
З	**Є**	**Ф**	**Г**	**Ш**

И	**Ч**	**Г**	**Я**	**Т**
Ф	**Ж**	**Ь**	**М**	**О**
З	**Ю**	**Р**	**В**	**Щ**
Ц	**У**	**П**	**Е**	**Л**
Є	**А**	**Н**	**Б**	**Х**
І	**К**	**С**	**Ш**	**Д**

Рисунок 1 – Дві таблиці з довільно розташованими символами української абетки для шифру "подвійний квадрат"

Припустімо, що шифрується біграма вихідного тексту ИЛ. Літера И міститься в стовпці 1 і рядку 2 лівої таблиці. Літера Л міститься в стовпці 5 і рядку 4 правої таблиці. Це означає, що прямокутник утворено рядками 2 й 4, а також стовпцями 1 лівої таблиці й 5 правої таблиці. Отже, до біграми шифртексту входять літера О, розміщена в стовпці 5 і рядку 2 правої таблиці, і літера В, розташована в стовпці 1 і рядку 4 лівої таблиці, тобто здобуваємо біграму шифртексту ОВ.

Якщо обидві літери біграми повідомлення містяться в одному рядку, то й літери шифртексту беруть з того самого рядка. Першу літеру біграми шифртексту беруть з лівої таблиці в стовпці, який відповідає другій літері біграми повідомлення. Друга ж літера біграми шифртексту береться з правої таблиці в стовпці, який відповідає першій літері біграми повідомлення. Тому біграма повідомлення ТО перетворюється на біграму шифртексту БЖ. В аналогічний спосіб шифруються всі біграми повідомлення:

Повідомлення		**ПО**		**ВТ**		**ОР**		**НА**		**ПЕ**		**РЕ**		**ДА**		**ЧА**
Шифртекст		**ЛЬ**		**ЛЖ**		**НЛ**		**ЧУ**		**ЧП**		**ЯА**		**ДА**		**УО**

Шифрування методом подвійного квадрата надає вельми стійкий до розкриття і простий у застосуванні шифр. Зламування шифртексту "подвійний квадрат" потребує потужних зусиль, при цьому довжина повідомлення має бути не менш ніж тридцять рядків.

Шифрування за допомогою шифру Гронсфельда

Шифр складної заміни, називаний *шифром Гронсфельда*, являє собою модифікування шифру Цезаря з числовим ключем. Для цього під літерами відкритого повідомлення записують цифри числового ключа. Якщо ключ є коротший за повідомлення, то його запис циклічно повторюють. Шифртекст здобувають приблизно так само, як в шифрі Цезаря, але відлічують за абеткою не третю літеру (як це чинять в шифрі Цезаря), а обирають ту літеру, котру зміщено за абеткою на відповідну цифру ключа. Шифр Гронсфельда являє собою, власне кажучи, окремий випадок системи шифрування Віженера (ключ надається в числовому вигляді), тому для шифрування відкритого повідомлення користуються таблицею Віженера , яка подана нижче

Приміром, застосовуючи як ключ групу з чотирьох початкових цифр числа *e* (основу натурального логарифму), а саме 2718, здобуваємо для відкритого повідомлення.

БЕЗ ОХОТИ НЕМА РОБОТИ

такий шифртекст:

Повідом-лення		**Б**	**Е**	**З**	**О**	**Х**	**О**	**Т**	**И**	**Н**	**Е**	**М**	**А**	**Р**	**О**	**Б**	**О**	**Т**	**И**
Ключ		**2**	**7**	**1**	**8**	**2**	**7**	**1**	**8**	**2**	**7**	**1**	**8**	**2**	**7**	**1**	**8**	**2**	**7**
Шифр-текст		**Г**	**Й**	**И**	**Ц**	**Ч**	**Х**	**У**	**О**	**П**	**Й**	**Н**	**З**	**Т**	**Х**	**В**	**Ц**	**Ф**	**Н**

Слід зазначити, що шифр Гронсфельда розкривається відносно легко, якщо врахувати, що в числовому ключі кожна цифра має лише десять значень, а отже, є лише десять варіантів прочитання кожної літери шифртексту. З іншого боку, шифр Гронсфельда припускає подальші модифікування, що вони поліпшують його стійкість, зокрема подвійне шифрування різними числовими ключами.

Т

(4 години)

м

а Практичну реалізацію шифру перестановки в системах зв'язку може бути подано на такому прикладі.

Практичне застосовування шифрів перестановки в системах зв'язку

Особливістю телефонного зв'язку є те, що акустичний сигнал у телефонному терміналі перетворюється на електричний і потім, після опрацьовування й посилення, передається лініями зв'язку. На приймальному кінці електричний сигнал знову перетворюється на акустичний; при цьому вихідна форма сигналу більш-менш зберігається. Акустичний і, відповідно, електричний сигнали характеризуються частотним спектром. Їх можна розглядати в розгорненні в часі й за спектром.

Відомо кілька типових перетворювань аналогового сигналу, котрі може бути легко реалізовано інженерними методами. Зазначимо головні з них.

Перестановка частот. За допомогою системи фільтрів уся ширина смуги стандартного телефонного каналу може бути поділена на певну кількість частотних смуг, котрі потім може бути переставлено поміж собою (рис. 1.1 та 1.2).

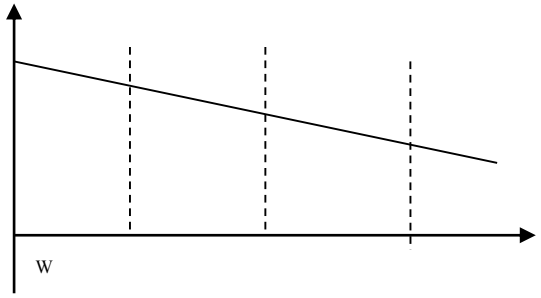


Рисунок 1- Поділення ширини смуги

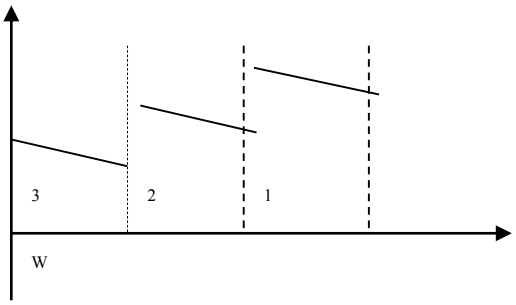


Рисунок 1.2- Перестановка частотних смуг

Найпростіший скремблер обмежує захист уведенням подібних найпростіших частотних перетворювань.

Серійний скремблер переставляє діапазони 250...675 Гц, 675...1100 Гц, 1100...1525 Гц та 1950...2375 Гц.

У даній схемі на вхід вузла накладання шифру подається одне й те саме керування, що воно зорганізовує переставляння.

Інвертування спектра. Більш складні скремблери додатково здійснюють інвертування спектра (рис. 1.3).

Рисунок 1.3 – Інвертування спектра

Частотно-часові перестановки. Ще більш складні системи розбивають сигнал на часовому інтервалі 60...500 мс і на кожнім інтервалі використовують у комбінації власні аналогові перетворювання. Змінюванням перетворювань у різні часові інтервали керує послідовність, яка надходить на вузол накладання шифру з блока ускладнювання. Той, хто просто послухає дешифроване аналоговим сигналом мовлення, почує якийсь булькіт, шум.

Про складність завдання зашифровування й розшифровування аналогових повідомлень писав Солженіцин у своєму «В круге первом»: «…Клиппирование, демпфирование, амплитудное сжатие, электронное дифференцирование и интегрирование привольной человеческой речи были таким же инженерным издевательством над ней, как если бы кто-нибудь взялся расчленить Новый Афон или Гурзуф на кубики вещества, втиснуть в миллиард спичечных коробков, перевезти самолетом в Нерчинск, на новом месте распутать, неотличимо собрать и воссоздать субтропики, шум прибоя, южный воздух и лунный свет. То же, в некоторых импульсах, надо было сделать и с речью, даже воссоздать ее так, чтобы не только было понятно, но Хозяин мог бы по голосу узнать, с кем говорит…».

Так само барвисто й дещо зневажливо Солженіцин відгукувався про власне роботу з розробляння вітчизняних засобів захисту телефонної інформації. Однак тут він був неправий. На думку висококваліфікованих фахівців, того часу робота йшла вельми цілеспрямовано й апаратуру насправді випускали в призначений термін. Це – апаратура часової стійкості. За

наявності спецтехніки типу видимого мовлення зміст перемов удається відновити. Якщо забути про інверсії, то ми маємо шифр перестановки. Для гарантованого засекречування телефонного мовлення його спочатку оцифровують – переводять у двійкову послідовність, а потім опрацьовують так само, як і будь-яке текстове повідомлення.

На практиці використовується апаратура як аналогового, так і цифрового засекречування.

Цифрова апаратура забезпечує гарантовану надійність захисту, але вона є більш вимоглива до каналів зв'язку.

Аналогова апаратура є менш стійка, але більш дешева, більш портативна, менш вимоглива до каналів зв'язку.

Тема 6: **Стандарт шифрування ГОСТ 28147-89**

Алгоритм криптографічного перетворення даних ГОСТ 28147-89 призначений для апаратної і програмної реалізації. Цей алгоритм задовольняє потребам, які пред′являються до сучасних криптосистем, не накладає обмежень на секретність інформації, яка передається.

Алгоритм побудований на основі мережі Фейстеля (рис.1). На кожному етапі блок вихідного повідомлення розбивається на ліву L_0 і праву R_0 частини, які шифрують за правилом

$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i),$$

При цьому використовують проміжне значення ключа K_i. Відмінність від алгоритму *DES* полягає у тому, що замість *56*–розрядного ключа використовують *256* – розрядний ключ. В алгоритмі *ГОСТ* використовують нелінійну функцію $f_{ГОСТ}$, яка суттєво відрізняється від функції f_{DES} що використовується в алгоритмі *DES*. Кількість етапів шифрування в два рази більше в порівняні з *DES*, тобто дорівнює 32.

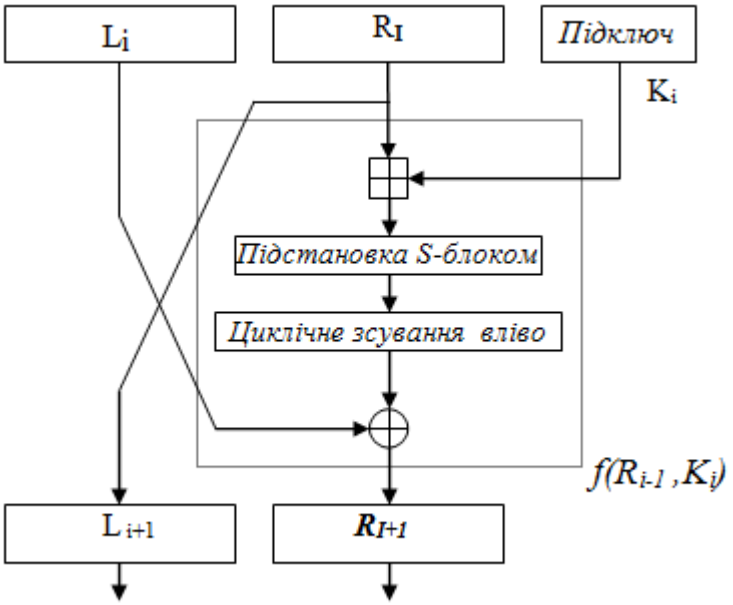


Рисунок 1 - Реалізація мережі Фейстеля в алгоритмі ГОСТ 28147-89

Крім цього, операція заміни, яка виконується *S-б*локами не постійна, а може змінюватися при необхідності і, до того ж, її треба держати в секреті, що еквівалентно збільшенню ключа, практично до *610* біт.

Алгоритм ГОСТ допускає декілька режимів роботи. До яких відноситься режим простої заміни, режим гаммування, режим гаммування із зворотнім зв′язком і режим формування імітовставки.

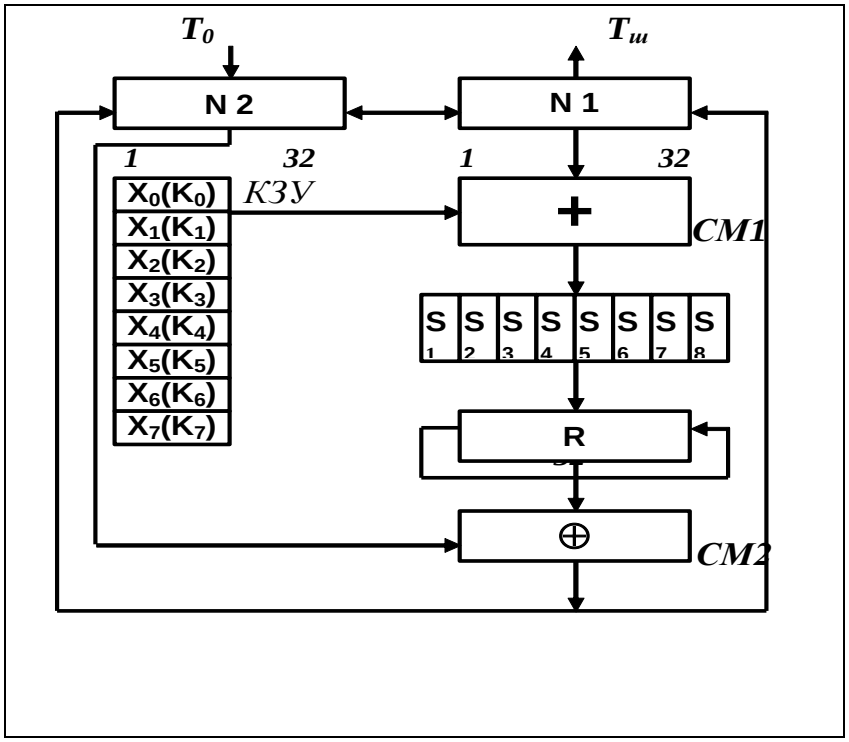
1 Режим простої заміни

Режим простої заміни передбачає розділення шифруючої послідовності на блоки довжиною *64* розряда і є складовою частиною інших режимів шифрування. Цей режим можна використовувати тільки для шифрування блоків, довжина яких кратна *64*–м бітам.

В зв′язку з цим його використовують тільки при передачі наступного ключа.

Процедура зашифровування в цьому режимі включає 32 цикли. В ключовий запам'ятовувальний пристрій вводять 256 біт ключа К в вигляді восьми 32- розрядних підключів. Зашифрування даних здійснюється за допомогою блока підстановки S, який складається з 8 вузлів заміни, та регістра зсування, який виконує циклічне зсування вліво (на 11 розрядів) 32 - розрядного вектору, отриманого з виходу блока S.

Схема алгоритму шифрування у режимі простої заміни зображена на рис. 2



КЗП - ключовий пристрій, що запам'ятовує на 256 біт;
СМ1 –32-розрядний суматор за модулем 2^{32}
СМ2 – 32-розрядний за модулем 2
N 1, N 2 – 32-розрядні накопичувачі
R - 32-розрядний регістр циклічного зсуву
S - 32-розрядний блок підстановки, який складається з восьми вузлів заміни $S_1 \div S_8$ з пам'яттю 64 біт кожен.

Рисунок 2 – Шифрування у режимі простої заміни

Відкрите повідомлення розбивається на 64-розрядні блоки T_0.

Процедура шифрування містить 32 циклу$(j=1,\ldots,32)$.

Перед початком шифрування в ключовий пристрій, що запам'ятовує, вводять 256-розрядний ключ, що розділяється на блоки завдовжки в 32 розряди кожен.

$$K = K_7\, K_6\, K_5\, K_4\, K_3\, K_2\, K_1\, K_0.$$

При цьому вхідна послідовність розбивається навпіл:

$$T_0 = a_1(0), a_2(0),\ldots, a_{32}(0), b_1(0), b_2(0),\ldots, b_{32}(0).$$

Біти $b(0)$, - старші, такі, що стоять ліворуч, а $a(0)$ - молодші біти, що стоять праворуч.

Послідовність

$a(0) = a_{32}(0), a_{31}(0),\ldots, a_0(0)$

вводять в накопичувач $N\,1$, послідовність

$b(0) = b_{32}(0), b_{31}(0),\ldots, b_0(0)$

вводять в накопичувач $N\,2$.

Першій цикл процедури шифрування можна описати таким чином

$$\begin{cases} a(1) = f\,(a(0) + K_0) \oplus b(0) \\ b(1) = a(0) \end{cases}$$

У даному виразі *a(1)* - заповнення *N1* після першого циклу зашифровування; *b(1)* - заповнення *N2* після першого циклу зашифровування; *f* - функція шифрування.

Аргументом функції f є сума по модулю 2^{32} числа *a(0)* і числа K_0. Ця функція включає дві операції над одержаною *32-х* розрядною сумою $(a(0) + K_0)$.

Перша операція є заміною і виконується блоком підстановки *S*.

Блок складається з восьми вузлів заміни $S_1 \div S_8$ з пам'яттю 64 біт кожен. 32-розрядний блок, що поступає від суматора, розбивають на вісім чотирьох- розрядних груп, кожна з яких замінюється іншою чотирьохрозрядною групою. Кожний вузол заміни можна уявити у вигляді таблиці-перестановки шістнадцяти чотирьохрозрядних двійкових чисел в межах 0000…1111. Вхідний вектор – це адреса рядка в таблиці, а число в цьому рядку є вектором на виході. Порядок заміни, що виробляється блоком *S*, тримається у секреті та відносно рідко змінюється.

Друга операція є циклічним зсуванням 32-х розрядного вектора ліворуч (на 11 розрядів). Ця операція виконується за допомогою регістра зсування *R*.

У наступному етапі здійснюється складання по модулю два результатів обчислення функції f і вмісту накопичувача *N 1*. Отриманий результат записують у накопичувач *N 1*, а у накопичувач *N 2* переписують попереднє значення вектора, що містився в *N 1*. На цьому закінчується перший цикл шифрування.

Останні цикли виконуються аналогічно. Відмінність полягає в тому, що, починаючи з 25-го циклу, ключі подаються з блоку КЗУ в зворотному порядку $K_7 \div K_0$.

Таким чином, протягом всіх 32-х циклів, подача ключів здійснюється в наступному порядку:

$$K_0, K_1, K_2, K_3, K_4, K_5, K_6, K_7, K_0, K_1, K_2, K_3, K_4, K_5, K_6, K_7,$$

$$K_0, K_1, K_2, K_3, K_4, K_5, K_6, K_7, K_7, K_6, K_5, K_4, K_3, K_2, K_1, K_0.$$

У 32-му циклі результат з суматора СМ2 вводиться в накопичувач *N 2*, а в накопичувачі *N 1* зберігається його попередній вміст. При цьому вміст обох накопичувачів *N 1* і *N 2* є блоком зашифрованих даних $T_ш$, відповідних блоку відкритих даних T_0.

У загальному вигляді рівняння, що описують процес шифрування можуть бути представлені в наступному вигляді:

$$\begin{cases} a(j) = f\,[a(j-1) + K_{j-1(\mathrm{mod}\ s)}] \oplus b(j-1) \\ b(j) = a(j-1) \end{cases} \quad \text{при } j = 1,\ldots, 24$$

$$\begin{cases} a(j) = f\,[a(j-1) + K_{32-j}] \oplus b(j-1) \\ b(j) = a(j-1) \end{cases} \quad \text{при } j = 25,\ldots, 31$$

$$\begin{cases} a(32) = a(31) \\ b(j) = f\,[a(31) + K_0] \oplus b(31) \end{cases} \quad \text{при } j = 32$$

де $a(j) = a_{32}(j), a_{31}(j) ., a_1(j),$ - заповнення $N\ 1$ після j -го циклу шифрування, а $b(j) = b_{32}(j), b_{31}(j) ., b_{1(j)},$ - заповнення $N\ 2$ після j -го циклу шифрування;

$j = 1,2 ., 31.$

Блок зашифрованих даних виводиться з обох накопичувачів в наступному порядку.

$$T_ш = [a_1(32), a_2(32),\ldots, a_{32}(32), b_1(32), b_2(32),\ldots, b_{32}(32)].$$

Решта блоків відкритих даних зашифровуваються в режимі простої заміни аналогічно.

Процедура розшифрування має той же вигляд, що і при шифруванні. Прийнятий з каналу зв'язку блок $T_ш$, записують в накопичувачі $N\ 1$ і $N\ 2$ так, щоб початкове значення вмісту накопичувача $N\ 1$ мало вигляд

$$[a_{32}(32), a_{31}(32),\ldots, a_2(32), a_1(32)],$$

а вміст накопичувача $N\ 2$, мало вигляд

$$[b_{32}(32), b_{31}(32),\ldots, b_2(32), b_1(32)] .$$

Розшифровування здійснюється таким чином, що і шифрування, проте порядок подачі 32-х розрядних ключів здійснюється в наступному порядку:

$K_0, K_1, K_2, K_3, K_4, K_5, K_6, K_7, K_7, K_6, K_5, K_4, K_3, K_2, K_1, K_0$
$K_7, K_6, K_5, K_4, K_3, K_2, K_1, K_0, K_7, K_6, K_5, K_4, K_3, K_2, K_1, K_0$.

Рівняння розшифрування при цьому мають вигляд:

$$\begin{cases} a(32-j) = f\,[a(32-j+1) + K_{j-1}] \oplus b(32-j+1) \\ b(32-j) = a(32-j+1) \end{cases} \quad \text{при } j = 1,\ldots,8;$$

$$\begin{cases} a(32-j) = f\,[a(32-j+1) + K_{32-j(\mathrm{mod}\,8)}] \oplus b(32-j+1) \\ b(32-j) = a(32-j+1) \end{cases} \quad \text{при } j = 9,\ldots,31;$$

$$\begin{cases} a(0) = a(1) \\ b(0) = f\,[a(1) + K_0] \oplus b(1) \end{cases} \quad \text{при } j = 32.$$

Одержані після 32-х циклів роботи заповнення накопичувачів *N 1* і *N 2* утворюють блок відкритих даних.

$T_0 = [a_1(0), a_2(0),\ldots, a_{32}(0)\, b_1(0), b_2(0)\,.,\, b_{32}(0)]$.

Аналогічно розшифровується решта блоків зашифрованих даних.

Введемо функцію алгоритму шифрування методом простої заміни $A(T_0) = T_ш$.

Режим простої заміни допустимо застосовувати для шифрування даних тільки в обмежених випадках – при виробці ключа і зашифровуванням його з забезпеченням імітозахисту для передачі по каналах зв′язку чи для збереження в пам'яті комп'ютера.

2 Режим гаммування

При використанні шифрування в режимі гаммування, вихідна послідовність, як і у попередньому випадку, розбивається на блоки завдовжки 64 біта.

$T_0(1), T_0(2), T_0(3)\,., T_0(i)\,., T_0(m)$

де $T_0^{(i)}$- i –й 64-х розрядний блок відкритих даних; $i = 1,\ldots, m$.

Шифрування цих блоків здійснюється в режимі гаммування шляхом накладення на них шифруючої гамми.

$Г_ш = (Г_ш^{(1)}, Г_ш^{(2)}, \ldots, Г_ш^{(i)}, \ldots, Г_ш^{(m)})$

де $Гш^{(i)}$ – i -ий 64-х розрядний блок відкритих даних; $i = 1, \ldots, m$.

Число двійкових розрядів в блоці може бути менше 64. В цьому випадку зайва частина гамми відкидається.

Рівняння шифрування має вигляд

$T_ш^{(i)} = T_0^{(i)} \oplus Г_ш^{(i)}$

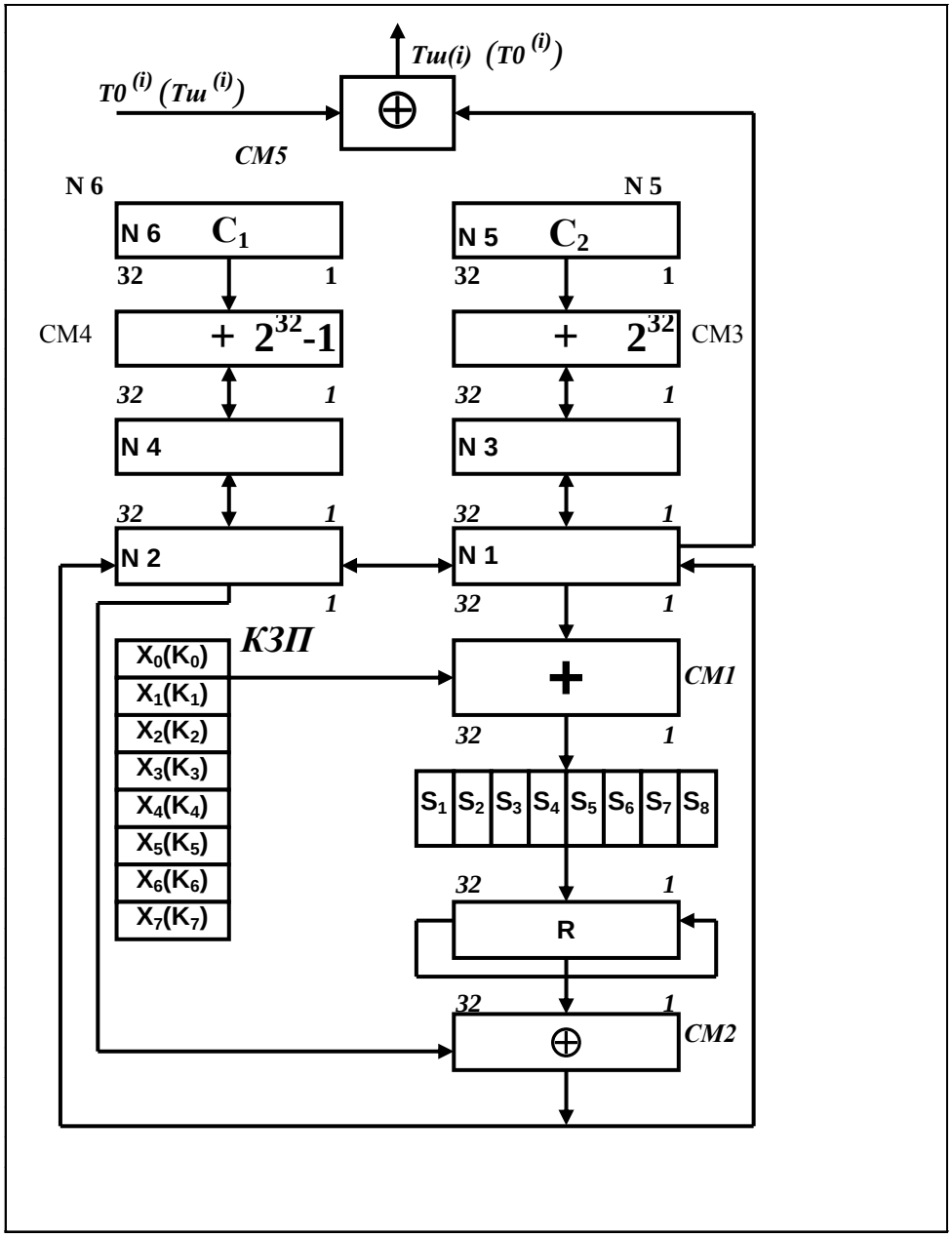
де $Г_ш^{(i)} = A(Y_{i-1} + C_2\,,\, Z_{i-1} + C_1),\ i = 1, \ldots, m;\ T_ш^{(i)}$ – i -й блок 64-х розрядного блоку зашифрованого тексту. $A(.)$ – шифрування в режимі простої заміни. $C1, C2$ – 32-х розрядні двійкові константи. Yi, Zi – 32-х розрядні двійкові послідовності.

Величини *Yi, Zi* визначаються ітераційно по мірі формування гамми шифру.

(Yi, Zi) = A(S)

де *S* – синхропосилка (64-розрядна двійкова послідовність).

(Yi, Zi) = A(Yi -1 + C2, Zi -1 + C1), i = 1 ... m.



КЗП - ключовий пристрій, що запам'ятовує на 256 біт;СМ1 –32-розрядний суматор за модулем 2^{32}

СМ2 – 32-розрядний за модулем 2

N 1, N 2 – 32-розрядні накопичувачі

R - 32-розрядний регістр циклічного зсуву

S - 32-розрядний блок підстановки, який складається з восьми вузлів заміни $S_1 \div S_8$ з пам'яттю 64 біт кожен.

Рисунок 2 - Схема реалізації режиму гаммування

Процедура гаммування здійснюється таким чином.

У накопичувачі *N 6* і *N 5* наперед записуються константи, що мають наступні значення в шістнадцятковій формі:

$C1 = 01010104_{(16)}$ $C2 = 01010101_{(16)}$

У ключовий запам'ятувальний пристрій КЗП вводиться 256 біт ключа; у накопичувачі *N 1* і *N 2* – 64-розрядна двійкова послідовність (синхропосилка)

$S = (S_1, S_2 ., S_i, ..., S_{64})$

Ця синхропосилка є початковим заповненням накопичувачів *N 1* і *N 2* для послідовного формування *m* блоків гамми шифру.

$N\,1 \rightarrow (S_{32}, S_{31} ., S_2, S_1)$ $N\,2 \rightarrow (S_{64}, S_{63} ., S_{32}, S_{31})$

Синхропосилка *S* шифрується в режимі простої заміни, а результат її шифрування

$A(S) = (Y_0, Z_0)$

переписується в 32-х розрядні накопичувачі *N 3* і *N 4*, таким чином:

$N\,1 \rightarrow N\,3$ $N\,2 \rightarrow N\,4$

Після цього, вміст накопичувачів *N 6* і *N 4* підсумовується по модулю $2^{32} - 1$ в СМ4, а вміст накопичувачів *N 5* і *N 3* підсумовується по модулю 2^{32} в СМ3. При цьому, в першому випадку результат підсумовування записується в накопичувач *N 4*, а в другому випадку – в накопичувач *N 3*.

На наступному кроці вміст накопичувача переписується в накопичувач *N 2*, а вміст накопичувача *N 3* переписується в *N 1*. При цьому вміст накопичувачів *N 4* і *N 3* зберігається, а вміст накопичувачів *N 1* і *N 2* використовується як шифруюча гамма

$\Gamma_ш^{(1)} = (\gamma_1^{(1)}, \gamma_2^{(1)}, \ldots, \gamma_{63}^{(1)}, \gamma_{64}^{(1)})$,

яка порозрядно підсумовується по модулю два в суматорі СМ5 з першим блоком послідовності

$T_0^{(1)} = (t_1^{(1)}, t_2^{(1)} ., t_{63}^{(1)}, t_{64}^{(1)})$.

В результаті, на виході суматора СМ5 одержують перший зашифрований блок.

$T_ш^{(1)} = T_0^{(1)} \oplus \Gamma_ш^{(1)} = (\tau_1^{(1)}, \tau_2^{(1)}, \ldots, \tau_{63}^{(1)}, \tau_{64}^{(1)})$.

де $\tau_i^{(1)} = t_i^{(1)} \oplus \gamma_i^{(1)}$; $i = 1, \ldots, 64$.

На наступному етапі шифрування, коли криптографічному перетворенню піддається блок $T_0^{(2)}$, зміст накопичувачів ще раз підсумовується з константами С1 і С2, що містяться в накопичувачах *N 5* і *N 6*. Коли результати підсумовування знов опиняться в накопичувачах *N 1*и *N 2*, вони наново шифруються в режимі простої заміни і, таким чином, формується нова гамма $\Gamma_ш^{(2)}$.

Формування подальших блоків гамми $\Gamma_ш^{(i)}$ здійснюється аналогічно.

Розшифровування в режимі гаммування здійснюється таким же чином як і шифрування. Рівняння розшифрування має вигляд

$$T_0^{(i)} = T_ш^{(i)} \oplus \Gamma_ш^{(i)} = T_ш^{(i)} \oplus A(Y_{i-1} + C_2, Z_{i-1} + C_1), \quad i = 1, \ldots, m.$$

Процес розшифровування можливий тільки за наявності синхропосилки, яка не є секретним числом і передається разом з повідомленням у відкритому вигляді.

3 Режим гаммування із зворотним зв′язком

Використання даного алгоритму, як і в попередніх випадках, припускає розділення послідовності, яка передється, на блоки, кратні 64-м символам.

$$T_0^{(1)}, T_0^{(2)}, T_0^{(3)}, \dots, T_0^{(i)}, \dots, T_0^{(m)}.$$

Шифрування, так само як і у попередньому випадку, здійснюється шляхом підсумовування шифрованого блоку з гаммою шифру

$$Г_ш = Г_ш^{(1)}, Г_ш^{(2)}, \dots, Г_ш^{(i)}, \dots, Г_ш^{(m)}.$$

При цьому рівняння шифрування має вигляд

$$T_ш^{(1)} = A(S) \oplus T_0^{(1)} = Г_ш^{(1)} \oplus T_0^{(1)}$$

$$T_ш^{(i)} = A(T_ш^{(i-1)}) \oplus T_0^{(i)} = Г_ш^{(i)} \oplus T_0^{(i)},\ i = 2, \dots, m.$$

де $T_ш^{(i)}$ – i -й блок, 64-х розрядний блок зашифрованого тексту; $A(S)$ - функція шифрування в режимі простої заміни; m - визначає об'єм відкритих даних.

Аргументом функції $A(S)$ на першому кроці ітеративного алгоритму є синхропосилка S, а на решті всіх етапів - попередній блок зашифрованих даних $T_ш^{(i-1)}$.

Процедура шифрування виконується таким чином. У *КЗП* вводиться 256 біт ключа; у накопичувачі *N 1* і *N 2* – 64-розрядна двійкова послідовність (синхропосилка)

$$S = (S_1, S_2, \dots, S_i, \dots, S_{64}).$$

Ця синхропосилка шифрується звичайним шляхом в режимі простої заміни. Таким чином, після 32-х циклів обробки в накопичувачах *N 1* і *N 2* буде сформовано перший 64-х бітовий блок шифруючої гамми $Г_ш(1) = A(S)$. Цей блок підсумовується по модулю два в суматорі СМ3 з блоком відкритих даних

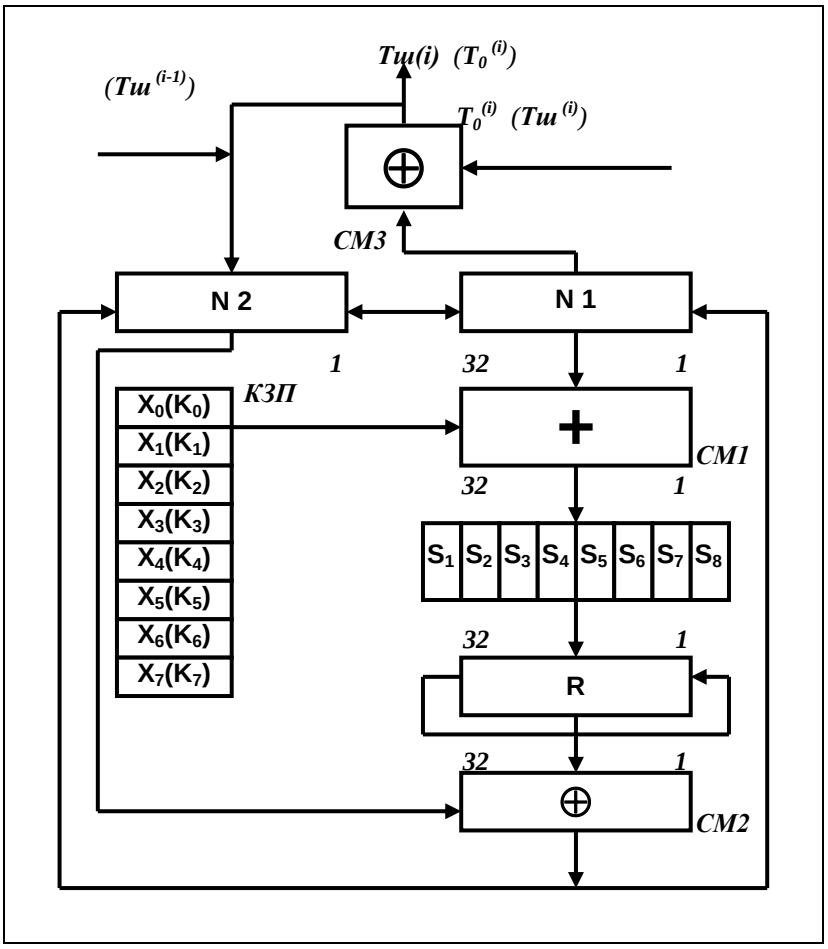
$$T_0^{(1)} = (t_1^{(1)}, t_2^{(1)}, \dots, t_{63}^{(1)}, t_{64}^{(1)}).$$

В результаті одержують перший блок зашифрованих даних

$$T_ш^{(1)} = T_0^{(1)} \oplus Г_ш^{(1)} = (\tau_1^{(1)}, \tau_2^{(1)}, \dots, \tau_{63}^{(1)}, \tau_{64}^{(1)}).$$

Одночасно з гаммуванням, по колу зворотнього зв'язку, блок записується в накопичувачі *N 1* і *N 2*, в подальшому, з нього формують чергову 64-х розрядну гамму $Г_ш^{(2)}$.

Всі подальші операції по шифруванню відкритих блоків повідомлення виконуються аналогічно.



КЗП - ключовий пристрій, що запам'ятовує на 256 біт;

СМ1 –32-розрядний суматор за модулем 2^{32}

СМ2 – 32-розрядний за модулем 2

N 1, N 2 – 32-розрядні накопичувачі

R - 32-розрядний регістр циклічного зсуву

S - 32-розрядний блок підстановки, який складається з восьми вузлів заміни $S_1 \div S_8$ з пам'яттю 64 біт кожен.

Рис 3 -Шифрування методом гаммування із зворотним зв'язком

У канал зв'язку, таким чином, передається синхропосилка і блоки зашифрованих даних

$$T_ш = T_ш^{(1)}, T_ш^{(2)}, \ldots, T_ш^{(i)}, \ldots, T_ш^{(m)}, i = 1, \ldots, m.$$

Розшифровування прийнятих з каналу зв'язку даних здійснюється в зворотньому порядку. Рівняння розшифрування має вигляд

$$T_0^{(1)} = A(S) \oplus T_ш^{(1)} = Г_ш^{(1)} \oplus T_ш^{(1)}$$

$$T_0^{(i)} = Г_ш^{(i)} \oplus T_0^{(i)} = A(T_ш^{(i-1)}) \oplus T_ш^{(i)}. i = 2, \ldots, m.$$

Процедура розшифровування здійснюється аналогічно шифруванню. Вона починається з формування першої гамми $Г_ш^{(1)}$ з синхропосилки S. "Дзеркальність" процедур шифрування і розшифрування пояснюється особливістю виконання операції підсумовування по модулю два.

4 Режим формування імітовставки

Імітовставка - це блок з P біт, який виробляють за певним правилом з

відкритих даних з використанням ключа, а потім додають до зашифрованих даних для забезпечення імітозахисту.

Імітозахист - це захист системи шифрування від нав'язування помилкових даних.

Імітовставка I_p виробляється з блоків відкритих даних або перед передачею повідомлення, або одночасно з його шифруванням.

Значення параметра *P (число розрядів в імітовставці)* вибирають з урахуванням того, що вірогідність нав'язування помилкових перешкод рівна $1/(2^P)$.

Перший блок відкритих даних T_0 (1) шифрують в режимі простої заміни $A'(\bullet)$ протягом перших 16-ти його циклів.

Одержане після 16-ти циклів 64-розрядне число $A'(T_0(1))$, підсумовують по модулю два з другим блоком даних $T_0(2)$. Результат підсумовування

$$(A'(T_0^{(1)}) \oplus T_0^{(2)})$$

знову піддають перетворенню $A'(\bullet)$. Потім процедура повторюється знов. Одержане 64-розрядне число $(A'(T_0^{(1)}) \oplus T_0^{(2)})$ підсумовують по модулю два з третім блоком $T_0^{(3)}$ і знову піддаються перетворенню $A'(\bullet)$ і так далі.

Останній блок $T_0^{(m)}$ (при необхідності добитий нулями до повного 64-розрядного блоку) підсумовують по модулю два з результатом підсумовування на кроці (m-1), після чого шифрують в режимі простої заміни за допомогою перетворення $A'(\bullet)$.

З одержаного 64-розрядного числа обирають відрізок $И_р$ (імітовставку), яка має довжину Р біт:

$$И_р = [a^{(m)}_{(32-p+1)}(16), a^{(m)}_{(32-p+2)}(16), \ldots, a^{(m)}_{32}(16)],$$

де $a_i^{(m)}$ - і-й біт 64-розрядного числа, що одержали після 16-го циклу останнього перетворення $A'(\bullet)$, $32-p+1 \le i \le 32$.

Імітовставка $И_р$ передається по каналу зв'язку наприкінці данних, що є зашифровані,

$$Т_ш^{(1)}, Т_ш^{(2)}, \ldots, Т_ш^{(m)}, И_р.$$

Зашифрованні данні, котрі були отримані одержувачем розшифровуваються та з блоків розшифровуванного тексту виробляють імітовставку $И_р'$. Ця імітовставка $И_р'$ дорівнюється з імітовставкою $И_р$, яку отримали водночас з шифртекстом.

Якщо $И_р' \neq И_р$, тоді данні, які отримали при розшифровуванні вважаються спотвореними.

Тема 7: **Основні схеми шифрування в асиметричній системі шифрування (4 години)**

Криптосистема шифрування даних RSA. Алгоритм RSA було запропоновано в 1978 році трьома авторами: Р. Райвестом (Rivest), А. Шаміром (Shamir) та А. Алдеманом (Aldeman). Алгоритм названо за першими буквами прізвищ його авторів. Алгоритм RSA став першим повноцінним алгоритмом з відкритим ключем,

який може працювати як у режимі шифрування даних, так і в режимі електронного цифрового підпису.

Надійність алгоритму базується на складності факторизації великих чисел та складності обчислення дискретних алгоритмів. В криптосистемі RSA відкритий ключ K_B, секретний ключ k_B, повідомлення M та криптограма C належать множині цілих чисел

$$Z_N=\{0,1,2,...,N-1\},$$

де N – модуль:

$$N=P*Q$$

Тут P і Q – випадкові великі прості числа. Для забезпечення максимальної безпеки P і Q вибирають однакової довжини і зберігають в секреті. Множина Z_N з операціями додавання та множення по модулю N утворює арифметику по модулю N.

Відкритий ключ K_B вибирають відкритим чином так, щоб виконувалися умови:

$$1<K_B\varphi(N),\ НСД(K_B,\varphi(N))=1,$$

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

де $\varphi(N)$ – функція Ейлера.

Функція Ейлера вказує кількість додатніх цілих чисел в інтервалі від 1 до N, які є взаємно простими з N [і]. Друга із вказаних вище умов означає, що відкритий ключ K_B та функція Ейлера $\varphi(N)$ повинні бути взаємно простими (їх найбільший спільний дільник (НСД) повинен бути рівним 1). Далі, використовуючи розширений алгоритм Евкліда, розраховують секретний ключ k_B, який задовольняє наступній умові:

$$k_B*K_B\equiv1(\mathrm{mod}\varphi(N)),$$

або

$$k_B=K_B^{-1}(\mathrm{mod}(P-1)(Q-1)).$$

Цей ключ легко розрахувати, оскільки отримувач знає пару простих чисел (P,Q) і може легко розрахувати $\varphi(N)$. Нагадаємо, що k_B та N повинні бути взаємно простими.

Відкритий ключ K_B використовують для шифрування даних, а секретний ключ k_B – для розшифровування. Перетворення шифрування визначає криптограму C через пару (відкритий ключ K_B, повідомлення M) відповідно до формули:

$$C = E_{K_B}(M) = E_B(M) = M^{K_B} (\mathrm{mod}\, N).$$

Знаходження функції, оберненої до $C = M^{K_B} (\mathrm{mod}\, N)$ (визначення значення M за відомими значеннями C, K_B та N), є практично нездійсненною задачею при $N\approx2^{512}$ [**Ошибка! Закладка не определена.**].

Проте обернену задачу – розшифровування криптограми C, – можна розв'язати, використовуючи пару – секретний ключ k_B та криптограма C за формулою:

$$M = D_{K_B}(C) = D_B(C) = C^{k_B} (\mathrm{mod}\, N).$$

Таким чином, якщо криптограму

$$C = M^{K_B} (\mathrm{mod}\, N)$$

піднести до степеня k_B, то в результаті відновлюється вихідний відкритий текст M, оскільки

$$\left(M^{K_B}\right)^{k_B} = M^{K_B k_B} = M^{n\varphi(N)+1} \equiv M (\mathrm{mod}\, N)$$

Таким чином, отримувач, який створює криптосистему, захищає два параметри:

1) секретний ключ k_B та 2) пару чисел (P,Q), добуток яких дає значення N. З іншої сторони, отримувач відкриває значення модуля N та відкритий ключ K_B.

Противнику відомі тільки значення K_B та N. Якщо б він зміг розкласти число N на множники P та Q, він зміг би обчислити значення функції Ейлера

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

та визначити значення секретного ключа k_B. Проте, як уже зазначалося, факторизація дуже великого N за допомогою обчислень є нездійсненною на даний час задачею (за умови, що довжина вибраних P та Q складає достатню кількість десяткових знаків).

Безпека та швидкодія криптосистеми RSA. Безпека алгоритму RSA базується на складності розв'язання задач факторизації великих чисел, які є добутком двох великих простих чисел. Дійсно, криптостійкість алгоритму RSA визначається тим, що після формування секретного ключа k_B та відкритого ключа K_B знищуються значення простих чисел P та Q, і тоді виключно складно визначити секретний ключ k_B за відкритим ключем K_B, оскільки для цього потрібно розв'язати задачу знаходження дільників P і Q та модуля N.

Розкладання величини N на прості множники P та Q дозволяє обчислити функцію $\varphi(N)=(P-1)(Q-1)$ та визначити секретне значення k_B, використовуючи рівняння:

$$K_B * k_B \equiv 1 (\mathrm{mod}\,\varphi(N)).$$

Іншим можливим способом криптоаналізу алгоритму RSA є безпосереднє обчислення чи підбір значення функції $\varphi(N)=(P-1)(Q-1)$. Якщо визначено значення $\varphi(N)$, то співмножники P та Q обчислюються досить просто, проте така задача не простіша за факторизацію модуля N.

Задача факторизації є складною задачею для великих значень модуля N. Спочатку автори алгоритму RSA пропонували для обчислення модуля N вибирати прості числа P та Q випадковим чином по 50 десяткових розрядів кожне. Вважалося, що такі великі числа N досить складно розкласти на прості множники. Один із авторів алгоритму RSA, Р.Райвест вважав, що для розкладання на прості множники числа із майже 130 десяткових цифр, наведеного у їх публікації, знадобиться більше 40 квадрільйонів років машинного часу [**Ошибка! Закладка не определена.**]. Проте цей прогноз не виправдався у зв'язку з відносно швидким прогресом комп'ютерів та їх обчислювальної потужності, а також покращення алгоритмів факторизації.

Один із найшвидших алгоритмів факторизації, відомих в наш час, – алгоритм NFS (Number Field Sieve) може виконати факторизацію великого числа N (з кількістю десяткових розрядів більше 120) за кількість кроків, яка оцінюється величиною

$$e^{2(\ln n)^{1/3}(\ln(\ln(n)))^{2/3}}.$$

В 1994 році було факторизовано число із 129 десятковими знаками. Це вдалося здійснити математикам А.Ленстра та М.Манассі з використанням організації розподілених обчислень на 1600 комп'ютерах, об'єднаних в мережу, на протязі восьми місяців. На думку А.Ленстра та М.Манассі, їх робота компрометує криптосистеми RSA та створює велику загрозу їх подальшим застосуванням. Тепер розробники криптоалгоритмів з відкритим ключем на базі RSA змушені уникати використання чисел довжиною менше 200 десяткових розрядів (таке число було факторизовано в 2005 році). Останні публікації пропонують застосовувати для цього числа довжиною не менше 250-300 десяткових розрядів.

Криптографи спробували розрахувати безпечні довжини ключів асиметричних криптосистем на найближчі 20 років виходячи з прогнозу розвитку комп'ютерів та їх обчислювальної потужності а також можливого удосконалення алгоритмів факторизації. Ці оцінки представлені в табл. 1. Оціночні дані наведено для трьох груп користувачів, відповідно до рівня вимог до їх інформаційної безпеки.

Таблиця 1 - Оцінки довжини ключів для асиметричних криптосистем, біт

Рік	**Окремі користувачі**	**Корпорації**	**Державні організації**
1995	768	1280	1536
2000	1024	1280	1536
2005	1280	1536	2048
2010	1280	1536	2048
2015	1536	2048	2048

Криптосистеми RSA реалізуються як апаратними, так і програмними засобами. Для апаратної реалізації операцій зашифровування та розшифровування RSA розроблено спеціальні процесори, які дозволяють виконувати операції, пов'язані з піднесенням до степеня по модулю N за відносно короткий час. І всеж таки апаратна реалізація RSA є приблизно в 100 разів повільнішою апаратної реалізації симетричного криптоалгоритму DES. Програмна реалізація RSA також приблизно в 100 разів повільніша програмної реалізації DES. З розвитком технологій ці оцінки можуть дещо змінюватися, але асиметрична криптосистема RSA ніколи не досягне швидкодії симетричних криптосистем. Слід зазначити, що невелика швидкодія криптосистем RSA обмежує область їх застосування, але не перекреслює їх цінності.

Схема шифрування Поліга-Хеллмана. Схема шифрування Поліга-Хеллмана подібна до схеми шифрування RSA. Вона представляє собою несиметричний алгоритм, оскільки використовуються різні ключі для шифрування і розшифровування. В той же час цю схему не можна віднести до класу криптосистем з відкритим ключем, оскільки ключі шифрування та розшифровування легко виводяться один з одного. Обидва ключі потрібно тримати у секреті.

Аналогічно до системи RSA криптограма C та відкритий текст P визначається із співвідношень:

$$C=P^e \bmod n,$$
$$P=C^d \bmod n.$$

де $e*d\equiv 1$ (по модулю деякого числа n).

На відміну від алгоритму RSA в цій схемі число n не визначається через два великі прості числа; число n повинно залишатися частиною секретного ключа. Якщо хтось знатиме e та n, то він зможе розрахувати значання d. Не знаючи значень e чи d, противник буде змушений розраховувати значення

$$e=\log_P C (\bmod n).$$

Відомо, що це є складною задачею. Схема шифрування Поліга-Хеллмана запатентована в США та Канаді.

Схема шифрування Ель Гамаля. Схема Ель Гамаля, запропонована в 1985 році, може використовуватися як для шифрування, так і для цифрових підписів. Безпека схеми Ель Гамаля зумовлена складністю розрахунку дискретних логарифмів в кінцевому полі [ii].

Для генерації пари ключів (відкритий ключ – секретний ключ), спочатку вибирають

деяке велике просте число P та велике ціле число G, причому $G<P$. Числа P та G можуть бути розповсюджені серед групи користувачів. Потім вибирають випадкове число X, причому $X<P$. Число X є секретним ключем і повинно зберігатися в таємниці. Потім обчислюють $Y=G^X \bmod P$. Число Y є відкритим ключем.

Для того, щоб зашифрувати повідомлення M, вибирають випадкове ціле число K, $1<K<P-1$, таке, що числа K та $(P-1)$ є взаємно простими. Потім вираховують числа

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

$$b=Y^K M \bmod P.$$

Пара чисел (a, b) є шифротекстом. Зазначимо, що довжина шифротексту вдвічі більша довжини вихідного відкритого тексту M. Для розшифровування шифротексту (a, b), обчислюють

$$M=b/a^X \bmod P.$$

Комбінований метод шифрування. Головною перевагою криптосистем з відкритим ключем є їх потенційно висока безпека: немає необхідності ні передавати, ні повідомляти будь-кому значення секретних ключів, ні впевнюватися в їх справжності. Проте алгоритми, що лежать в основі криптосистем з відкритим ключем, мають ряд недоліків:

– генерація нових секретних та відкритих ключів базується на генерації нових великих простих чисел, а перевірка простоти числа займає багато процесорного часу;

– процедури шифрування та розшифровування, зв'язані з піднесенням до степеня багатозначного числа є досить громіздкими.

У зв'язку з цим швидкодія криптосистем з відкритим ключем зазвичай в сотні та більше разів нижча швидкодії симетричних криптосистем.

Комбінований (гібридний) метод шифрування дозволяє поєднувати переваги високої секретності, притаманні асиметричним системам з відкритим ключем, з перевагами високої швидкості роботи, які притаманні криптосистемам із секретним ключем. При такому підході криптосистема з відкритим ключем застосовується тільки для шифрування, передачі та наступного розшифровування тільки секретного ключа симетричної криптосистеми. А симетрична криптосистема застосовується для шифрування і передачі вихідного відкритого тексту. В результаті криптосистема з відкритим ключем не заміняє симетричну криптосистему з секретним ключем, а тільки доповнює її, забезпечуючи кращу захищеність інформації. Такий підхід іноді називають схемою електронного цифрового конверта.

Якщо користувач A хоче передати зашифроване комбінованим методом повідомлення M користувачу B, то порядок його дій буде наступним.

1. Створити симетричний ключ, який в цьому методі називають ***сеансовим ключем*** K_S.

2. Зашифрувати повідомлення M на сеансовому ключі K_S.

3. Зашифрувати сеансовий ключ K_S на відкритому ключі K_B користувача B.

4. Передати відкритим каналом зв'язку на адресу користувача B зашифроване повідомлення разом із зашифрованим сеансовим ключем.

Дії користувача B при отриманні зашифрованого повідомлення та зашифрованого сеансового ключа повинні бути зворотніми:

5. Розшифрувати на своєму секретному ключі k_B сеансовий ключ K_S.

6. За допомогою отриманого сеансового ключа K_S розшифрувати та прочитати повідомлення M.

При використанні комбінованого методу шифрування можна бути впевненим в тому,

що тільки користувач *B* зможе правильно розшифрувати ключ K_S та прочитати повідомлення *M*.

Таким чином, при комбінованому методі шифрування застосовуються криптографічні ключі як симетричних, так і асиметричних криптосистем. Очевидно, що вибір довжин ключів для кожного типу криптосистеми слід здійснювати таким чином, щоб зловмиснику було однаково складно атакувати будь-який механізм захисту комбінованої криптосистеми. В табл.2 наведено розповсюджені довжини ключів симетричних та асиметричних криптосистем, для яких складність атаки повного перебору приблизно рівна складності факторизації відповідних модулів асиметричних криптосистем.

Таблиця 2 - Довжини ключів для симетричних та асиметричних криптосистем при однаковій їх криптостійкості

Довжина ключа симетричної криптосистеми, біт	**Довжина ключа асиметричної криптосистеми, біт**
56	384
64	512
80	768
112	1792
128	2304

Комбінований метод допускає можливість виконання процедури автентифікації. Для цього користувач *A* на основі функції хешування повідомлення та свого секретного ключа k_A за допомогою відомого алгоритму електронного цифрового підпису генерує свій підпис та записує його, наприклад, в кінець повідомлення. Користувач *B*, прочитавши прийняте повідомлення, може впевнитися в справжності цифрового підпису абонента *A*. Використовуючи той же алгоритм електронного цифрового підпису та результат хешування прийнятого повідомлення, користувач *B* перевіряє отриманий підпис. Комбінований метод шифрування є найбільш раціональним, оскільки поєднує в собі високу швидкодію симетричного шифрування та високу криптостійкість, яку гарантують системи з відкритим ключем.

Тема 8: **Електронний підпис (ЕП). Історія виникнення, призначення застосування**

1 Історія виникнення

У 1976 році УітфілдомДіффі і Мартіном Хеллманом було вперше запропоновано поняття «електронний цифровий підпис», хоча вони всього лише припускали, що схеми ЕЦП можуть існувати.

У 1977 році, Рональд Ривест, Аді Шамір і Леонард Адлеман розробили криптографічний алгоритм RSA, який без додаткових модифікацій можна використовувати для створення примітивних цифрових підписів.

Незабаром після RSA були розроблені інші ЕЦП, такі як алгоритми цифрового підпису Рабина, Меркле.

У 1984 році Шафи Гольдвассер, Сільвіо Мікалі і Рональд Ривест першими строго визначили вимоги безпеки до алгоритмів цифрового підпису. Ними були описані моделі атак на алгоритми ЕЦП, а також запропонована схема GMR, що відповідає описаним вимогам (криптосистема Гольдвассера - Мікалі).

2 Електронний підпис (ЕП), Електронний цифровий підпис (ЕЦП):

- інформація в електронній формі, приєднана до іншої інформації в електронній формі (електронний документ) або іншим чином пов'язана з такою інформацією;

- використовується для визначення особи, яка підписала інформацію (електронний документ).

За своєю суттю електронний підпис є *реквізитом електронного документа*, що дозволяє:

1) встановити відсутність спотворення інформації в електронному документі з моменту формування ЕП;

2) перевірити приналежність підпису власникові сертифіката ключа ЕП.

Значення реквізиту виходить за результатом криптографічного перетворення інформації з використанням закритого ключа ЕП.

3 Призначення і застосування ЕП

Електронний підпис призначений для ідентифікації особи, яка підписала електронний документ, і є повноцінною заміною (аналогом) власноручного підпису у випадках, передбачених законом.

Використання електронного підпису дозволяє здійснити:

- *Контроль цілісності переданого документа:* при будь-якій випадковій або навмисній зміні документа підпис стане недійсним, тому що він обчислений на підставі вихідного стану документа і відповідає лише йому.

-*Захист від змін (підроблення) документа:* гарантія виявлення підробки при контролі цілісності робить підроблення недоцільним в більшості випадків.

- *Неможливість відмови від авторства.* Так як створити коректний підпис можливо, лише знаючи закритий ключ, а він відомий тільки власнику, він не може відмовитися від свого підпису під документом.

Доказове підтвердження авторства документа: Так як створити коректний підпис можливо, лише знаючи закритий ключ, а він відомий тільки власнику, він може довести своє авторство підпису під документом. Залежно від деталей визначення документа можуть бути підписані такі поля, як «автор», «внесені зміни», «мітка часу» і т. д.

Тема 9: Використання механізму електронного цифрового підпису

План

1 Поняття електронного цифрового підпису

2 Механізм електронного цифрового підпису

3 Загрози ЕЦП

Ключові слова: електронний підпис, електронний цифровий підпис, особистий ключ

1 Поняття електронного цифрового підпису

Важливим механізмом підтвердження дійсності даних і забезпечення невідмовності

об’єкта від ознайомлення/підписання даних є механізм електронного підпису.

Електронно-цифровий підпис (ЕЦП) – вид електронного підпису, отриманого за результатом криптографічного перетворення набору електронних даних, який додається до цього набору або логічно з ним поєднується і дає змогу підтвердити його цілісність та ідентифікувати підписувача. ЕЦП накладається за допомогою особистого ключа та перевіряється за допомогою відкритого ключа [35].

Закон України про цифровий підпис є законодавчою основою застосування електронного підпису й формування електронного документообігу. У ньому розкриваються такі поняття.

Електронний підпис — дані в електронній формі, які додаються до інших електронних даних або логічно з ними пов’язані та призначені для ідентифікації підписувача цих даних.

Електронний цифровий підпис (ЕЦП) — вид електронного підпису, отриманого за результатом криптографічного перетворення набору електронних даних, який додається до цього набору або логічно з ним поєднується і дає змогу підтвердити його цілісність та ідентифікувати підписувача. (ЕЦП) представляє собою відносно невеликий об’єм додаткової автентифікуючої цифрової інформації, що передається разом із „підписаними“ даними.

Для реалізації ЕЦП використовуються принципи асиметричного шифрування.

Система ЕЦП включає процедуру формування цифрового підпису відправником з використанням секретного ключа відправника та процедуру перевірки підпису отримувачем з використанням відкритого ключа відправника.

Електронний цифровий підпис накладається за допомогою особистого ключа та перевіряється за допомогою відкритого ключа.

Особистий ключ — параметр криптографічного алгоритму формування електронного цифрового підпису, доступний тільки підписувачу.

Відкритий ключ — параметр криптографічного алгоритму перевірки електронного цифрового підпису, доступний суб’єктам відносин у сфері використання електронного цифрового підпису.

Засвідчення чинності відкритого ключа — процедура формування сертифіката відкритого ключа.

Сертифікат відкритого ключа — документ, виданий центром сертифікації ключів, який засвідчує чинність і належність відкритого ключа підписувачу. Сертифікати ключів можуть поширюватися в електронній формі або у формі документа на папері та використовуватися для ідентифікації особи підписувача. Вони застосовуються в багатьох захищених протоколах, у тому числі, IPSec і SSL.

ЕЦП потрібен для забезпечення діяльності фізичних і юридичних осіб, здійснюваної з використанням електронних документів. ЕЦП використовується суб’єктами електронного документообігу для ідентифікації особи, що підписує, і підтвердження цілісності даних, що надаються в електронній формі. Алгоритмічною основою ЕЦП є комбінування методів шифрування й хешування.

2 Механізм електронного цифрового підпису

Для контролю цілісності переданих мережами даних використовується електронний цифровий підпис, який реалізується за методом шифрування з відкритим ключем.

Електронний цифровий підпис є відносно невелика кількість додаткової аутентифицирующей інформації, переданої разом з підписується текстом. Відправник

формує цифровий підпис, використовуючи секретний ключ відправника. Одержувач перевіряє підпис, використовуючи відкритий ключ відправника.

Ідея технології електронного підпису полягає в наступному. Відправник передає два примірники одного повідомлення: відкрите і розшифроване його закритим ключем (тобто назад шифрування). Одержувач шифрує за допомогою відкритого ключа відправника розшифрований екземпляр. Якщо він співпаде з відкритим варіантом, то особистість і підпис відправника вважається встановленими.

При практичній реалізації електронного підпису також шифруються не всі повідомлення, а лише спеціальна контрольна сума - *хеш*, що захищає послання від нелегального зміни. Електронний підпис тут гарантує як цілісність повідомлення, так і засвідчує особу відправника.

Безпека будь-якої криптосистеми визначається використовуваними криптографічними ключами. У разі ненадійного управління ключами зловмисник може заволодіти ключовою інформацією і отримати повний доступ до всієї інформації в системі або мережі. Розрізняють такі види функцій управління ключами: *генерація, зберігання і розподіл ключів.*

Способи генерації ключів для симетричних і асиметричних криптосистем різні. Для генерації ключів симетричних криптосистем використовуються апаратні і програмні засоби генерації випадкових чисел. Генерація ключів для асиметричних криптосистем складніша, тому що ключі повинні володіти певними математичними властивостями.

Функція зберігання передбачає організацію безпечного зберігання, обліку та видалення ключової інформації. Для забезпечення безпечного зберігання ключів застосовують їх шифрування за допомогою альтернативних джерел. Такий підхід призводить до концепції ієрархії ключів. У ієрархію ключів зазвичай входить головний ключ (тобто, майстер-ключ), ключ шифрування ключів і ключ шифрування даних. Слід зазначити, що генерація і зберігання майстер-ключа є найбільш критичним питанням криптозахисту.

Розподіл - найвідповідальніший процес в управлінні ключами. Цей процес повинен гарантувати скритність розподілу ключів, а також бути оперативним і точним. Між користувачами мережі ключі розподіляють двома способами:

- за допомогою прямого обміну сеансовими ключами;
- використовуючи один або кілька центрів розподілу ключів.

Сучасні механізми захисту ЕЦП

Сучасні механізми захисту ЕЦП базуються на таких асиметричних криптосистемах: криптосистема RSA; криптосистема ECC; криптосистема Ель-Гамалі.

Криптосистема RSA

є широким прикладом криптосистеми з відкритим ключем. Вона всебічно досліджена і визнана криптостійкою при достатній довжині ключів. Її стійкість заснована на трудомісткості розкладанні на множники великих чисел. Однак зі зростанням потужності процесорів криптосистема RSA може втратити стійкість до атаки повного перебору, тому збільшення потужності процесорів приводить до застосування більш довгих ключів, що підвищує її стійкість.

Криптосистема RSA – дозволяє вирішити завдання підтвердження істинності електронного документа. Ця можливість заснована на тому, що зашифрувати дані, використовуючи закритий ключ замість відкритого ключа може тільки той, кому закритий ключ відомий. При цьому існує можливість перевірки застосування закритого

ключа до даних без його розкриття. На сьогодні RSA є найбільш поширеною криптосистемою – стандартом для багатьох криптографічних додатків.

Криптосистема ECC– система з відкритим ключем, що використовує алгебраїчну систему, яка описується в термінах точок еліптичних кривих, для реалізації асиметричного алгоритму шифрування. Широкому впровадженню ECC довго заважала слабка вивченість її математичного фундаменту криптосистеми, але проведені дослідження не виявили в її технології серйозних недоліків.

Стійкість шифрування системи ECC базується на складності задачі дискретного логарифмування, при цьому висока стійкість криптосистеми досягається при значно менших довжинах ключів, ніж у RSA. Згідно з рекомендаціями Національного інституту стандартів і технологій США, еквівалентом Збірник наукових праць Центру воєнно-стратегічних досліджень Національного університету оборони України 39 1024-бітного ключа RSA, наприклад, є ECC-ключ довжиною всього 163 біта (співвідношення 6:1). Причому ця залежність нелінійна, тому для 512- бітного ключа ECC розмір аналога в системі RSA складає вже 15360 біт (співвідношення 30:1). Її сучасні реалізації показують, що вона набагато ефективніша, ніж інші системи з відкритими ключами. На сьогодні стандарт електронного цифрового підпису України ґрунтується на еліптичних кривих (ДСТУ 4145-2002 який набув чинності з 1 липня 2003 року).

Криптосистема Ель-Гамалі – система з відкритим ключем, яка заснована на обчисленні дискретних логарифмів у кінцевому полі та складається з алгоритму шифрування і алгоритму ЕЦП.

Головною перевагою схеми цифрового підпису Ель-Гамалі є можливість виробляти цифрові підписи для великого числа повідомлень з використанням тільки одного закритого ключа. Щоб порушнику підробити підпис, йому потрібно вирішити складні математичні завдання із знаходженням логарифма в кінцевому полі. Недоліком цієї системи є відсутність семантичної стійкості. На відміну від RSA алгоритм ЕльГамалі не був запатентований і, тому, став більш дешевою альтернативою, оскільки не вимагалася оплата внесків за ліцензію. Схема Ель-Гамалі полягає в основі стандартів електронного цифрового підпису в США (DSA) і Росії (ГОСТ Р 34.10-94).

3 Загрози ЕЦП

Для ЕЦП можна виділити такі види загроз:

екзистенціальна підробка (криптоаналітик може створити для довільного повідомлення правильний цифровий підпис);

селективна підробка (загроза створення для раніше обраного повідомлення правильного цифрового підпису);

універсальна підробка (використання ненадійності математичного стандарту ЕЦП для визначення іншого функціонально еквівалентного алгоритму ЕЦП, і використання його для створення або модифікації повідомлень);

повне розкриття (успішне розв’язання завдання криптоаналізу й знаходження секретного ключа — можливість створювати підписи для довільних повідомлень, а потім нав’язувати ці повідомлення).

На сьогодні для формування ЕЦП переважно використовуються перетворення в групі точок еліптичних кривих над простими полями.

Повноцінне використання механізму електронного цифрового підпису неможливо без впровадження інфраструктури відкритого ключа (Public Key Infrastructure, PKI) — комплексу програмного й апаратного забезпечення, кадрів, а також політик і процедур,

необхідних для створення, управління, зберігання, поширення й анулювання сертифікатів відкритих ключів. Призначення PKI — управління ключами й сертифікатами. Для того щоб PKI могла виконувати ці функції, потрібно реалізувати такі елементи:

- сертифікати відкритих ключів і центри сертифікації;
- збереження й відновлення ключів;
- підтримка прийняття цифрового підпису;
- автоматичне відновлення пар ключів і сертифікатів;
- управління архівом ключів;
- підтримка взаємної сертифікації.

Центр сертифікації (ЦС) є представником об'єкта при створенні цифрових сертифікатів. Ці сертифікати зв'язують імена об'єктів з їхніми відкритими ключами. ЦС діє в PKI як довірена особа, і якщо об'єкти довіряють ЦС видачу й управління сертифікатами, вони можуть довіряти й сертифікатам, виданим цим ЦС.

Центр сертифікації створює сертифікати об'єктів шляхом цифрового підпису набору даних, що містить таку інформацію (і додаткові елементи):

- повне ім'я об'єкта (ім'я, номер об'єкта, пароль і будь-які додаткові атрибути, необхідні для однозначної ідентифікації);
- відкритий ключ об'єкта;
- строк дії сертифіката;
- конкретні операції, у яких цей відкритий ключ може бути використаний (ідентифікація, шифрування або й те й інше).

Те, що центр сертифікації підписує сертифікат, засвідчує, що будь-які спроби зміни змісту сертифіката можна буде помітити. Дійсність сертифіката визначається шляхом перевірки підпису ЦС. Оскільки підпис ЦС на сертифікаті можна перевірити, то можна вважати, що сертифікат захищений від підробок і що йому можна довіряти.

Сертифікати відкритих ключів використовують у процесі валідації (підтвердження) завірених цифровим підписом даних, коли одержувач перевіряє, щоб:

1) інформація, що ідентифікує відправника, відповідала даним, що містяться в сертифікаті;

2) жоден сертифікат з ланцюжка сертифікатів не був анульований, і в момент підписання повідомлення всі сертифікати були дійсними;

3) сертифікат використовувався відправником за цільовим призначенням;

4) дані не були змінені з моменту створення ЕЦП.

У результаті перевірок одержувач може прийняти дані, підписані відправником, якщо засвідчувальний центр (ЗЦ) перевіряє запит на сертифікат. Якщо той проходить верифікацію, то підписується й випускається сертифікат. Для публікації сертифікат направляється до реєстру сертифікатів; залежно від конкретної конфігурації PKI ця функція може бути покладена на реєстраційний або засвідчувальний центр.

На рис. 1 показано всі можливі комунікації між кінцевим суб'єктом і засвідчувальним центром. Процес анулювання сертифіката аналогічний процесу його генерації. Кінцевий об'єкт запитує ЗЦ про анулювання свого сертифіката, реєстраційний центр приймає рішення й направляє запит про анулювання в ЗЦ. ЗЦ вносить зміни до списку анульованих сертифікатів і публікує його в реєстрі.

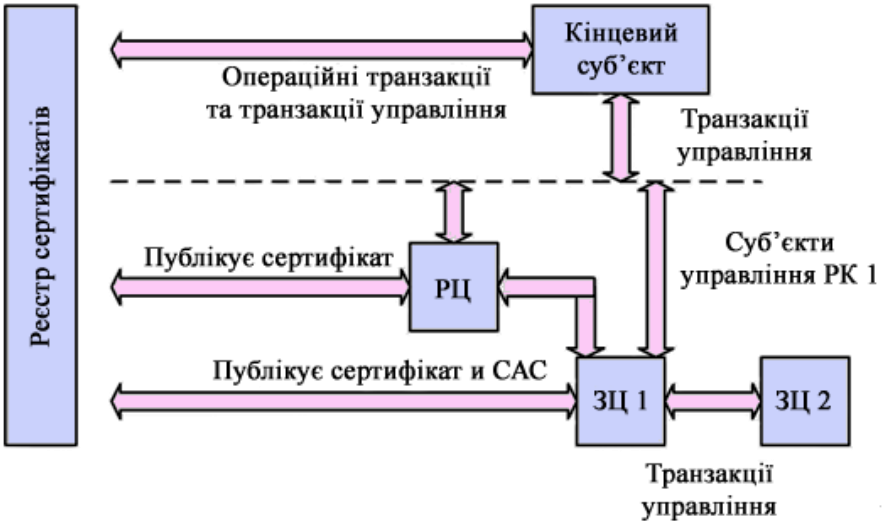


Рисунок 1 - Схема функціонування PKI

Кінцеві об'єкти можуть перевірити дійсність конкретного сертифіката через операційний протокол.

Операційні протоколи — це протоколи для доставки сертифікатів (або інформації про їхній статус) і списків анульованих сертифікатів до клієнтських систем, що використовують сертифікати. Існують різні механізми поширення сертифікатів і їх анулювання з використанням протоколів LDAP, HTTP і FTP.

Атрибутні сертифікати (AC) призначені для зв'язування ідентифікаційної інформації із сертифікатом. Формат атрибутного сертифіката дозволяє зв'язати будь-яку додаткову інформацію про власника із сертифікатом відкритого ключа, включаючи в структуру даних, завірених цифровим підписом, посилання на один або кілька сертифікатів відкритих ключів того самого суб'єкта. Атрибутний сертифікат може мати кілька призначень (наприклад, призначатися для доступу до веб-сервера або хосту електронної пошти). Застосування AC пов'язане з появою потреби в управлінні доступом, заснованому на певних принципах, ролях або посадах, що властива сучасним телекомунікаційним додаткам.

Успіх розгортання PKI багато в чому залежить від навколишньої й підтримувальної інфраструктури: основних засобів, обладнання й персоналу. Безпека практики сертифікації залежить від якості розробки політики PKI і регламенту засвідчувального центру. PKI набула застосування в багатьох сферах діяльності людини, де використовуються телекомунікації й необхідно забезпечити надійний захист даних й операцій: банківська й фінансова сфера (автентифікація платежів, контроль доступу, захищена електронна пошта, захищене зберігання й пошук документів, цифровий нотаріат, захищені транзакції), охорона здоров'я (захищене зберігання й пошук документів, кваліфікаційна ідентифікація, персональна ідентифікація), урядова сфера й сфера бізнесу (контроль доступу, автентифікація платежів, захищений обмін повідомленнями, захищене зберігання й пошук документів).

Тема 10: **Проблема автентифікації даних та електронний цифровий підпис (4 години)**

При обміні електронними документами по мережі суттєво знижуються затрати на обробку та зберігання документів. Однак при цьому виникає проблема автентифікації автора документу та самого документу – встановлення автентичності автора та відсутності змін в отриманому документі. При звичайному (паперовому) документообгові ці проблеми вирішуються за рахунок того, що інформація в документі та рукописний підпис автора жорстко зв'язані з фізичним носієм (папером). В електронних документах на машинних носіях такого зв'язку немає.

Метою автентифікації електронних документів є їх захист від можливих видів зловмисних дій, до яких відносяться [**Ошибка! Закладка не определена.**]:

– ***активне перехоплення*** – порушник, підключившись до мережі, перехоплює документи (файли) і змінює їх;

– ***маскарад*** – абонент *C* посилає документ абоненту *B* від імені абонента *A*;

– ***ренегатство*** – абонент *A* заявляє, що не відправляв повідомлення абоненту *B*, хоча в дійсності відправляв;

– ***підміна*** – абонент *B* змінює чи формує новий документ і заявляє, що отримав його від абонента *A*;

– ***повтор*** – абонент *C* повторює переданий раніше документ, який абонент *A* відправляв абоненту *B*.

Ці види зловмисних дій можуть нанести суттєві збитки різноманітним підприємствам, організаціям та приватним особам, що застосовують у своїй діяльності комп'ютерні інформаційні технології. При обробці документів в електронній формі стають непридатними традиційні способи встановлення автентичності за рукописним підписом та відтиском печатки на паперовому документі. Принципово новим рішенням є електронний цифровий підпис (ЕЦП).

Електронний цифровий підпис використовується для автентифікації файлів, що передаються по телекомунікаційних каналах зв'язку. Функціонально він аналогічний звичайному рукописному підпису і володіє усіма його основними перевагами:

– засвідчує, що підписаний документ виходить від особи, яка поставила свій підпис;

– не дає цій особі можливості відмовитися від обов'язків, зв'язаних із підписаним документом;

– гарантує цілісність підписаного документу.

Цифровий підпис представляє собою відносно невеликий об'єм додаткової цифрової інформації, що передається разом із підписаним документом.

Система ЕЦП включає дві процедури: 1) процедуру встановлення підпису; 2) процедуру перевірки підпису. У процедурі встановлення підпису використовується секретний ключ відправника повідомлення, в процедурі перевірки підпису – відкритий ключ відправника.

При формуванні ЕЦП відправник у першу чергу обчислює хеш-функцію $h(M)$ підписуваного тексту M. Обчислене значення хеш-функції $h(M)$ представляє собою один короткий блок інформації m, який характеризує весь текст M в цілому. Потім

число m шифрується секретним ключем відправника. Отримана при цьому пара чисел представляє собою ЕЦП для даного тексту M. При перевірці ЕЦП отримувач повідомлення знову обчислює хеш-функцію $m=h(M)$ прийнятого по каналу зв'язку тексту M, після чого за допомогою відкритого ключа відправника перевіряє, чи відповідає отриманий підпис обчисленому значенню m хеш-функції.

Принциповим моментом в системі ЕЦП є неможливість підробки ЕЦП користувача без знання його секретного ключа підписування. В якості підписуваного документу може бути використано будь-який файл. Підписаний файл створюється із непідписаного шляхом додавання до нього одного чи більше електронних підписів. Кожний підпис вміщує наступну інформацію:

– дату підпису;

– термін закінчення дії ключа даного підпису;

– інформацію про особу, що підписала файл (ПІП, посада, скорочена назва підприємства/організації);

– ідентифікатор особи, яка підписала документ (ім'я відкритого ключа);

– власне, сам цифровий підпис.

Однонаправлені хеш-функції. Хеш-функція призначена для стиснення підписуваного документу M до кількох десятків чи сотень біт. Хеш-функція $h(\cdot)$ приймає в якості аргументу повідомлення (документ) M довільної довжини і повертає хеш-значення $h(M)=H$ фіксованої довжини. Зазвичай хешована інформація є стиснутим двійковим представленням основного повідомлення довільної довжини. Слід зазначити, що значення хеш-функції $h(M)$ складним чином залежить від документу M і не дозволяє відновити сам документ M. Хеш-функція повинна задовольняти цілому ряду умов [iii]:

– хеш-функція повинна бути чутливою до усіх можливих змін в тексті M, таким як вставки, видалення, перестановки тощо;

– хеш-функція повинна мати властивості необоротності – задача підбору документу M', який би володів потрібним значенням хеш-функції повинна бути нерозв'язною обчислювальними методами;

– ймовірність того, що значення хеш-функцій двох різних документів (не залежно від їх довжини) співпадуть, повинна бути надзвичайно малою.

Більшість хеш-функцій будується на основі однонаправленої функції $f(\cdot)$, яка утворює вихідне значення довжиною n при заданні двох вхідних значень довжиною n. Цими вхідними значеннями є блок вихідного тексту M_i та хеш-значення H_{i-1} попереднього блоку тексту, рис. 1:

$$H_i=f(M_i,H_{i-1}).$$

Хеш-значення, яке обчислюється для останнього блоку тексту стає хеш-значенням всього повідомлення M. В результаті однонаправлена хеш-функція завжди формує вихід фіксованої довжини n, незалежно від довжини вхідного тексту.

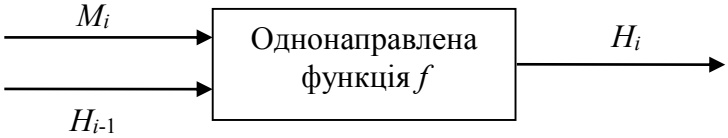


Рисунок 1 - Побудова однонаправленої функції

Однонаправлені хеш-функції на основі симетричних блочних алгоритмів.

Однонаправлену хеш-функцію можна побудувати, використовуючи симетричний блочний алгоритм. Найбільш очевидний підхід полягає в тому, щоб шифрувати повідомлення *M* з використанням блочного алгоритму в режимі CBC чи CFB за допомогою фіксованого ключа та деякого вектора ініціалізації IV. Останній блок шифротексту можна розглядати в якості хеш-значення повідомлення *M*. При такому підході не завжди можна побудувати безпечну однонаправлену хеш-функцію, але завжди можна отримати код автентифікації повідомлення MAC (Message Autentication Code).

Безпечніший варіант хеш-функції можна отримати, використовуючи блок повідомлення в якості ключа, попереднє хеш-значення – в якості входу, а поточне хеш-значення – в якості виходу. Реальні хеш-функції проектуються іще складніше. Довжина блоку зазвичай визначається довжиною ключа, а довжина хеш-значення співпадає з довжиною блоку. Оскільки більшість блочних алгоритмів є 64-бітними, деякі схеми хешування проектують так, щоб хеш-значення мало довжину, рівну подвійній довжині блоку.

Якщо прийняти, що отримується коректна хеш-функція, безпека системи хешування базується на безпеці блочного алгоритму, що лежить в її основі. Схема хешування, у якої довжина хеш-значення рівна довжині блоку, наведена на рис. 2.

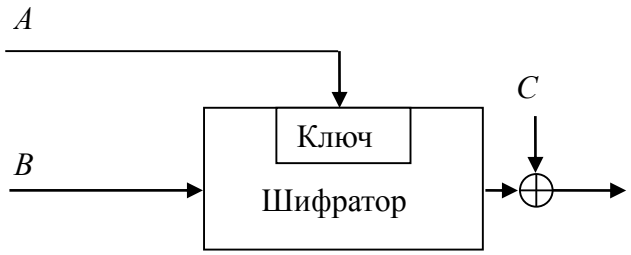


Рисунок 2 - Узагальнена схема формування хеш-функції

Робота цієї схеми описується виразами:

$$H_o=I_H,$$

$$H_i=E_A(B)\oplus C,$$

де I_H – деяке випадкове початкове значення; *A*, *B* та *C* можуть приймати значення M_i, H_{i-1}, $(M_i \oplus H_{i-1})$ або бути константами. Повідомлення *M* розбивається на блоки M_i прийнятої довжини, які обробляються один за одним.

Три різні змінні *A*, *B* та *C* можуть приймати одне із чотирьох можливих значень, у зв'язку з чим в принципі можна отримати 64 варіанти загальної схеми цього типу. Із них 52 варіанти є або тривіально слабкими, або небезпечними. Інші 12 безпечних схем хешування наведено в табл. 1.

Таблиця 1 - Схеми безпечного хешування у яких довжина хеш-значення рівна довжині блоку

Номер схеми	**Функція хешування**
1	$H_i = E_{H_{i-1}}(M_i) \oplus M_i$
2	$H_i = E_{H_{i-1}}(M_i \oplus H_{i-1}) \oplus M_i \oplus H_{i-1}$
3	$H_i = E_{H_{i-1}}(M_i) \oplus H_{i-1} \oplus M_i$
4	$H_i = E_{H_{i-1}}(M_i \oplus H_{i-1}) \oplus M_i$
5	$H_i = E_{M_i}(H_{i-1}) \oplus H_{i-1}$

6	$H_i = E_{M_i}(M_i \oplus H_{i-1}) M_i \oplus H_{i-1}$
7	$H_i = E_{M_i}(H_{i-1}) M_i \oplus H_{i-1}$
8	$H_i = E_{M_i}(M_i \oplus H_{i-1}) \oplus H_{i-1}$
9	$H_i = E_{M_i \oplus H_{i-1}}(M_i) \oplus M_i$
10	$H_i = E_{M_i \oplus H_{i-1}}(H_{i-1}) \oplus H_{i-1}$
11	$H_i = E_{M_i \oplus H_{i-1}}(M_i) \oplus H_{i-1}$
12	$H_i = E_{M_i \oplus H_{i-1}}(H_{i-1}) \oplus M_i$

Перші чотири з цих схем, які є безпечними при усіх атаках, графічно представлені на рис. 3.

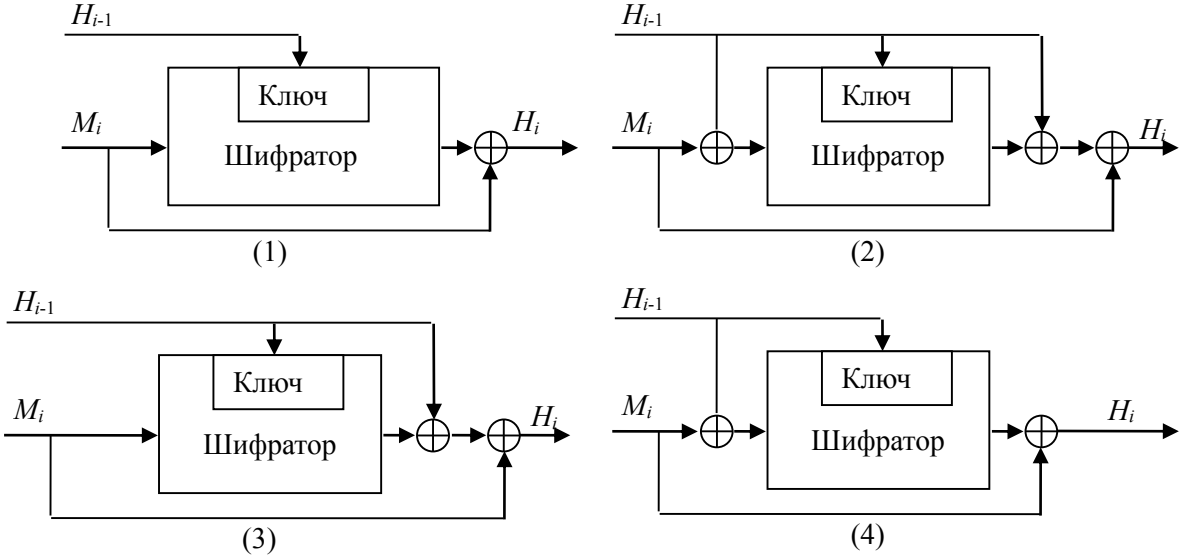


Рисунок 3- Чотири схеми безпечного хешування

Алгоритми електронного цифрового підпису.

Технологія застосування системи ЕЦП передбачає наявність мережі абонентів, які пересилають один одному підписані електронні документи. Для кожного абонента генерується пара ключів: секретний та відкритий. Секретний ключ зберігається абонентом у таємниці і використовується ним для формування ЕЦП. Відкритий ключ відомий усім іншим користувачам і призначений для перевірки ЕЦП отримувачем підписаного електронного документу. Відкритий ключ не дозволяє вирахувати секретний ключ.

Для генерації пари ключів (секретного та відкритого) в алгоритмах ЕЦП, як і в асиметричних системах шифрування, використовують різні математичні схеми, що базуються на використанні однонаправлених функції. Ці схеми поділяються на дві групи. В основі такого поділу лежать відомі складні обчислювальні задачі:

– задача факторизації (розкладання на множники) великих цілих чисел;
– задача дискретного логарифмування.

Алгоритм цифрового підпису RSA. Першою реально працюючою і найвідомішою в усьому світі системою ЕЦП стала система RSA, математична схема якої була розроблена у 1977 році в Масачусетському технологічному інституті США [iv].

Спочатку потрібно обчислити пару ключів (секретний та відкритий). Для цього

відправник (автор) електронних документів обчислює два великих простих числа P та Q, після чого знаходить їх добуток

$$N=PQ$$

і значення функції

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

Далі відправник обчислює число E з умов:

$$E\leq\varphi(N),\ \text{НСД}(E,\ \varphi(N))=1$$

та число D з умов:

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

Пара чисел (E, N) є відкритим ключем. Цю пару чисел автор передає партнерам по переписці для перевірки його цифрових підписів. Число D зберігається автором як секретний ключ для підписування. Узагальнена схема формування та перевірки цифрового підпису RSA представлена на рис. 5.18. Припустимо, що відправник хоче підписати повідомлення M перед його відправкою. Спочатку повідомлення M стискують за допомогою хеш-функції $h(\cdot)$ в ціле число m:

$$m=h(M).$$

Потім розраховують цифровий підпис S під електронним документом M, використовуючи хеш-значення m та секретний ключ D:

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

Пара (M,S) передається партнеру-отримувачу як електронний документ M, підписаний цифровим підписом S, причому підпис S сформовано власником секретного ключа D.

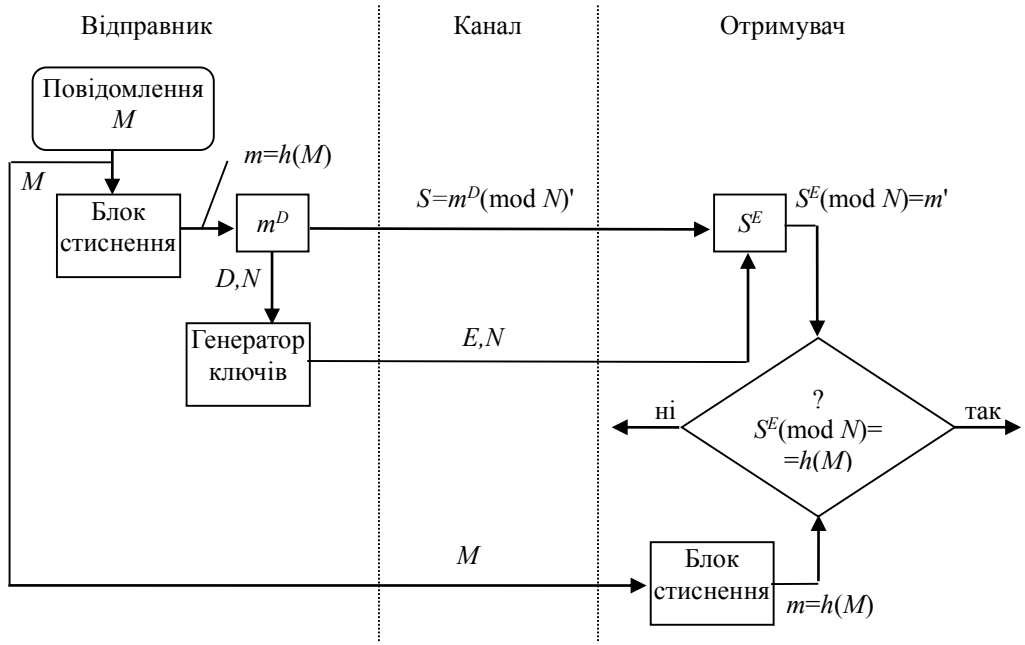


Рисунок 5 - Узагальнена схема цифрового підпису RSA

Після прийому пари (M,S) отримувач обчислює хеш-значення повідомлення M двома різними способами. Перш за все він відновлює хеш-значення m', застосовуючи криптографічне перетворення підпису S з використанням відкритого ключа E:

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

Крім того, він знаходить результат хешування прийнятого повідомлення M за допомогою такої ж хеш-функції $h(\cdot)$:

$$m=h(M).$$

Якщо підтверджується рівність обчислених значень:

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

то отримувач визнає пару (M,S) автентичною. Доведено, що тільки власник секретного

ключа D може генерувати цифровий підпис S для документу M, а визначити секретне число D по відкритому ключу E не легше, ніж розкласти модуль N на множники.

Крім того, можна строго математично довести, що результат перевірки цифрового підпису S буде позитивним тільки в тому випадку, якщо при обчисленні S було використано секретний ключ D, який відповідає відкритому ключу E. Тому відкритий ключ E іноді називають „ідентифікатором“ підписуючого.

На жаль, алгоритм електронного цифрового підпису RSA має ряд недоліків [v]:

1. При обчисленні модуля N, ключів E та D для системи цифрового підпису RSA необхідно перевірити більшу кількість додаткових умов, що досить складно реалізувати практично. Невиконання будь-якої з цих умов робить можливим фальсифікацію цифрового підпису зі сторони того, хто виявить таке невиконання. При підписуванні важливих документів не можна допускати таку можливість навіть теоретично.
2. Для забезпечення криптостійкості цифрового підпису RSA по відношенню до спроб фальсифікації на рівні, наприклад, національного стандарту США на шифрування інформації (10^{18}), необхідно використовувати при обчисленнях N, D та E цілі числа не менше 2^{512} (біля 10^{154}) кожне, що вимагає великих обчислювальних затрат, що перевищують на 20-30% обчислювальні затрати інших алгоритмів цифрового підпису при збереженні того ж рівня криптостійкості.
3. Цифровий підпис RSA вразливий до так званої мультиплікативної атаки. Інакше кажучи, алгоритм цифрового підпису RSA дозволяє зловмиснику без знання секретного ключа D сформувати підписи під тими документами, у яких результат хешування можна обчислити як добуток результатів хешування уже підписаних документів.

Алгоритм цифрового підпису Ель Гамаля (EGSA). Більш надійний та зручний для реалізації на персональних комп’ютерах алгоритм цифрового підпису був розроблений в 1984 році американцем арабського походження Тахером Ель Гамалем. В 1991 році цей алгоритм був прийнятий за основу для національного стандарту США [**Ошибка! Закладка не определена.**].

Назва EGSA походить від слів El Gamal Signature Algorithm (алгоритм цифрового підпису Ель Гамаля). Ідея EGSA базується на тому, що для обгрунтування практичної неможливості фальсифікації цифрового підпису може бути використана більш складна обчислювальна задача ніж розкладання на множники великого цілого числа, – задача дискретного логарифмування. Крім того, Ель Гамалю вдалося уникнути явної слабкості алгоритму цифрового підпису RSA, пов’язаної з можливістю підробки цифрового підпису під деякими повідомленнями без визначення секретного ключа.

Розглянемо детальніше алгоритм цифрового підпису Ель Гамаля. Для генерації пари ключів (відкритий ключ – секретний ключ), спочатку вибирають деяке велике просте число P та велике просте число G, причому $G<P$. Відправник та отримувач підписаного документу використовують при обчисленнях однакові великі цілі числа P (~10^{308} або ~2^{1024}) та G (~10^{154} або ~2^{512}), які не є секретними. Відправник вибирає випадкове ціле число X, $1<X\leq(P-1)$, та обчислює

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

Число Y є відкритим ключем, який використовується для перевірки підпису відправника. Число Y відкрито передається усім потенційним отримувачам документів. Число X є секретним ключем відправника для підписування документів та повинно зберігатися в секреті.

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

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

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

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

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

$$m=X*a+K*b(\bmod(P\text{-}1).$$

пара чисел (a,b) утворює цифровий підпис S:

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

який проставляється під документом M.

Трійка чисел (M,a,b) передається отримувачу, тоді як пара чисел (X,K) тримається у секреті. Після прийому підписаного повідомлення (M,a,b) отримувач повинен перевірити, чи відповідає підпис $S=(a,b)$ повідомленню M. Для цього отримувач спочатку обчислює за прийнятим повідомленням M число

$$m=h(M),$$

(хешує прийняте повідомлення M). Потім отримувач обчислює значення

$$A=Y^aA^b(\bmod P)$$

та визнає повідомлення M автентичним, якщо, і тільки якщо

$$A=G^m(\bmod P).$$

Іншими словами, отримувач перевіряє справедливість співвідношення

$$Y^aA^b(\bmod P)=G^m(\bmod P).$$

Можна строго математично довести, що остання рівність буде виконуватися тоді, і тільки тоді, коли підпис $S=(a,b)$ під документом M отриманий за допомогою саме того секретного ключа X, з якого було отримано відкритий ключ Y. Таким чином, можна надійно впевнитися, що відправником повідомлення M був власник саме даного секретного ключа X, не розкриваючи при цьому сам ключ, і те, що відправник підписав саме цей конкретний документ M.

Слід відмітити, що виконання кожного підпису за методом Ель Гамаля вимагає нового значення K, причому це значення повинно вибиратися випадковим чином. Якщо зловмисник колись розкриє значення K, яке повторно використовується відправником, то він зможе розкрити секретний ключ X відправника. Також слід відмітити, що схема Ель Гамаля є характерним прикладом підходу, який допускає пересилку повідомлення M у відкритій формі разом з приєднаним автентифікатором (a,b). В таких випадках процедура встановлення автентичності прийнятого повідомлення полягає в перевірці відповідності автентифікатора повідомленню.

Схема цифрового підпису Ель Гамаля має ряд переваг порівняно зі схемою цифрового підпису RSA [v]:

1. При заданому рівні стійкості алгоритму цифрового підпису цілі числа, що приймають участь в обчисленнях, мають на 25% коротший запис, що зменшує складність обчислень майже у два рази і дозволяє значно скоротити об’єм використовваної пам’яті.
2. При виборі модуля P досить перевірити, що це число є простим і що у числа $(P\text{-}1)$ є великий простий множник (всього дві умови, які легко перевіряються).
3. Процедура формування підпису за схемою Ель Гамаля не дозволяє обчислити цифрові підписи під новими повідомленнями без знання секретного ключа (як у RSA).

Проте алгоритм цифрового підпису Ель Гамаля має і деякі недоліки порівняно зі схемою підпису RSA. Зокрема, довжина цифрового підпису отримується в 1,5 рази більшою,

що, у свою чергу, збільшує час її обчислення.

Алгоритм цифрового підпису DSA. Алгоритм цифрового підпису DSA (Digital Signature Algorithm) запропонований в 1991 році у США для використання у стандарті цифрового підпису DSS (Digital Signature Standard) [vi]. Алгоритм DSA є розвитком алгоритмів цифрового підпису Ель Гамаля та К. Шнорра.

Відправник та отримувач електронного документу використовують при обчисленні великі цілі числа: G та P – прості числа, L біт кожне ($512 \le L \le 1024$); q – просте число, довжиною 160 біт (дільник числа (P-1)). Числа G, P, q є відкритими і можуть бути спільними для всіх користувачів мережі. Відправник вибирає випадкове ціле число X, $1<X<q$. Число X є секретним ключем відправника для формування електронного цифрового підпису. Потім відправник обчислює значення

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

Число Y є відкритим ключем для перевірки підпису відправника. Число Y передається усім отримувачам документів.

Цей алгоритм також передбачає використання односторонньої функції хешування $h(\cdot)$. У стандарті DSS визначено алгоритм безпечного хешування SHA (Secure Hash Algorithm). Для того, щоб підписати документ M, відправник хешує його в ціле хеш-значення m:

$$m=h(M),\ 1<m<q,$$

потім генерує випадкове ціле число K, $1<K<q$, та обчислює число r:

$$r=(G^K \bmod P) \bmod q.$$

Потім відправник обчислює за допомогою секретного ключа X ціле число s:

$$s = \frac{m + r * X}{K} \bmod q.$$

Пара чисел r та s утворює цифровий підпис

$$S=(r,s)$$

під документом M. Таким чином, підписане повідомлення представляє собою трійку чисел (M,r,s).

Отримувач підписаного повідомлення (M,r,s) перевіряє виконання умов

$$0<r<q,\ 0<s<q$$

і відхиляє підпис, якщо хоча б одна із цих умов не виконана. Якщо обидві умови виконуються, отримувач обчислює значення

$$w = \frac{1}{s} \bmod q,$$

хеш-значення

$$m=h(M)$$

та числа

$$u_1=(m*w) \bmod q,$$
$$u_2=(r*w) \bmod q.$$

Далі отримувач за допомогою відкритого ключа Y обчислює значення

$$v = \left(\left(G^{u_1} * Y^{u_2}\right) \bmod P\right) \bmod q$$

та перевіряє виконання умови $v=r$. Якщо ця умова виконується, тоді підпис $S=(r,s)$ під документом M визнається отримувачем справжньою.

Можна строго математично довести, що остання рівність буде виконуватися тоді, і тільки тоді, коли підпис $S=(r,s)$ під документом M отримано за допомогою саме того секретного ключа X, з якого було отримано відкритий ключ Y. Таким чином можна надійно впевнитись, що відправник повідомлення володіє саме даним секретним ключем X (не

розкриваючи при цьому значення ключа X) і що відправник підписав саме даний документ M.

Порівняно з алгоритмом цифрового підпису Ель Гамаля алгоритм DSA має ряд переваг [iv]:

1. При довільному допустимому рівні стійкості (при будь-якій парі чисел G і P (від 512 до 1024 біт)), числа q, X, r, s мають довжину по 160 біт, скорочуючи довжину підпису до 320 біт.
2. Більшість операцій з числами K, r, s, X при обчисленні підпису здійснюється по модулю числа q, довжиною 160 біт, що скорочує час обчислення підпису.
3. При перевірці підпису більшість операцій з числами u_1, u_2, v, w також здійснюється за модулем числа q, довжиною 160 біт, що скорочує об'єм пам'яті та час обчислення.

Недоліком алгоритму DSA є те, що при підписуванні та при перевірці підпису потрібно виконувати складні операції ділення по модулю q:

$$s = \frac{m + r * X}{K} \bmod q,\ w = \frac{1}{s} \bmod q,$$

що не дозволяє добитися максимальної швидкодії.

Слід відмітити, що реальне виконання алгоритму DSA може бути прискорене за допомогою виконання попередніх обчислень. Зазначимо, що значення r не залежить від повідомлення M та його хеш-значення m. Тому можна заздалегідь створити набір випадкових значень K і для кожного з них обчислити значення r. Можна також заздалегідь обчислити обернені значення K^{-1} для кожного із значень K. Потім при надходженні повідомлення M, можна обчислити значення s для даних значень r та K^{-1}. Такі попередні обчислення значно прискорюють роботу алгоритму DSA.

Цифрові підписи з додатковими функціональними можливостями. Розглянуті в цьому пункті цифрові підписи володіють додатковими функціональними властивостями, крім звичайних властивостей автентифікації повідомлення та неможливості відмови особи, яка підписала повідомлення від обов'язків, пов'язаних із підписаним документом. У більшості випадків вони об'єднують базову схему цифрового підпису, наприклад на основі алгоритму RSA, зі спеціальним протоколом, який забезпечує досягнення тих додаткових властивостей, якими базова схема цифрового підпису не володіє. До схем цифрового підпису з додатковими функціональними властивостями відносяться [iv]:

- схеми сліпого (blind) підпису;
- схеми незаперечного (undeniable) підпису.

Схеми сліпого підпису. На відміну від звичайних схем цифрового підпису, схеми сліпого підпису (іноді називають схемами підпису всліпу) є двохсторонніми протоколами між відправником A та стороною B, що підписує документ. Розглянемо головну ідею цих схем. Відправник A посилає порцію інформації стороні B, яку B підписує і повертає A. Використовуючи отриманий підпис, сторона A може вирахувати підпис сторони B на важливому для себе повідомленні m. По завершенні цього протоколу сторона B нічого не знає ні про повідомлення m, ні про підпис під цим повідомленням. Мета сліпого підпису полягає в тому, щоб перешкодити підписуючій особі B ознайомитися з повідомленням сторони A, яке вона підписує, і з відповідним підписом під цим повідомленням. Тому в подальшому підписане повідомлення неможливо зв'язати зі стороною A.

Розглянемо приклад використання сліпого підпису. Схема сліпого підпису може знайти застосування в тих випадках, коли відправник A (клієнт банку) не хоче, щоб підписуюча сторона B (банк) мала можливість в подальшому зв'язати повідомлення m та підпис $s_B(m)$ з певним кроком виконаного раніше протоколу. Зокрема це може бути важливим

для організації анонімних безготівкових розрахунків, коли повідомлення m могло б представляти грошову суму, яку A хоче витратити. Коли повідомлення m з підписом $s_B(m)$ надається банку B для оплати, банк B не може прослідкувати, хто саме із клієнтів надає підписаний документ. Це дозволяє користувачу A залишитися анонімним. Цей метод використовується при організації системи анонімних безготівкових розрахунків із застосуванням „електронних грошей“.

Для побудови протоколу сліпого підпису необхідні наступні компоненти [iii]:

1. Механізм звичайного цифрового підпису для підписуючої сторони B. Нехай $s_B(X)$ позначає підпис сторони B на документі X.

2. Функції $f(\cdot)$ та $g(\cdot)$ (відомі тільки відправнику) такі, що

$$g(s_B(f(m)))=s_m(m),$$

де $f(\cdot)$ – маскуюча (blinding) функція; $g(\cdot)$ – демаскуюча (unblinding) функція; $f(m)$ – замасковане (blinded) повідомлення m.

При виборі s_B, f та g існує ряд обмежень.

Виберемо в якості алгоритму підпису s_B для сторони B схему цифрового підпису RSA з відкритим ключем (N,E) та секретним ключем D, причому $N=P*Q$ – добуток двох великих випадкових простих чисел. Нехай k – деяке фіксоване ціле число, взаємно просте з N (НСД(N,k)=1). Маскуюча функція $f:Z_n\rightarrow Z_n$ визначається як $f(m)=m*k^E \bmod N$, а демаскуюча функція $g:Z_n\rightarrow Z_n$ – як $g(m)=k^{-1}m \bmod N$. При такому виборі f,g та s отримуємо

$$g(s_B(f(m)))=g(s_B(mk^E \bmod N))=g(m^Dk \bmod N=m^D \bmod N=s_B(m),$$

що відповідає наведеній вище вимозі 2.

Згідно з протоколом сліпого підпису, який запропонував Д. Чом [iii], відправник A спочатку отримує підпис сторони B на замаскованому повідомленні m. Використовуючи цей підпис, сторона A обчислює підпис B на заздалегідь вибраному повідомленні m, де $0\le m\le N$-1. При цьому стороні B нічого не відомо ні про значення m, ні про підпис, зв’язаний з m. Нехай сторона B має для підпису за схемою RSA відкритий ключ (N,E) та секретний ключ D. Нехай k – випадкове секретне ціле число, вибране стороною A і таке, що задовольняє умовам $0\le k\le N$-1 та НСД(N,k)=1. Протокол сліпого підпису Д. Чома включає наступні кроки:

1. Відправник A обчислює замасковане повідомлення $m*=mk^E \bmod N$ та відправляє його стороні B.

2. Підписуюча сторона B обчислює підпис $s*=(m*)D \bmod N$ та відправляє цей підпис стороні A.

3. Сторона A обчислює підпис $s=k^{-1}s* \bmod N$, який є підписом B на повідомленні m.

Нескладно побачити, що

$$(m*)^D\equiv(mk^E)^D\equiv m^Dk \pmod N,$$

тому

$$k^{-1}s*\equiv m^Dkk^{-1}\equiv m^D \pmod N.$$

Д. Чом розробив декілька алгоритмів сліпого незаперечуваного підпису для створення системи анонімних безготівкових електронних розрахунків eCash.

Системи незаперечного підпису. Незаперечуваний підпис, як і звичайний цифровий підпис, залежить від підписаного документу та секретного ключа. Проте, на відміну від звичайних цифрових підписів, незаперечуваний підпис не може бути верифікований без участі особи, що поставила цей підпис. Розглянемо два можливі сценарії застосування незаперечуваного підпису.

Сценарій 1. Сторона A (клієнт) хоче отримати доступ до захищеної зони, яка контролюється стороною B (банком). Цією захищеною зоною може бути, наприклад,

депозитарій (сховище цінностей клієнтів). Сторона *B* вимагає від *A* поставити до надання клієнту доступу на заявці про допуск у захищену зону підпис, час та дату. Якщо *A* застосує незаперечуваний підпис, тоді сторона *B* не зможе у майбітньому довести будь-кому, що *A* отримав допуск без безпосередньої участі *A* у процесі верифікації підпису.

Сценарій 2. Нехай відома корпорація *A* розробила пакет програмного забезпечення. Для гарантування автентичності пакету та відсутності у ньому вірусів, сторона *A* підписує цей пакет незаперечуваним підписом та продає його стороні *B*. Сторона *B* вирішує зробити копії цього пакету програмного забезпечення і перепродати його третій стороні *C*. При використання стороною *A* незаперечуваного підпису сторона *C* не зможе впевнитися в автентичності цього пакету програмного забезпечення та відсутності в ньому вірусів без участі сторони *A*. Цей сценарій, звичайно, не перешкоджає стороні *B* поставити на пакеті свій підпис, але тоді для сторони *B* будуть втрачені усі маркетингові переваги, пов'язані з використанням торгової марки корпорації *A*. Крім того буде легше розкрити шахрайську діяльність сторони *B*.

Розглянемо алгоритм незаперечуваного цифрового підпису, розроблений Д. Чомом [iv]. Спочатку опишемо алгоритм генерації ключів, за допомогою якого кожна сторона *A*, що підписує документ, вибирає секретний ключ та відповідний йому відкритий ключ. Кожна сторона *A* повинна виконати наступне:

1. Вибрати випадкове просте число $p=2q+1$, де q – також просте число.
2. Вибрати генераторне число α для підгрупи порядку q в циклічній групі Z_p^*:

2.1. Вибрати випадковий елемент $\beta \in Z_p$ та обчислити $\alpha=\beta^{(p-1)/q} \bmod p$.

2.2. Якщо $\alpha=1$, тоді повернутися до кроку 2.1.

3. Вибрати випадкове ціле $x \in \{1,2,...,q-1\}$ і обчислити $y=\alpha^x \bmod p$.
4. Для сторони *A* відкритий ключ рівний (p,α,y), секретний ключ рівний x.

Згідно алгоритму незаперечуваного підпису Д. Чома, сторона *A* підписує повідомлення *m*, що належить підгрупі порядку q в Z_p^*. Будь-яка сторона *B* може перевірити цей підпис за участі *A*.

В роботі алгоритму незаперечуваного підпису можна виділити два етапи:

– генерація підпису;
– верифікація підпису.

На етапі генерації підпису сторона *A* обчислює $s=m^x \bmod p$, де *s* – підпис сторони *A* на повідомленні *m*. Повідомлення *m* з підписом *s* відсилається стороні *B*. Етап верифікації підпису виконується стороною *B* за участі сторони *A* та включає наступні кроки:

1. *B* отримує справжній ключ (p,α,y) сторони *A*.
2. *B* вибирає два випадкових секретних цілих числа $a, b \in \{1,2,...,q-1\}$.
3. *B* обчислює $z=s^a y^b \bmod p$ та відправляє значення *z* стороні *A*.
4. *A* обчислює $w=(z)^{1/x} \bmod p$, де $xx^{-1} \equiv 1 (\bmod q)$, і відправляє значення *w* стороні *B*.
5. *B* обчислює $w'=m^a \alpha^b \bmod p$ і визнає підпис *s* справжнім, якщо і тільки якщо $w=w'$.

Впевнимося, що перевірка підпису *s* працює:

$$w \equiv (z)^{1/x} \equiv (s^a y^b)^{1/x} \equiv (m^{xa} \alpha^{xb})^{1/x} \equiv m^a \alpha^b \equiv w' \bmod p.$$

Можна довести, що з високою ступінню ймовірності зловмисник не зможе заставити *B* прийняти фальшивий підпис. Припустимо, що *s* являє собою підроблений підпис сторони *A* на повідомленні *m* ($s \neq m^x \bmod p$). Тоді ймовірність прийняття цього підпису в даному алгоритмі складає тільки $1/q$, причому ця ймовірність не залежить від обчислювальних ресурсів зловмисника.

Сторона *A*, яка підписала повідомлення, за деяких обставин могла б відмовитися від

свого справжнього підпису одним з трьох способів:

1) відмовитися від участі у протоколі верифікації;

2) некоректно виконати протокол верифікації;

3) оголосити підпис фальшивим, навіть якщо протокол верифікації виявився успішним.

Відмова від підпису першим способом розглядалася б як очевидна спроба неправомірної відмови. Проти другого та третього способів боротися складніше – тут потрібен спеціальний протокол дезавуювання. Цей протокол визначає, чи пробує сторона *A* дезавуювати правильний підпис *s* чи цей підпис є фальшивим. В цьому протоколі по суті двічі застосовується протокол верифікації і потім здійснюється перевірка з метою впевнитися, що сторона *A* виконує цей протокол коректно.

Протокол дезавуювання для схеми незаперечуваного підпису Д. Чома включає такі кроки:

1. *B* приймає від сторони *A* повідомлення *m* з підписом *s* та отримує справжній відкритий ключ (p,α,y) сторони *A*.

2. *B* вибирає випадкові секретні цілі числа $a,\ b \in\{1,2,...,q\text{-}1\}$, обчислює $z=s^a y^b \bmod p$ та відправляє значення *z* стороні *A*.

3. *A* обчислює , $w=(z)^{1/x} \bmod p$, де $xx^{-1}\equiv 1(\bmod q)$, і відправляє значення *w* стороні *B*.

4. Якщо $w=m^a \alpha^b \bmod p$, тоді *B* визнає підпис *s* справжнім і виконання протоколу зупиняється.

5. *B* вибирає випадкові секретні цілі числа $a',\ b' \in\{1,2,...,q\text{-}1\}$, обчислює $z'=s^{a'} y^{b'} \bmod p$ та відправляє значення *z'* стороні *A*.

6. *A* обчислює $w'=(z')^{1/x} \bmod p$ та відправляє значення *w'* стороні *B*.

7. Якщо $w'= s^{a'} y^{b'} \bmod p$, тоді *B* приймає підпис *s* і виконання протоколу зупиняється.

8. *B* обчислює $c=(w\alpha^{-b})^{a'} \bmod p$, $c'=(w'\alpha^{-b'})^{a} \bmod p$. Якщо $c=c'$, тоді *B* впевнюється, що підпис *s* є фальшивим; в іншому випадку *B* робить висновок, що підпис *s* справжній, а сторона *A* робить спробу дезавуювати підпис *s*.

Нескладно впевнитися в тому, що цей протокол досягає поставленої мети. Нехай *m* – повідомлення, і нехай *s* – підпис сторони *A* під повідомленням *m*. Якщо підпис *s* фальшивий ($s\neq m^x \bmod p$), і якщо сторони *A* та *B* правильно слідують протоколу, тоді $w=w'$ (і тому справедливий висновок *B*, що підпис *s* є фальшивим). Нехай *s* насправді є підписом сторони *A* під повідомленням *m* ($s=m^x \bmod p$). Нехай *B* точно слідує протоколу, а *A* не слідує. Тоді ймовірність того, що $w=w'$ (і *A* вдалося дезавуювати підпис), складає тільки $1/q$.

Слід відмітити, що третя сторона *C* ніколи не повинна приймати в якості доказу справжності підпису *s* запис стороною *B* протоколу верифікації, оскільки сторона *B* може фальсифікувати успішний запис кроку 2 і наступних кроків протоколу верифікації без участі підписуючої сторони *A*. Незаперечуваний підпис може бути верифікований тільки шляхом безпосередньої взаємодії з підписуючою стороною *A*.

Також розроблено алгоритм для оборотного незаперечуваного підпису, який може бути верифікований, дезавуйований, а також перетворений у звичайний цифровий підпис. Цей алгоритм базується на використанні алгоритму цифрового підпису Ель Гамаля.

Тема 11: **Технічний захист інформації. Основні терміни та визначення**

Технічний захист інформації - вид захисту інформації, спрямований на забезпечення за допомогою інженерно-технічних заходів та/або програмних і технічних засобів унеможливлення витоку, знищення та блокування інформації, порушення цілісності та режиму доступу до інформації.

1 Завдання у сфері захисту інформації

До основних *завдань у сфері захисту інформації (ЗІ)* в інформаційно-телекомунікаційних системах у цілому належать:

- керування доступом користувачів до інформаційних ресурсів систем з метою захисту від неправомірного випадкового або навмисного втручання у роботу і несанкціонованого (із перевищенням наданих повноважень) доступу до програм-них і апаратних ресурсів як персоналу, так і сторонніх осіб;

захист даних, що передаються каналами зв'язку;

захист інформації з обмеженим доступом від витоку;

захист інформації від спеціальних впливів;

реєстрація, збереження і надання даних про події, що відбувалися у системі і стосувалися *інформаційної безпеки (ІБ)*;

контроль роботи користувачів системи адміністраторами та обов'язкове повідомлення адміністратора безпеки про будь-які спроби несанкціонованого доступу до ресурсів системи;

контроль і підтримка цілісності критичних ресурсів *системи захисту* і середовища виконання прикладних програм;

-забезпечення функціонування програмно-технічних комплексів з метою *захисту інформації* від впровадження у роботу потенційно небезпечних програм і засобів подолання *системи захисту*;

керування та моніторинг засобів *захисту інформації.*

2 Концепція технічного захисту інформації в Україні

Вищезазначені завдання у сфері *захисту інформації* та *інформаційної безпеки* покладені в основу Концепції технічного захисту інформації в Україні, яка є складовою забезпечення національної безпеки України.

Концепція технічного захисту інформації визначає основи державної політики у сфері захисту інформації інженерно-технічними заходами. *Технічний захист інформації в Україні* – це діяльність, що спрямована на забезпечення інженерно-технічними заходами порядку доступу, цілісності та доступності (унеможливлення блокування) інформації з обмеженим доступом, а також цілісності та доступності відкритої інформації, важливої для особи, суспільства і держави. Кон-цепція технічного захисту інформації також визначає загрози безпеці інформації, стан її технічного захисту та систему технічного захисту інформації як сукупність суб'єктів, об'єднаних цілями та певними завданнями. У *Концепції технічного захисту інформації* зазначені принципи формування і проведення державної політики у цій сфері, описані основні функції організаційних структур.

Конкретні завдання у сфері захисту інформації, що розробля-ються на рівні певної організації, системи публічного управління чи держави в цілому, мають підпорядковуватися заздалегідь визначеній стратегії у цій сфері. Стратегія *захисту інформації* є основою для побудови комплексу заходів щодо *інформаційної безпеки*. Суть організації стратегії захисту інформації визначається як пошук опти-мального компромісу

між необхідністю використання конкретних засобів захисту і наявними ресурсами для реалізації цього захисту. У стратегії захисту інформації визначаються мета, критерії, принцип і процедури, необхідні для побудови надійної системи7. Таким чином, важливою особливістю загальної стратегії захисту інформації є про-ектування системи інформаційної безпеки, де визначаються напрями:

визначення мети та задач захисту інформації, цільової установки об'єктів та суб'єктів захисту інформації; визначення загроз інформаційної безпеки; визначення рівнів протидії загрозам та побудова політики інформаційної безпеки.

Тема 12: **Захист інформації в оптоволоконних мережах**

Оптоволоконні мережі забезпечують надійну передачу даних на велику відстань за рахунок своєї індивідуальної будови.

Так, протягом всієї магістралі встановлені підсилювачі сигналу, які забезпечують високу швидкість та відстань передачі сигналу. Але саме ці передавачі та оптоволокно можуть стати причинами витоку інформації.

Одним з методів захисту є покращення властивостей самого кабелю.

Так наприклад можна застосувати багато шаровий оптоволоконний кабель. По серцевині цього кабелю проходить інформаційний сигнал, а по його обмотці контролюючий сигнал. Це дозволяє знизити рівень випромінювання інформаційного сигналу, а також контролювати цілісність кабелю по всій його довжині.

При нормальній роботі магістралі, зв'язок між інформаційним та контрольним сигналами відсутній. При намаганні пошкодити захисний шар оптоволокна, ці сигнали змішуються, або падає рівень сигналу зашумлення, що фіксується на приймальній стороні.

Іншим методом захисту від несанкціонованого доступу до інформації що проходить через оптоволокно, є діагностика інформації самої магістралі. На прийомному кінці аналізується отриманий сигнал. Якщо цей сигнал приходить зі змінами, то в магістраль посилається зондуючий сигнал, який може точно вказати на місце підключення до мережі.

Також для захисту оптоволоконного кабелю можна використовувати криптографічний захист. Цей захист полягає в кодовому зашумленні сигналу при передачі. При зниженні оптичної потужності на виході оптоволоконного кабелю підвищується коефіцієнт помилок, що фіксується обладнанням контролю оптоволоконних мереж.

Список використаних джерел

1. Манько А., Каток В., Задорожний М.. Защита информации на волоконно-оптических линиях связи от несанкционированного доступа.

2. Кашаганова Г.Б., Омарова Г.А. Физические принципы каналов утечки информации в ВОЛС.

ПЕРЕЛІК ПОСИЛАНЬ

1 Кормич Б.А. Інформаційна безпека: організаційно-правові основи: Навч. посіб. – К.: Кондор, 2004. – 384 с.

2 Закон України „Про Національну програму інформатизації” від 04.02.1998 р. № 74/

3 Рубан В.Я. Інформаційна безпека України: сутність та проблеми // Стратегічна панорама. – 1998. – № 3-4. – С. 170.

4 Організація і сучасні методи захисту інформації; За ред. С.А. Дієва та А.Г. Шаваєва – М., 1998. – 52 с.

5 Закон України „Про телекомунікації” від 18.11.03 р. № 1280-IV. – Режим доступу: -

1. Закон України „Про захист інформації в автоматизованих системах” від 5.06.1994 р. № 81/94-ВР. – Режим доступу: http://www.dstszi.gov.ua

2. Правила забезпечення захисту інформації в інформаційних, телекомунікаційних та інформаційно-телекомунікаційних системах», затверджено постановою Кабінету Міністрів України від 29 березня 2006 р. № 373. – С 12.

3. НД ТЗІ 1.1-001-99. Технічний захист інформації на програмно-керованих АТС загального користування. Основні положення. – Режим доступу:

4. НД ТЗІ 2.5-001-99. Технічний захист інформації на програмно-керованих АТС загального користування. Специфікації функціональних послуг захисту.

5. НД ТЗІ 2.5-002-99. Технічний захист інформації на програмно-керованих АТС загального користування. Специфікації гарантій захисту.

6. НД ТЗІ 2.5-003-99. Технічний захист інформації на програмно-керованих АТС загального користування. Специфікації довірчих оцінок коректності реалізації захисту.

7. НД ТЗІ 2.7-001-99. Технічний захист інформації на програмно-керованих АТС загального користування. Порядок виконання робіт. – Режим доступу:

8. НД ТЗІ 3.7-002-99. Технічний захист інформації на програмно-керованих АТС загального користування. Методика оцінювання захищеності інформації (базова).

9. НД ТЗІ 2.1-001-2001. Створення комплексів технічного захисту інформації. Атестація комплексів. Основні положення.

10. КНД 45-164-2001. Типова модель загроз для формування ресурсів цифрових АТС, що використовуються в мережах електрозв’язку загального користування України. – Режим доступу: http://www.stc.gov.ua

11. ISO/IEC 15408:2000 Части 1 – 3. “Информационные технологии. Общие критерии оценки безопасности информационных технологий (ИТ – безопасности)”, ISO/IEC 13335:1997 «Руководство по управлению ИТ – безопасностью», ИСО/МЭК 17799:2000 “Практические рекомендации по управлению ИТ-безопасностью”.

12. НД ТЗІ 3.7-003-03. Порядок проведення робіт із створення комплексної системи захисту інформації в інформаційно-телекомунікаційній системі.

13. НД ТЗІ 1.4-001-2000. Типове положення про службу захисту інформації в автоматизованій системі.

14. Домарев В.В. Математические модели систем и процессов защиты информации.Електронний ресурс http://www.domarev.kiev.ua/nauka/. С.56.

15. Кравченко В.П. Система поддержки принятия решений. Електронный ресурс: http://it2b.ru/it2b2.viev5.page21.html.

16. Черноруцкий И.Г. Методы оптимизации и принятия решений. С.-Пб, 2001, С. 248.
17. Организация планирование и управление предприятиями связи / Е.В. Демина, Е.К. Иодко, Л.И. Майофис, Н.П. Резникова. – М.: Радио и связь, 1990. – 352 с.
18. Котенко М. Центры обработки вызовов // Телеком № 9-10 (27-28)/2000. С.56-63.
19. Гольдштейн Б.С., Зарубин А.А., Потапов А.И. Центры обработки вызовов для органов внутренних дел: учебное пособие // СПбГУТ. – СПб. – 2005. – 52 с.
20. Петренко С. Методические основы защиты информационных активов компании // Режим доступу: – www.infosecurity.ru/ gazeta/content/031104/article03.html.
21. Шварц М. Сети ЭВМ. Анализ и проектирование / Пер. с англ. Под ред. В. А.Жожикашвили – М.: Радио и связь, 1981. – 336 с.
22. Корнышев Ю. Н., Мамонтова Н. П. Задачник по теории телефонных и телеграфных сообщений. − Одесса: ОЭИС, 1974. 139 с.
23. Корявцев П.М. Общий комплекс мер по обеспечению инфрмационной безопасности. Методические рекомендации // Вестник МВД РФ. – М.: 2000.
24. НД ТЗІ 2.5-005-99. Класифікація автоматизованих систем і стандартні функціональні профілі захищеності оброблюваної інформації від несанкціонованого доступу. Режим доступу: http://www.dsszzi.gov.ua/
25. НД ТЗІ 2.5-010-03. Вимоги до захисту інформації WEB-сторінки від несанкціонованого доступу. – Режим доступу: http://www.dsszzi.gov.ua/
26. Бельфер Р. А. Классификация угроз информационной безопасности сетей связи ВСС России (ISDN, IN, UMTS) и методы их количественной оценки // “Электросвязь”, М: – 2002, № 7. – С.14 – 18.
27. Забезпечення інформаційної безпеки цифрових програмно керованих АТС Інформаційна безпека телефонного зв’язку: навч. посібник / [Кононович В.Г., Стайкуца С.В., Тардаскіна Т.М., Шинкарчук Т.М.] За ред. чл.-кор. МАЗ В.Г. Кононовича. – Одеса: ОНАЗ ім. О.С. Попова, 2010. – С. 168.
28. Б
а 29. 34 Венбо Мао. Современная криптография: теория и практика. – М.: р Вильямс, 2005. – 768 с.
а 30. ДСущевский. Г., Панченко О. В., Кугураков В. Н. СОВРЕМЕННЫЕ к КРИПТОСИСТЕМЫ И ИХ ОСОБЕННОСТИ //Вестник технологического университета. 2015. Т.18, №11 с.194-197
к 31. https://sites.google.com/site/anisimovkhv/learning/kripto/rabprog
32. https://sites.google.com/site/anisimovkhv/learning/kripto/lecture/tema3#kriptogra
фія
.
М 33. Про інформацію: Закон України від 02.10.1992 р. № 2657-XII, із змінами. – Режим доступу: zakon2.rada.gov.ua/laws/ show/2657-12. – Назва з екрану
.
С
т 34. Про захист інформації в інформаційно-телекомунікацій-них системах: Закон України від 05.07.1994 р. № 80/94-вр, із змінами. – Режим доступу: zakon5.rada.gov.ua/laws/ show/80/94-вр. – Назва з екрану.
е
ц 35. Про Положення про технічний захист інформації в Україні: Указ Президента України від 27.09.1999 р. № 1229/99. – Режим доступу: zakon3.rada.gov.ua/laws/show/1229/99. – Назва з екрану
ь
.
О 36. Про затвердження Концепції технічного захисту інформа-ції в Україні: постанова Кабінету Міністрів України від 08.10.1997 р. № 1126, із змінами. – Режим доступу: zakon3. rada.gov.ua/laws/show/1126-97-п. – Назва з екрану
.
В
І 37. .Остапов С.Е. Технології захисту інформації: навч. посіб. /С.Е. Остапов, С.П. Євсеєв, О.Г. Король. – Х.: ХНЕУ, 2013. – 476 с.
н
ф
о
р
м
а

Айерленд К., Роузен М. Классическое введение в теорию чисел/ Пер. с англ. Под. ред. С.П.Демушкина, А.Н.Паршина. – М.: Мир, 1987. – 420 с.
Схема Эль-Гамаля// Википедия. Свободная энциклопедия. – [Електронний ресурс]. – http://ru.wikipedia.org/wiki/Эль-гамаль. Заголовок з екрану. Мова рос., дата цитування: 19.12.2009 р.
Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. – М.: Изд-во Триумф, 2002. – 816 с.

V. Нильс Фергюсон, Брюс Шнайдер. Практическая криптография. – М.: Диалектика, 2004. – 432 с.

vi
D
Википедия. Свободная энциклопедия. – [Електронний ресурс]. – http://ru.wikipedia.org/wiki/DSA. Заголовок з екрану. Мова рос., дата цитування: 20.12.2009 р.