Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

Безпека інформаційних технологій

Методичні рекомендації до виконання лабораторних робіт для студентів денної форми навчання галузі 12 Інформаційні технології

ЗАТВЕРДЖЕНО
на засіданні кафедри кібербезпеки та програмного забезпечення, протокол № 1 від 15.08.2022

Кропивницький
2023

Безпека інформаційних технологій: Методичні рекомендації до виконання лабораторних робіт для студентів денної форми навчання галузі 12 Інформаційні технології. – Кропивницький: ЦНТУ – 2023. – 48 с./М-во освіти і науки України, Центральноукр. нац. техн. ун-т; /уклад. Смірнов О.А., Буравченко К.О., Смірнова Т.В., Конопліцька-Слободенюк О.К., Смірнов С.А./ – Кропивницький: ЦНТУ – 2023. – 48 с.

Укладачі: Смірнов О.А., Буравченко К.О., Смірнова Т.В., Конопліцька-Слободенюк О.К., Смірнов С.А.

Рецензенти: Коваленко О.В., докт. техн. наук, доцент;
Улічев О.С., канд. техн. наук.

ЗМІСТ

ВСТУП

Метою освітньої компоненти «Безпека інформаційних технологій» є формування у здобувачів вищої освіти ґрунтовних теоретичних знань, практичних умінь та навичок, необхідних для застосування в професійній діяльності у сфері захисту інформації в комп'ютерних системах.

Основними **завданнями** вивчення дисципліни є формування наступних **компетенцій бакалавра з комп'ютерної інженерії**:

– Р2. Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення.

– Р4. Здатність забезпечувати захист інформації, що обробляється в комп'ютерних та кіберфізичних системах та мережах з метою реалізації встановленої політики інформаційної безпеки.

– Р7. Здатність використовувати та впроваджувати нові технології, включаючи технології розумних, мобільних, зелених і безпечних обчислень, брати участь в модернізації та реконструкції комп'ютерних систем та мереж, різноманітних вбудованих і розподілених додатків, зокрема з метою підвищення їх ефективності.

– Р9. Здатність системно адмініструвати, використовувати, адаптувати та експлуатувати наявні інформаційні технології та системи.

У результаті вивчення дисципліни студент повинен забезпечити наступні **програмні результати навчання**:

Знати:

– N1. Знати і розуміти наукові положення, що лежать в основі функціонування комп'ютерних засобів, систем та мереж.

– N2. Мати навички проведення експериментів, збирання даних та моделювання в комп'ютерних системах.

Вміти:

– N6. Вміти застосовувати знання для ідентифікації, формулювання і розв’язування технічних задач спеціальності, використовуючи методи, що є найбільш придатними для досягнення поставлених цілей.

– N7. Вміти розв’язувати задачі аналізу та синтезу засобів, характерних для спеціальності.

– N12. Вміти ефективно працювати як індивідуально, так і у складі команди.

– N16. Вміти оцінювати отримані результати та аргументовано захищати прийняті рішення.

Набути навичок комунікації:

– N17. Спілкуватись усно та письмово з професійних питань українською мовою та однією з іноземних мов (англійською, німецькою, італійською, французькою, іспанською).

– N18. Використовувати інформаційні технології та для ефективного спілкування на професійному та соціальному рівнях

Набути навичок автономії і відповідальності:

– N19. Здатність адаптуватись до нових ситуацій, обґрунтовувати, приймати та реалізовувати у межах компетенції рішення.

– N20. Усвідомлювати необхідність навчання впродовж усього життя з метою поглиблення набутих та здобуття нових фахових знань, удосконалення креативного мислення.

– N21. Якісно виконувати роботу та досягати поставленої мети з дотриманням вимог професійної етики.

У результаті вивчення навчальної дисципліни студент повинен:

– **знати:** економне кодування; префіксний код та його дерево; алгоритм Шеннона-Фано; алгоритм Хаффмана; боротьба з помилками, які виникають у каналах передачі даних; стисла характеристика методів боротьби з помилками; принципи завадостійкого кодування; основні характеристики

завадостійких кодів; класифікація завадостійких кодів; математичний опис процесу кодування і декодування; блочні лінійні коди; коректуючи властивості блочних кодів; коди з перевіркою на парність; коди Хеммінга; циклічні коди; засоби опису циклічних кодів; властивості циклічних кодів по виявленню помилок; укорочені циклічні коди; коди Боуза-Чоудхурі-Хоквінгема; коди Ріда-Соломона; код Файра; згорточні коди; принципи кодування і декодування; ланцюговий код; каскадні коди; методи та засоби забезпечення захист інформації, що обробляється в комп'ютерних та кіберфізичних системах та мережах; загрози, яким підлягає інформація; основні міри протидії загрозам безпеці, принципи побудови систем захисту, основні механізми захисту; криптографічні методи захисту; види засобів криптозахисту даних; переваги і недоліки; місце і роль засобів криптозахисту; симетричні та асиметричні алгоритми шифрування; сучасні симетричні алгоритми шифрування (AES), (Калина); основні поняття захисту web-ресурсів; типовий комплекс засобів захисту web-ресурсів від несанкціонованого доступу; класифікація комп'ютерних та кіберфізичних систем та мереж і стандартні функціональні профілі захищеності оброблюваної інформації від несанкціонованого доступу

– **вміти** програмно реалізовувати наступні проекти: Визначення ймовірності події; Оптимальний код Хафмана та Шеннона-Фано; Код Хеммінга; Циклічні коди та їх застосування; Шифрування та дешифрування методами Цезаря й Видженера; Шифрування та дешифрування методами методом RSA; Шифрування та дешифрування методами ДСТУ 28147:2009 та DES.

Пререквізити

Враховуючи послідовність накопичення знань та інформації, дисципліна вивчається після викладання наступних дисциплін: «Вища математика», «Алгоритми та методи обчислень», «Базові методології та технології програмування», «Бази даних», «Інженерія програмного забезпечення».

Контроль знань

Критерії оцінки іспиту:

оцінку «відмінно» (90-100 балів, А) – заслуговує студент, який:

– всебічно, систематично і глибоко володіє навчально-програмовим матеріалом;

– вміє самостійно виконувати завдання, передбачені програмою, використовує набуті знання і вміння у нестандартних ситуаціях;

– засвоїв основну і ознайомлений з додатковою літературою, яка рекомендована програмою;

– засвоїв взаємозв'язок основних понять дисципліни та усвідомлює їх значення для професії, яку він набуває;

– вільно висловлює власні думки, самостійно оцінює різноманітні життєві явища і факти, виявляючи особистісну позицію;

– самостійно визначає окремі цілі власної навчальної діяльності, виявив творчі здібності і використовує їх при вивченні навчально-програмового матеріалу, проявив нахил до наукової роботи.

оцінку « добре» (82-89 балів, В) – заслуговує студент, який:

– повністю опанував і вільно (самостійно) володіє навчально-програмовим матеріалом, в тому числі застосовує його на практиці, має системні знання достатньому обсязі відповідно до навчально-програмового матеріалу, аргументовано використовує їх у різних ситуаціях;

– має здатність до самостійного пошуку інформації, а також до аналізу, постановки і розв'язування проблем професійного спрямування;

– під час відповіді допустив деякі неточності, які самостійно виправляє, добирає переконливі аргументи на підтвердження вивченого матеріалу;

оцінку «добре» (74-81 бал, C) заслуговує студент, який:

– в загальному роботу виконав, але відповідає на екзамені з певною кількістю помилок;

– вміє порівнювати, узагальнювати, систематизувати інформацію під керівництвом викладача, в цілому самостійно застосовувати на практиці, контролювати власну діяльність;

– опанував навчально-програмовий матеріал, успішно виконав завдання, передбачені програмою, засвоїв основну літературу, яка рекомендована програмою;

оцінку «задовільно» (64-73 бали, D) – заслуговує студент, який:

– знає основний навчально-програмовий матеріал в обсязі, необхідному для подальшого навчання і використання його у майбутній професії;

– виконує завдання, але при рішенні допускає значну кількість помилок;

– ознайомлений з основною літературою, яка рекомендована програмою;

– допускає на заняттях чи екзамені помилки при виконанні завдань, але під керівництвом викладача знаходить шляхи їх усунення.

оцінку «задовільно» (60-63 бали, E) – заслуговує студент, який:

– володіє основним навчально-програмовим матеріалом в обсязі, необхідному для подальшого навчання і використання його у майбутній професії, а виконання завдань задовольняє мінімальні критерії. Знання мають репродуктивний характер.

оцінка «незадовільно» (35-59 балів, FX) – виставляється студенту, який:

– виявив суттєві прогалини в знаннях основного програмового матеріалу, допустив принципові помилки у виконанні передбачених програмою завдань.

оцінку «незадовільно» (35 балів, F) – виставляється студенту, який:

– володіє навчальним матеріалом тільки на рівні елементарного розпізнавання і відтворення окремих фактів або не володіє зовсім;

– допускає грубі помилки при виконанні завдань, передбачених програмою;

– не може продовжувати навчання і не готовий до професійної діяльності після закінчення університету без повторного вивчення даної дисципліни.

При виставленні оцінки враховуються результати навчальної роботи студента протягом семестру

Критерії оцінки заліку:

– «зараховано» – студент має стійкі знання про основні поняття дисципліни, може сформулювати взаємозв'язки між поняттями.

– «незараховано» – студент має значні пропуски в знаннях, не може сформулювати взаємозв'язку між поняттями, що вивчаються в курсі, не має уявлення про більшість основних понять дисципліни, що вивчається.

Шкала оцінювання: національна та ЄКТС

<table>
<tr><td rowspan="2">Сума балів за всі види навчальної діяльності</td><td rowspan="2">Оцінка ЄКТС</td><td colspan="2">Оцінка за національною шкалою</td></tr>
<tr><td>для екзамену, курсового проекту (роботи), практики</td><td>для заліку</td></tr>
<tr><td>90-100</td><td>A</td><td>відмінно</td><td rowspan="5">зараховано</td></tr>
<tr><td>82-89</td><td>B</td><td rowspan="2">добре</td></tr>
<tr><td>74-81</td><td>C</td></tr>
<tr><td>64-73</td><td>D</td><td rowspan="2">задовільно</td></tr>
<tr><td>60-63</td><td>E</td></tr>
<tr><td>35-59</td><td>FX</td><td>незадовільно з можливістю повторного складання</td><td>не зараховано з можливістю повторного складання</td></tr>
<tr><td>1-34</td><td>F</td><td>незадовільно з обов’язковим повторним вивченням дисципліни</td><td>не зараховано з обов’язковим повторним вивченням дисципліни</td></tr>
</table>

Лабораторна робота N1

Тема: Визначення ймовірності події

Мета: Оволодіти методикою визначення ймовірності події.

Теоретичні відомості.

Для визначення ймовірності події можливо використати наступний "частотний" метод, що ґрунтується на стійкості послідовності значень появи події. Згідно цього методу слід провести серію дослідів однієї розмірності, в кожному з яких підраховують N_i – кількість тих випадків, коли настає подія, де і-номер досліду. Накопичена частота V обчислюється за такою формулою:

$$V_n = \frac{\sum_{i=1}^{n} N_i}{n \cdot d},$$

де *d* – розмірність кожного досліду

Отримані числа записуються в таблицю:

№ досліду	1	2		……	N
N_i				…….	
V_i				…….	

Отриману послідовність { V_i } дослідимо на предмет наявності властивості стабільності. Ця властивість полягає в тому, що починаючи з номера N для всіх i>N має місце.

$$\mathbf{|V_i - P| < \varepsilon} \text{ де } \mathbf{\varepsilon = 0{,}0001}$$

Якщо ε – нескінченно мала величина, то маємо рівність.

$$\mathbf{P = \lim_{i \to \infty} V_i}$$

де і – номер серії досліду, V_i – накопичена частота появи букви після і-тої серії.

Завдання:

Побудувати таблицю ймовірностей всіх літер алфавіту та записати цю таблицю до файлу. Файл з якого беруться частота появи усіх букв, повинен бути не менш ніж 30 кБ.

Частота вживання літер українського алфавіту

А	0,072	Ї	0,006	У	0,04
Б	0,017	Й	0,008	Ф	0,001
В	0,052	К	0,035	Х	0,012
Г	0,016	Л	0,036	Ц	0,006
Д	0,035	М	0,031	Ч	0,018
Е	0,017	Н	0,065	Ш	0,012
Є	0,008	О	0,094	Щ	0,001
Ж	0,009	П	0,029	Ю	0,004
З	0,023	Р	0,047	Я	0,029
И	0,061	С	0,041	Ь	0,029
І	0,057	Т	0,055	пробіл	0,17

Лабораторна робота №2

Тема: Оптимальний код Хафмана та Шеннона-Фано.

Мета: Вивчення кодів, що стискають інформацію.

Теоретичні відомості.

Цей метод ґрунтується на ідеї кодування найкоротшим кодовим словом те повідомлення, що має найбільшу ймовірність. Ця ідея реалізується шляхом послідовного стискання початкової множини повідомлень та ілюструється наступним прикладом: для п'яти різних повідомлень із ймовірностями:

$$P_1=0.4;\ P_2=0.2;\ P_3=P_4=0.15;\ P_5=0.1$$

Номер	Ймовірність та кодові позначення				
	Вхідна множина ймовірностей	1-стискання	2-стискання	3-стискання	коди
1	0.4	0,4	0,4	0,4 0	0
2	0.2	0,2 0			100
			0,35 0		
3	0.15	0,15 1		0,6 1	101
4	0,15 0		0,25 1		110
		0,25			
5	0,1 1				111

В цій таблиці наведено три послідовні стискання вихідної множини:

Перше стискання отримане шляхом заміни двох (4+5) повідомлень одним та переписування трьох інших повідомлень.Друге стискання отримане із результату першого шляхом заміни (2 та 3) із найменшими двома ймовірностями.Так само виконується третє стискання в результаті якого

отримаємо два числа, сума яких=1. Кодуємо кожну пару "стиснутих" чисел 0 та 1 та виписуємо кодові слова, рухаючись у зворотньому напрямку (3стиск-->2стиск-->1стиск). Отримаємо колонку кодових слів. Обчислимо середню довжину кодового слова:

$$L= \sum_{i=5}^{5} L_i P_i = 1 \cdot 0.4 + 2(0.2 + 0.15 + 0.15 + 0.1) = 0.4 + 1.2 = 1.6 \text{ біт}$$

Середня довжина L вимірюється в бітах, бо слово "Біт" означає, як кодовий символ, так і одиницю виміру інформації. З іншого боку L може визначитися, як M (ξ) ,де ξ – випадкова величина, визначена на множині з п'яти повідомлень A (i-те повідомлення має ймовірність Pi,i=1,2,3..5) та має значення ξi= ξ {i}=i. Тобто L=M (ξ)."Середнє" слово містить кількість інформації, що дорівнює H (ξ), яка обчислюється за формулою

$$H(\xi) = \sum_{i=1}^{5} P_i \cdot \log_2\left(\frac{1}{P_i}\right) \quad \text{(бітів)}$$

Дробове число H (ξ) вказує на те , що деякі кодові символи несуть такі частини інформації, що мають спільну частину, тобто перекриваються за рахунок спільної частинки інформації. Виконується нерівність:

$$L \geq H(\xi),$$

яка вказує на теоретичну границю стискання інформації шляхом кодування.

Код Фано має подібну (до наведенної вище) схему стискання. Тільки використовується розбиття на дві "рівноймовірні" групи та подальше кодування цих частин 0 та 1 до того моменту коли залишиться одне число в групі.

1	0.4	0	0		00
2	0.1		1		01
3	0.2	1	0		10
4	0.15		1	0	110
5	0.15			1	111

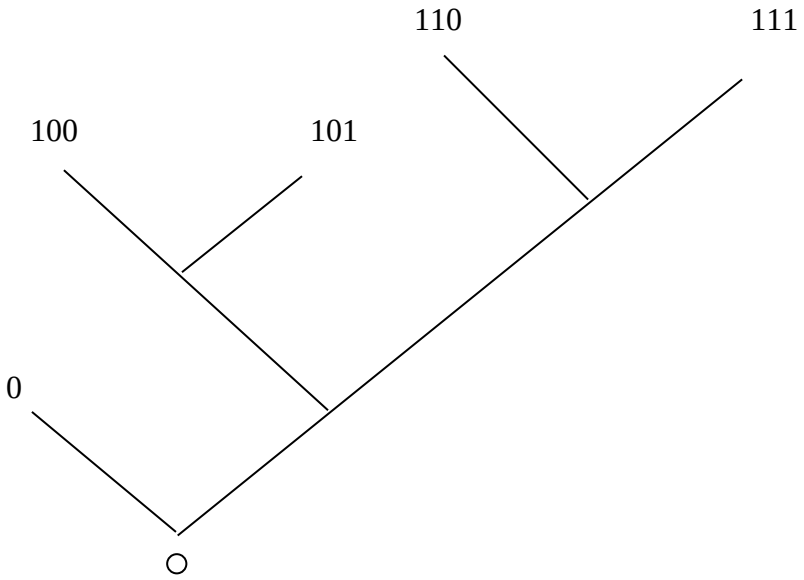
Кодові слова для кожного повідомлення отримаємо шляхом "зворотнього" виписування рядка 0 та 1.

Середня довжина L1 слова буде такою:

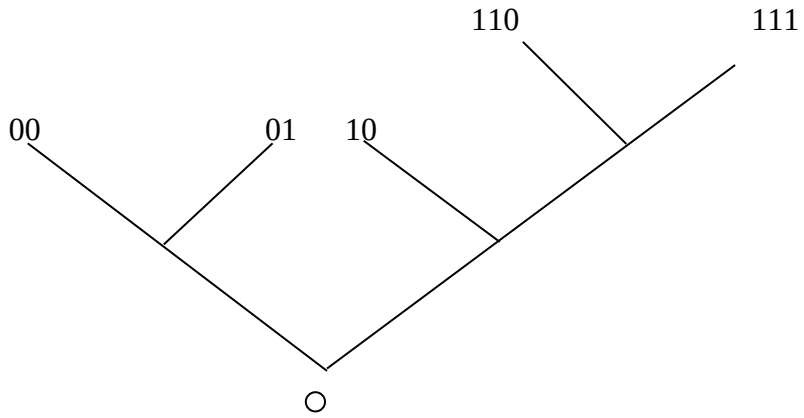
$$L_1 = 2 \cdot 0.4 + 2 \cdot 0.1 + 2 \cdot 0.2 + 3(0.15 + 0.15) = 1.4 + 0.9 = 2.3 \text{ біта}$$

Порівнюючи наведені методи маємо , що $L_1 > L$.

Дерево для оптимального коду Хафмана



Дерево для коду Фано



Практична реалізація кода Шеннона-Фано

Спочатку кожному символу інформаційного алфавіту ставиться у відповідність код нульової довжини («порожній» код) і вага, рівна ймовірності появи символу на виході інформаційного джерела в рамках обраної інформаційної моделі. Всі символи алфавіту сортуються по убуванню або зростанню їхніх ваг, після чого впорядкований ряд символів у деякому місці ділиться на дві частини так, щоб у кожній з них сума ваг символів була приблизно однакова. До коду символів, що належать однієї із частин, додається «0», а до коду символів, що належать іншої частини, додається «1» (додається значення, що, формує черговий крайній правий розряд коду). Як неважко зрозуміти, кожна із зазначених частин сама по собі є впорядкованим рядом символів. Кожний із цих рядів, якщо він містить більше одного символу, у свою чергу ділиться на дві частини відповідно до описаного вище принципом, і до коду символів знову додаються відповідні двійкові значення й т.д. Процес завершується тоді, коли у всіх отриманих у такий спосіб рядах залишається рівно по одному символу.

Як видно, алгоритм Шеннона-Фано в дійсності не будує кодове дерево, однак його роботу можна проілюструвати такою побудовою. При поділі групи символів відбувається як би розгалуження деякого вузла бінарного дерева (листовий вузол стає вузлом батьком для двох новостворених

дочірніх вузлів), а присвоєння нулів і одиниць рівносильно встановленню відповідності між кодами й маршрутами руху по дереву від кореневого вузла до листового вузла. Робота алгоритму Шеннона-Фано проілюстрована на наступному прикладі (мал. 2).

Як же ми ділили на групи? Досить просто:

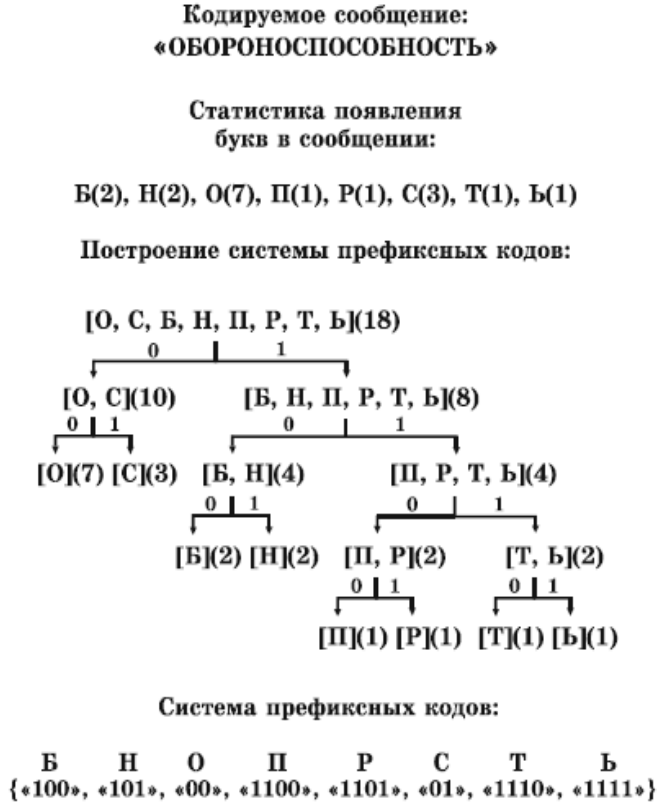
Розміщаємо символи по ймовірності появи.

1. ймовірність першої групи (p1) і другої (p2) дорівнює нулю;
2. p1 <= p2 ?

так: додати в першу групу символ з початку таблиці;

ні: додати в другу групу символ з кінця таблиці;

3. якщо всі символи розділені на групи, то завершити алгоритм, інакше перейти до кроку 2



Ілюстрація роботи алгоритму Шеннона-Фано на прикладі кириллиці.

Практична реалізація кода Хафмана

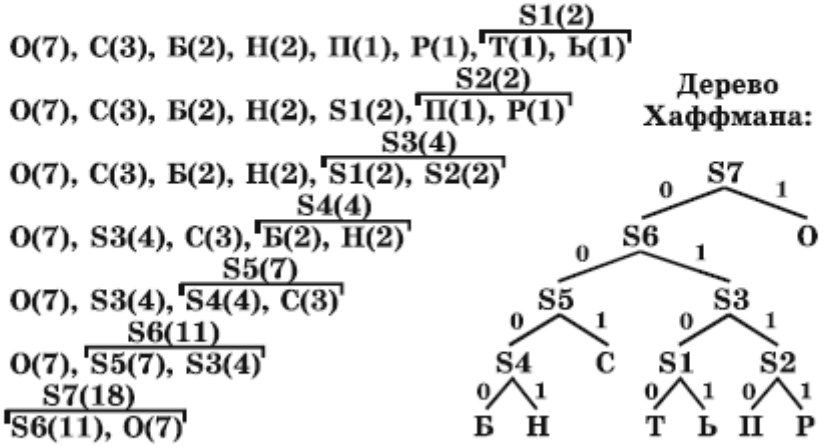
На початковому етапі роботи алгоритму кожному символу інформаційного алфавіту ставиться у відповідність вага, рівна ймовірності (частоті) появи даного символу в інформації. Символи містяться в список, що сортується по убуванню ваг. На кожному кроці (ітерації) два останніх елементи списку поєднуються в новий елемент, що потім міститься в список замість двох поєднуваних елементів. Новому елементу списку ставиться у відповідність вага, дорівнює сумі ваг елементів, що заміщаються. Кожна ітерація закінчується впорядкуванням отриманого нового списку, що завжди містить на один елемент менше, ніж старий список. Паралельно з роботою зазначеної процедури здійснюється послідовна побудова кодового дерева. На кожному кроці алгоритму будь-якому елементу списку відповідає кореневий вузол бінарного дерева, що складається з вершин, які відповідають елементам, об'єднанням яких був отриманий даний елемент. При об'єднанні двох елементів списку відбувається об'єднання відповідних дерев в одне нове бінарне дерево, у якому кореневий вузол відповідає новому елементу, що поміщається в список, а елементам списку, що заміщуються, відповідають дочірні вузли цього кореневого вузла. Алгоритм завершує роботу, коли в списку залишається один елемент, що відповідає кореневому вузлу побудованого бінарного дерева. Це дерево називається деревом Хаффмана. Система префіксних кодів може бути отримана шляхом присвоювання конкретних двійкових значень ребрам цього дерева. Приклад побудови дерева Хаффмана наведений нижче.

Кодируемое сообщение:
«ОБОРОНОСПОСОБНОСТЬ»

Статистика появленя букв в сообщении:

Б(2), Н(2), О(7), П(1), Р(1), С(3), Т(1), Ь(1)

Построение системы префиксных кодов:



Система префиксных кодов:

Б	Н	О	П	Р	С	Т	Ь
«0000»	«0001»	«1»	«0110»	«0111»	«001»	«0100»	«0101»

Ілюстрація роботи алгоритму Хаффмана

Завдання:

1. Обчислити ймовірність літер вашого прізвища, доповнити отриману множину чисел ще одним числом, так щоб сума усіх дорівнювала 1. Закодувати прізвище оптимальним кодом Хафмана та кодом Фано, побудувати їх дерева, обчислити середні довжини слів та кількість інформації в них.

2. Побудувати програми, що стискають текстову інформацію в файлі за оптимальним методом Шеннона–Фано та методом Хафмана, а потім розтискають її. Розмір файлу не менший 30 кБ.

Лабораторна робота N 3

Тема. **Код Хеммінга**

Мета: Вивчення кодів, що виправляють помилки, породжені при передачі інформаціі по каналу зв'язку. Побудувати код Хемінга, що виправляє одиничні помилки та виявляє подвійні помилки.

Теоретичні відомості.

Розглянемо задачу побудови коду, що складається із 16-ти кодових слів виду $a_1a_2a_3a_4$, де a1=0 або 1 в якому можливе виявлення та виправлення помилок.

Для розв'язку використаємо три допоміжні (перевірочні) символи, тобто кожне з 16-ти кодових слів кодуємо двійковим словом довжини 7: $a_1a_2a_3a_4a_5a_6a_7$. Нашу задачу можливо переформулювати, як визначення одного із чисел 0,1...7, які б вказували на місце похибки (0 – немає похибки). Накладемо умови (у вигляді рівностей по **mod 2**) на перевірочні символи:

$$a_5=a_2+a_3+a_4$$
$$a_6=a_1+a_3+a_4$$
$$a_7=a_1+a_2+a_4$$

Для виявлення похибки достатньо обчислити суму:

$$S_1=a_4+a_5+a_6+a_7.$$

Якщо $S_1=1$ то похибка є, інакше "немає".

При наявності похибки перевіримо, чи не міститься вона серед a_1 чи a_7. Для цього обчислимо $S_2=a_2+a_3+a_6+a_7$.

Якщо $S_1=S_2=1$ то похибка в a_6 або a_7.

Якщо $S_1=1$, $S_2=0$ то похибка в a_4 або a_5.

Якщо $S_1=0$, $S_2=1$ то похибка в a_2 або a_3.

Якщо $S_1=S_2=0$ то похибка в a_1 або немає.

Для визначення місця похибки слід обчислити суму

$$\mathbf{S_3=a_1+a_3+a_5+a_7.}$$

Тобто матимемо три перевірочні відношення

$$\mathbf{S_1=a_4+a_5+a_6+a_7 \neq 0}$$
$$\mathbf{S_2=a_3+a_2+a_6+a_7 \neq 0}$$
$$\mathbf{S_3=a_1+a_3+a_5+a_7 \neq 0}$$

які шляхом порівняння з 0(mod2) "обчислюють" або відсутність похибки, або вказують її місце. Положення одиничної похибки визначається числом $\mathbf{S_1S_2S_3}$ в двійковій системі лічби. Таким чином маємо код (4,7) Хемінга довжина слова 7 та 4 інформаційні символи.

Якщо виявляти подвійну похибку, то слід виконати перевірку:

$$\mathbf{S_0=a_0+a_1+a_2+a_3+a_4+a_5+a_6+a_7 \neq 0}$$

де $\mathbf{a_0}$ – додатковий (четвертий) перевірочний символ, значення обчислємо за формулою

$$\mathbf{a_0=a_1+a_2+a_3+a_4+a_5+a_6+a_7 \ (mod\ 2)}$$

Попередні $\mathbf{a_5, a_6, a_7}$ допомагали виявити та виправити одиночну похибку.

Якщо виникла подвійна похибка, то $\mathbf{S_0=0(mod\ 2)}$ та хоча б одне із чисел $\mathbf{S_1,S_2,S_3}$ буде не нульовим. Виправлення такої похибки неможливе. Побудована множина кодових слів зветься розширеним кодом (4,8) Хемінга (довжина слова 8 та з них інформаційними символами).

Завдання:

Скласти програму побудови коду Хемінга, його систему перевірочних відношень та його розширений варіант для кодових слів довжини 2^{m-1} (символів), серед яких присутні m перевірочних символів.

Якщо N ваш номер в журнальному списку групи, то $\mathbf{m=N(mod\ 5)+3}$.

Лабораторна робота № 4

Тема: Циклічні коди та їх застосування

Теоретичні відомості.

Циклічні коди – найцінніша частина теорії кодування завдяки ідеальній пристосованості до реалізації в сучасних технічних засобах. Але теоретичне обґрунтування алгоритмів кодування–декодування та їх простих реалізацій вимагає використання складного алгебраїчного апарату, тому познайомимося лише із результатами без їхнього обґрунтування.

Циклічним зсувом вектора $\vec{a}$, $\vec{a}=(a_0,a_1,...,a_{n-1})$ із координатами із множини F, F={0,1}, будемо називати вектор $\vec{a}'$, $\vec{a}'=(a_{n-1},a_0,a_1,...,a_{n-1})$. Наприклад для вектора (01101) послідовні зсуви мають вигляд: (10110),(01011),(10101),(11010).

Циклічним кодом зветься лінійний код, який разом із кожним своїм вектором містить також його циклічний зсув. Наприклад циклічним є код із породжуючою матрицею G, де

$$G=\begin{pmatrix}1 & 0 & 1 & 0\\ 0 & 1 & 0 & 1\end{pmatrix}.$$

При розгляді циклічних кодів використовують операцію циклічного зсуву для опису якої використовують поліноми . Для цього із кожним вектором $\vec{a}=(a_0,a_1,...,a_{n-1})$ зв'яжемо поліном a(x), де $a(x)=a_0+a_1x+...+a_{n-1}x^{n-1}$. Циклічному зсуву $\vec{a}$ вправо на 1 відповідатиме поліном $a'(x)=a_{n-1}+a_0x+a_1x^2+...+a_{n-2}x^{n-1}$. Порівнявши $a(x)$ та $a'(x)$ отримаємо, що $a'(x)=xa(x)-a_{n-1}(x^n-1)$. Будемо вважати x породжуючим для групи степенів x від 0 до n-1, причому $x^n=1$ та $x^kx^m=x^r$, де $r\equiv(k+m)(\mathrm{mod}\,n)$. Із урахуванням цього маємо наступне порівняльне співвідношення для $a(x)$ та $a'(x)$: $a'(x)=xa(x)$, тобто циклічний зсув довільного вектора маємо після множення цього вектора на x. Враховуючи

породжуючи властивості x, маємо таке правило множення двох поліномів степені $\le n-1$: для

$$a(x) = a_0 + a_1 x + ... + a_{n-1} x^{n-1},$$
$$b(x) = b_0 + b_1 x + ... + b_{x-1} x^{n-1},$$

знаходимо добуток спочатку звичайним способом: розкриваючи дужки, множимо x^k та x^m; згідно правила маємо x^n, $r = (k+m)(\mathrm{mod}\, n)$, а коефіцієнти a_k та b_m визначаємо по правилу множення на множині F, $F = \{0,1\}$, беручи добуток за модулем 2. Наприклад: n=5, $a(x) = 1 + x + x^2 + x^4, b(x) = 1 + x^3 + x^4$ двійкові поліноми. Тоді $a(x)b(x) = \langle x + x = 2x = 0 \rangle = 1 + x^4$.

Таким же чином визначимо операцію додавання поліномів. Наприклад, $a(x) + b(x) = x + x^2 + x^3$. Відносно цих двох операцій на множині F_n – всіх поліномів степеня $\le n$ має місце наступне твердження для деякої V, $V, V \subset F_n$- довільний поліном $a(x) \in V$ можливо уявити, як добуток фіксованого породжуючого полінома $q(x)$ та деякого підходящого $s(x)$, то

$$a(x) = q(x)s(x).$$

Тобто, якщо відомий породжуючий поліном, то тим самим вичерпуються можливі добутки на поліном $s(x)$- довільний поліном степеня менше n.

Нагадаємо, що для визначення довільного коду треба навести повний список кодових слів, а для лінійного коду потрібен список кодових векторів. Для циклічного коду достатньо привести лише один кодовий поліном $q(x)$ – породжуючий. Визначимо породжуючу матрицю циклічного коду по заданому $q(x)$, де $q(x) = q_0 + q_1 x + ... + q_m x^m$- поліном степеня m, де $n = m + k$. Розглянемо наступні поліноми $\{q(x), xq(x), x^2 q(x), ..., x^{k-1} q(x)\}$, які є кодовими поліномами, бо степінь їх не перевищує $n-1$. З іншого, векторного, боку всі вони утворюють лінійно незалежну систему і всякий кодовий вектор є лінійною комбінацією цих базисних векторів. Тоді випишемо в рядок

матриці ці поліноми із наведеної вище множини та отримаємо породжуючу матрицю порядку $k \times n$:

Матриця матиме вигляд:

$$k_{рядків}\left(\begin{Bmatrix} g_0 & g_1 & \dots & g_m & 0 & \dots & 0 \\ 0 & g_0 & g_1 & g_{m-1} & g_m & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & g_0 & g_1 & g_m \end{Bmatrix}\right) = \begin{pmatrix} g(x) \\ xg(x) \\ \dots \\ x^{k-1}g(x) \end{pmatrix}$$

Наприклад, розглянемо двійковий циклічний код довжини 7 із породжуючим поліномом $g(x) = 1 + x^2 + x^3 = (1011000)$. Тоді x g(x)=x+x^3+x^4=(0101100)= $0*1+1*x+0*x^2+1*x^3+1*x^4+0*x^5+0*x^6$;

$x^2 g(x) = (0010110)$, $x^3 g(x) = (0001011)$.

Отримаємо матрицю:

$$G = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

Серед циклічних кодів існують такі, що виправляють довільну задану кількість помилок при передачі, бо справедлива наступна теорема(без доведення).

Для довільних значень m та t (t<(2^m-1)/2) існує двійковий циклічний код довжини 2^m-1, який виправляє всі комбінації із t або меншого числа помилок і має не більше ніж mt перевірочних символів.

Про перевірочний поліном h(x) треба сказати, що він задовольняє рівності $h(x) = \frac{x^n - 1}{g(x)}$, де g(x) – породжуючий поліном. Використовуючи таке визначення, можливо побудувати перевірочну матрицю циклічного коду. А саме, нехай h(x)=h$_0$+h$_1$x+h$_2$x^2+…+h$_k$x^к – породжуючий поліном. Для множини F$_n$- всіх поліномів степеня n маємо, що x^n=1,тобто наведена умова для g(x) та h(x) матиме вигляд g(x)*h(x)=0.

Розглянемо матрицю H порядку m x n , що має наступний вид:

$$H = \begin{pmatrix} 0 & \dots & 0 & h_k & \dots & h_1 & h_0 \\ 0 & 0 & h_k & h_{k-1} & \dots & h_0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ h_k & \dots & \dots & h1 & h_0 & 0 & 0 \end{pmatrix}$$

де перший рядок складають коефіцієнти полінома h(x) записані в оберненому порядку, а наступні рядки складені для поліномів xh(x),… $x^{m+1}h(x)$.

Наприклад, для наведеного вище циклічного коду(7,4) n=7 із породжуючим поліномом $g(x)=1+x^2+x^3$, якій ділить X^7-1.

Дійсно:

$$X^7-1=(X-1)(X^3+X+1)(X^3+X^2+1),$$

тобто перевірочний поліном:

$$h(x) = (X-1)(X^3+X+1)=1+X^2+X^3+X^4=(1011100),$$

тоді перевірочна матриця має вигляд:

$$H=\begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

Перевірка побудованих матиць здійснюється наступним матричним рівнянням

$$G * H^T = \varnothing$$

де $\varnothing$ – нульова матриця.

Завдання:

1) Знайти G-породжуючу та H-перевірочну матриці циклічного коду заданої довжини n з породжуючим поліномом g(x) , та зробити перевірку свого варіанту.

2) Побудувати всі кодові слова для коду із завдання 1.

3) Скласти програму для реалізації алгоритму.

Варіанти:

1. n=7, $g(x)=1+x^3+x$;
2. n=7, $g(x)=(x-1)(1+x^2+x^3)$;

3. $n=9$, $g(x)=1+x^3+x^6$;
4. $n=9$, $g(x)=1+x+x^2$;
5. $n=9$, $g(x)=(x-1)(1+x+x^2)$;
6. $n=9$, $g(x)=(x-1)(1+x^3+x^6)$;
7. $n=15$, $g(x)=x-1$;
8. $n=15$, $g(x)=1+x+x^2$;
9. $n=15$, $g(x)=1+x^3+x^4$;
10. $n=15$, $g(x)=1+x+x^4$;
11. $n=15$, $g(x)=1+x+x^2+x^3+x^4$;
12. $n=15$, $g(x)=x^2+x+1$;
13. $n=15$, $g(x)=(x-1)(1+x+x^2)$;
14. $n=15$, $g(x)=(x-1)(1+x^3+x^4)$;
15. $n=15$, $g(x)=(x-1)(1+x+x^4)$;

Лабораторна робота N5

Тема: Шифрування та дешифрування методами Цезаря й Відженера

Мета: Оволодіти методами кодування тексту засобами криптографії

Метод Цезаря.

Теоретичні відомості. Розглянемо задачу, що полягає в захисті інформації від несанкціонованого доступу. Розглянемо алфавіт укр. мови та занумеруємо літери так, щоб А мала номер 0, Б – номер 1 і т.д. та "розташуємо" їх в рядок згідно номерів. Внизу випишемо ще такий самий рядок, зсунувши його вправо на одну літеру та переставимо літеру Я на початок другого рядка:

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

Цим самим виконаємо зсув на одну літеру. Аналогічним чином визначаємо зсув на N, або що те саме-циклічну перестановку літер алфавіту. Використаємо цей другий рядок для кодування тексту з метою конфіденційності (захисту від несанкціонованого використання).

Розглянемо обернену задачу – декодування тексту, закодованого при певному зсуві алфавіту. Для того щоб взламати текст закодований методом цезаря застосовується наступний підхід:

Використаємо статистичні методи для визначення величини зсуву наступним чином:

1. Побудуємо таблицю 1 ймовірності всіх літер алфавіту на базі величезного текстового файлу.
2. Побудуємо таблицю 2 частот появи літер в закодованому текстовому файлі.
3. Визначаємо величину зсуву алфавіту:

а) Підрахувати суму різниць частоти літери із кожного рядка та ймовірності появи відповідної літери. Наприклад для уявного зсуву на 1 маємо таку суму:

$$S_1=|V_2(А)-V_1(Я)+V_2(Б)-V_1(А)+V_2(В)-V_1(Б)+...+V_2(Я)-V_1(Ю)|$$

де $V_2(А)$ – частота появи літер А в декодованому тексті, $V_1(Я)$ – ймовірність появи літери Я в укр. тексті.

б) Якщо позначити через S_i-суму, побудовану в а) для уявного зсуву на *i* всіх літер алфавіту, то можливо побудувати множину $\{S_i\}$,i=0,...30 в якій найменший елемент буде відповідати величині зсуву літер алфавіту.

4.Таким чином статистичним методом знайдемо можливість для декодування тексту, про який відомий спосіб кодування.

Завдання 1:

Використати номер свого прізвища в списку групи, як величину зсуву рядка літер алфавіту для кодування інформації. Побудувати програму на основі описаного вище алгоритму для кодування та декодування тексту з лаб. роботи №1.

Метод Відженера.

Кодування тексту більш складним засобом криптографії.

Теоретичні відомості.

Розглянемо задачу "шифрування" тексту (інформації) з метою захисту її від зловмисників, розглянуту в попередній роботі, та відмітимо слабкість методу Цезаря щодо "зламу" шифру-зсуву алфавіта. Вдосконалимо цей метод наступним чином:

1) випишемо в рядок №1 весь текст для кодування;

2) виберемо слово (зашифровуючий ключ) та заповнимо його копіями рядок, №2, розташований під рядком №1.

3) використаємо цифрові коди літер алфавіту якими виписано текст та слово-ключ.

Наприклад:

код(А)=0,код(Б)=1,...,код(Я)=30

4) виконаємо операцію додавання за модулем кодів літер вказаних рядків та запишемо результат такого "політерного" додавання до третього рядка.

Наприклад, закодуємо український алфавіт ключем **КУ.**

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
А	Б	В	Г	Д	Е	Є	Ж	З	І	Ї	Й	К	Л	М	Н	О	П	Р	С
12	**21**	**12**	**21**	**12**	**21**	**12**	**21**	**12**	**21**	**12**	**21**	**12**	**21**	**12**	**21**	**12**	**21**	**12**	**21**
К	**У**	**К**	**У**	**К**	**У**	**К**	**У**	**К**	**У**	**К**	**У**	**К**	**У**	**К**	**У**	**К**	**У**	**К**	**У**
12	**22**	**14**																	
К	**У**	**М**																	

Якщо код(А)=0,

код(К)=12,

код(У)=21,

то (код(А)+ код(К))mod30=12 = «К»

код(Б)=1,

то (код(Б) + код(У))mod 30=22 = «У»

код (В)=2,

то код (В) + код (К)) mod 30=14 = «М»

код(М)=14

В результаті дії 4) отримаємо зашифрований текст в третьому рядку, який в подальшому передається відправником по каналу зв'язку для отримувача інформації, який знає, як було закодовано текст т.ч. розшифрує інформацію. Можлива також передача ключа разом з інформацією. Якщо ключ забуто, але відома його довжина(кількість літер), то можливе декодування при наявності програмного засобу.

Використовуючи датчик випадкових чисел від 0 до 30 для побудови „довгого” зашифровуючого ключа, можливо створити засіб кодування тексту, який неможливо декодувати без отримання ключа по каналах зв'язку.

Завдання 2:

Використати номер свого прізвища в списку групи для визначення в алфавіті першої літери слова-ключа, а друга літера цього слова буде наступною за першою. Побудувати програмний засіб для кодування тексту описаним вище алгоритмом та декодування .

Лабораторна робота N 6

Тема: Шифрування та дешифрування алгоритмом RSA

Мета: Освоєння несиметричних алгоритмів криптографії.

Теоретичні відомості.

У RSA відкритим параметром є багаторазрядний модуль перетворення (не менш 512 біт)

$$N_j = P_j \cdot Q_j ,$$

де P_j, Q_j – сильні прості числа відповідної розрядності.

Ключі E_k. і D_k зв'язані співвідношенням

$$E_k \cdot D_k \equiv 1(\mathrm{mod}\ \varphi(N_j)) ,$$

де $\varphi(N_j) = (P_j - 1)(Q_j - 1)$ – функція Ейлера.

Ключ формування підпису E_k є закритим, а ключ зняття підпису D_k зазавичай відкритий.

Формування зашифрованої інформації здійснюється за правилом:

$$ЗІ = ВІ^{E_k} (\mathrm{mod}\ N_j),$$

де $ВІ$ – відкрита інформація;

$ЗІ$ – зашифрована інформація.

Для розшифрування повідомлення використовується співвідношення

$$ВІ' = (ЗІ')^{D_k} (\mathrm{mod}\ N_j),$$

де $ВІ'$ – відкритий текст, отриманий прийомною стороною із прийнятої закритої інформації ($ЗІ'$).

Вважаємо, що :

1) Символ тексту розглядаємо, як числа (ASCII кодом), та позначаємо через m .

2) Відомі обом сторонам взаємнопрості числа e і d,

де e – ключ шифрування, d – ключ дешифрування.

Піднесення до ступеня e числа m означає шифрування, а піднесення

отриманого числа до ступеня d буде дешифруванням символу, що має код m .

Завдання:

1. Побудувати програмний засіб для реалізації наведеного методу.
2. Яким чином можливо "зламати" текст, зашифрований цим методом?

Лабораторна робота №7

Тема: Шифрування та дешифрування алгоритмами ДСТУ 28147:2009 та DES

Мета: Побудувати шифруючі перетворення на принципах розсіювання та перемішування з метою максимально ускладнити підбір ключа шифрування.

Теоретичні відомості.

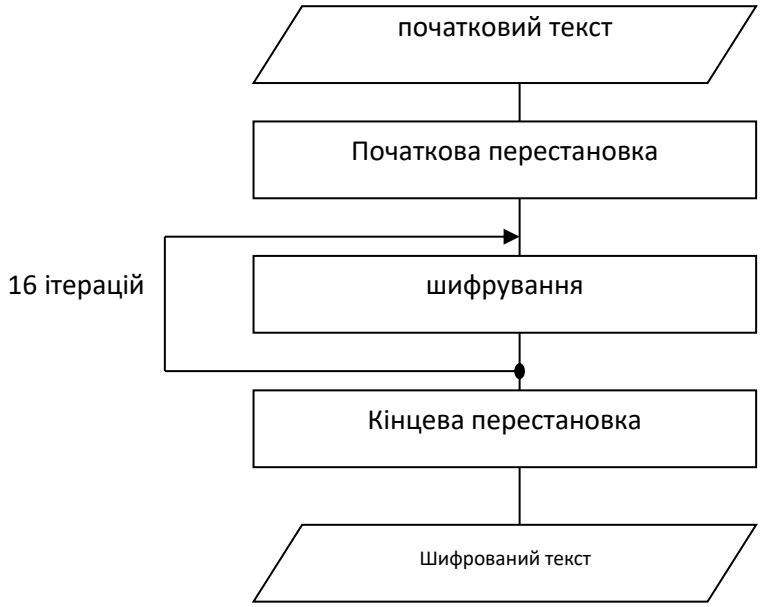
Розсіювання являє собою розповсюдження впливу одного знаку тексту на багатьох знаків зашифрованого тексту з метою заховати статистичні властивості тексту.

Перемішування означає використання таких шифруючих перетворень, які ускладнюють відтворення зв’язку між текстом та зашифрованим його видом.

Найрозповсюдженішим способом досягнення ефектів розсіювання та перемішування є використання послідовності простих ключів, де кожен простий шифр найчастіше використовує прості перестановки та підстановки. Перестановка означає просте переставлення символів текстів, виконане за допомогою секретного ключа. Під підстановкою розуміють заміну кожного символа тексту іншим символом із того ж самого алфавіту, яка також визначається секретним ключем.

DES (Data Encryption Standard)

DES (Data Encryption Standard) – стандарт шифрування США використовує комбінацію підстановок та перестановок шляхом шифрування 64-бітових блоків даних за допомогою 64-бітового ключа, в якому значущими є 56, а решта служить для контролю на парність. Дешифрування здійснюється повторенням операцій шифрування в оберненому порядку. Узагальнена схема шифрування в алгоритмі DES має вигляд:



Спрощений алгоритм має вигляд:

Нехай прочитано 8-байтів в буфер Т, який перетворюється за допомогою матриці початкової перестановки IP вигляду:

$$\begin{bmatrix} 58 & 50 & 42 & 34 & 26 & 18 & 10 & 2 \\ 60 & 52 & 44 & 36 & 28 & 20 & 12 & 4 \\ 62 & 54 & 46 & 38 & 30 & 22 & 14 & 6 \\ 64 & 56 & 48 & 40 & 32 & 24 & 16 & 8 \\ 57 & 49 & 41 & 33 & 25 & 17 & 9 & 1 \\ 59 & 51 & 43 & 35 & 27 & 19 & 11 & 3 \\ 61 & 53 & 45 & 37 & 29 & 21 & 13 & 5 \\ 63 & 55 & 47 & 39 & 31 & 23 & 15 & 7 \end{bmatrix}$$

Згідно наведеної матриці IP 58 біт з Т стає 1-м, 50-й стає 2-м і т.д., тобто з Т маємо T_0 і запишемо $T_0 = IP(T)$. Потім T_{10} розділимо на L_0 - ліву та R_0 - праву половини. Потім виконується 16 ітерацій, де в результаті першої ітерації маємо $T_1, T_1 = L_1 R_1$ - тобто конкатенація рядків, а друга така $L_2 = R_1$, $R_2 = L_1 + f(R_1, K_1)(\mathrm{mod}2)$, де f – функція шифрування, аргументами якої будуть R_1 - права (4-байти) частина з попереднього кроку та K_1 – 48-бітовий ключ, отримані в результаті перетворень з 64-бітового ключа К. З метою

спрощення вважатимемо, що $K_i = K$ а функція $f\ xor$ – операція додавання R_1 до K по модулю 2. На останньому кроці матимемо $T_{16} = R_{16}L_{16}$ з якої відновлюємо позиції бітів за допомогою матриці оберненої перестановки IP^{-1} :

40	6	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Отримаємо зашифрований блок T' із вхідного T.

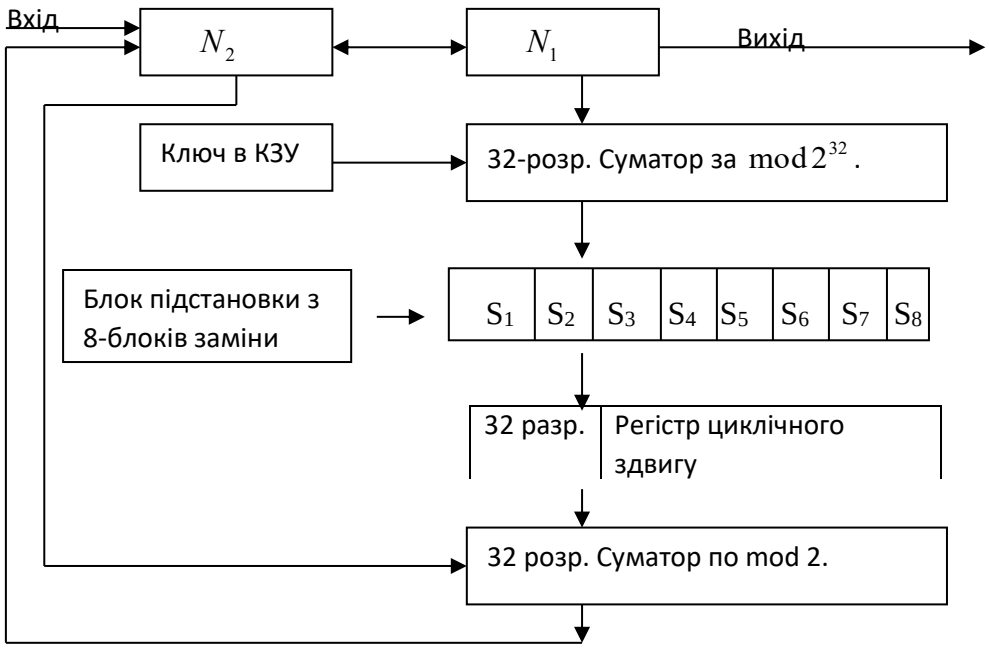
Дешифрування є інверсним (оберненим) по відношенню до процесу шифрування.

Стандарт шифрування даних – ДСТУ 28147:2009

Стандарт шифрування даних – ДСТУ 28147:2009 являє собою алгоритм для апаратної та програмної реалізації криптографічного перетворення блоків даних (по 64 біти) із 256-бітовим ключем. Серед чотирьох режимів роботи алгоритму схематично розглянемо:

- шифрування в режимі простої заміни;
- шифрування в режимі гамування.

Для реалізації алгоритму шифрування даних наведемо схему:



Дані, що підлягають шифруванню, розбивають на 64-розрядні блоки та поблочно проходять тут процедуру 32-х ітерацій шифрування ключем К, що зберігається в КЗУ у вигляді восьми 32-розрядних К і підключів (чисел), $К=К_7К_6К_5К_4К_3К_2К_1К_0$.

Робота цієї процедури розпочинається з поділу вхідного блоку на дві половини, по 32 біти, де в першій половині містяться молодші біти зліва направо, а в правій – старіші, які вводять до накопичувачів N_1 та N_2 в оберненому порядку, тобто зліва направо розміщено 32-й, 31-й…1-й біти.

Перша ітерація процедури полягає в послідовному виконанні двох операцій додавання:

1) змісту N_1 із ключем K_0 за модулем 32

2) потім додавання за модулем 2 до результату попередньої операції змісту накопичувача N_2, результат пишемо в N_1 а в N_2 заносимо початковий зміст регістру N_1.

Першу операцію додавання звуть підстановкою (заміною) бо виконується блоком підстановки , що має 8 вузлів по 64-біта кожний, в першому суматорі СМ1. Кожен вузол можливо уявити у вигляді таблиці – перестановки шістнадцяти чотирирозрядних двійкових числа (в діапазоні від 0000….1111) та є загальним (рідко змінюються). Вхідні дані визначають рядок 8 таблиці, а число 8 рядку буде вихідним, які послідовно об'єднуються у 32-розрядний блок. Наступна операція – зсув циклічно вліво на 11 розрядів вихідного блоку, отриманого після .

Друга операція – порозрядне додавання по модулю 2 виконується в другому суматорі СМ2 над вихідним блоком після зсуву, та змістом накопичувача N_2. Результат цієї операції записують в N_1, а збережене старе значення N_1 записують в N_2. Перша ітерація процедури шифрування на цьому завершена. Аналогічно, на другому циклі з КЗУ зчитується K_1, потім K_2, а починаючи із 25 – 32-ї ітерацій порядок зчитування підключів K_i змінюється на протележний.

На заключній 32-й ітерації результат із суматора вводиться в накопичувач N_2, а в N_1 вводимо попереднє значення накопичувача N_2. Об'єднавши послідовно значення із N_1 та N_2 отримаємо зашифрований блок даних.

Розшифрування здійснюється в оберненому порядку, тільки порядок зчитування підключів із КЗУ в наступному порядку: K_0, 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.

Опис режиму простої заміни закінчено.

Шифрування в режимі гамування полягає у застосуванні гами шифрів, що побудовані "випадковим" чином із використанням датчиків псевдовипадкових чисел. Ці шифри використовують не тільки для процедури кодування, але й передаються для виконання дешифрування. Робота алгоритму ДСТУ 28147:2009 відрізняється від роботи в попередньому

режимі саме тим, що замість блоку підстановки використано гаму шифрів, що буде змінена після закінчення шифрування та передачі даних.

Завдання:

Побудувати програмну реалізацію алгоритмів.

1) DESa
2) ДСТУ 28147:2009

Порівняти ефективність шифрування між цими методами

Список використаної літератури

1. Смірнов О.А., Конопліцька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В., Поліщук Л.І. Інформаційна безпека в комп'ютерних мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2020. – 294 с. Режим доступу: http://dspace.kntu.kr.ua/jspui/handle/123456789/9799
2. Смірнов О.А., Гнатюк С.О., Кавун С.В., Терейковський І.А., Жмурко Т.О., Смірнов С.А., Коваленко А.С. Основи безпеки в комп'ютерних мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2018. – 177 с.
3. Смірнов О.А., Кавун С.В., Доренський О.П., Вялкова В.І. Інформаційна безпека в комп'ютерних мережах. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 151 с.
4. Смірнов О.А., Стасєв Ю.В. Бараннік В.В. Захист інформації в автоматизованих системах управління. Навчальний посібник – Харків: ХУПС, 2015. – 264 с.
5. Смірнов О.А., Кузнецов О.О., Євсеєв С.П., Мелешко Є.В., Король О.Г. Методи та алгоритми симетричної криптографії. Навчальний посібник для студентів вищих навчальних закладів напрямів підготовки 8.050102 «Комп'ютерна інженерія». За ред. О.О. Кузнецова. Гриф "Навчальний посібник" надано у відповідності з листом Міністерства освіти і науки, молоді та спорту України від 26.04.2012 року № 1/11-5762. – Кіровоград: КНТУ 2012. – 315 с.
6. Смірнов О.А., Мелешко Є.В., Семенов С.Г. Методи та засоби обробки сигналів і даних в інформаційних системах. Навчальний посібник для студентів вищих навчальних закладів напрямів підготовки 8.050102 «Комп'ютерна інженерія». За ред. О.А. Смірнова Гриф "Навчальний посібник" надано у відповідності з листом Міністерства освіти і науки, молоді та спорту України від 17.04.2012 року № 1/11-5249. – Кіровоград: КНТУ 2012. – 250 с.

7. Захист інформації в автоматизованих системах управління : навчальний посібник / Уклад. І. А. Пількевич, Н. М. Лобанчикова, К. В. Молодецька. – Житомир : Вид-во ЖДУ ім. І. Франка, 2015. – 226 с.

8. Остапов С. Е. Технологія захисту інформації : навчальний посібник / С. Е. Остапов, С. П. Євсеєв, О. Г. Король. – Х. : Вид. ХНЕУ, 2013. – 476 с.

9. Електронне урядування та електронна демократія: навч. посіб.: у 15 ч. / за заг. ред. А.І. Семенченка, В.М. Дрешпака. – К., 2017. Частина 13: Захист інформації в системах електронного урядування / [О.М. Хошаба]. – К.: ФОП Москаленко О. М., 2017. – 72 с.

10. Абакумов, В. Г. Теорія інформації та кодування. Ч. 1 [Електронний ресурс] : навчальний посібник / В. Г. Абакумов ; НТУУ «КПІ». – Електронні текстові дані (1 файл: 3,42 Мбайт). – Київ : НТУУ «КПІ», 2011.

11. Anubhab Baksi. «Classical and Physical Security of Symmetric Key Cryptographic Algorithms (Computer Architecture and Design Methodologies)». Springer. 2022 . – 300 p.

12. Tsuyoshi Takagi, Kirill Morozov, Dung Hoang Duong. «Mathematics of Post-quantum Cryptography». Springer. 2021. – 300 p.

13. Dr.Sonali Ridhorkar. «Elliptic Curve Cryptography: Implementation Issues: Key Establishment Protocol». LAP LAMBERT Academic Publishing. 2021. – 152 p.

14. Aiden A. Bruen, Mario A. Forcinito, James M. McQuillan «Cryptography, Information Theory, and Error-Correction». Wiley. 2021. – 688 p.

15. Duncan Buell. «Fundamentals of Cryptography: Introducing Mathematical and Algorithmic Foundations». Springer. 2021. – 296 p.

16. Lisa Bock. «Modern Cryptography for Cybersecurity Professionals». Packt Publishing. 2021. – 286 p.

Допоміжна

17. Smirnov, O., Lakhno, V., Akhmetov, B., Chubaievskyi, V., Khorolska, K., Bebeshko, B. «Selection of a Rational Composition of İnformation Protection Means Using a Genetic Algorithm». In: Rajakumar, G., Du, KL., Vuppalapati, C., Beligiannis, G.N. (eds) Intelligent Communication Technologies and Virtual Mobile Networks. Lecture Notes on Data Engineering and Communications Technologies, vol 131. 2023. Springer, Singapore. pp. 21-34. (Scopus). Режим доступу: https://www.scopus.com/record/display.uri?eid=2-s2.0-85134768958&origin=resultslist&sort=plf-f&featureToggles=FEATURE_NEW_DOC_DETAILS_EXPORT:1,FEATURE_EXPORT_REDESIGN:1

18. Smirnov O., Kuznetsov A., Kryvinska N., Kiian A., Kuznetsova K. «Full Non-Binary Constant-Weight Codes». SN Computer Science, Vol 2, 337, 2021. https://doi.org/10.1007/s42979-021-00739-w (Scopus). Режим доступу: https://www.scopus.com/record/display.uri?eid=2-s2.0-85131801425&origin=resultslist&sort=plf-f&featureToggles=FEATURE_NEW_DOC_DETAILS_EXPORT:1

19. Smirnov O., Kuznetsov A., Lokotkova I., Kuznetsova T., Florov S., Lebid O. «Using Orthogonal Signals to Hide Information in Images». 4 IEEE International Conference on Advanced Information and Communication Technologies (AICT) – 2021, Lviv, Ukraine, September 21-25, 2021. P. 255-260. (Scopus). Режим доступу: https://www.scopus.com/record/display.uri?eid=2-s2.0-85124008010&origin=resultslist&sort=plf-f

20. Smirnov O., Kuznetsov A., Girzheva O., Kiian A., Nakisko O., Kuznetsova T. «Advanced Code-Based Electronic Digital Signature Scheme». 2020 IEEE International Conference on Problems of Infocommunications Science and Technology, PIC S and T 2020, Kharkiv, 6 October 2020-9 October 2020, P. 358-362. (Scopus). Режим доступу: https://www.scopus.com/record/display.uri?eid=2-s2.0-

85114388319&origin=SingleRecordEmailAlert&dgcid=raven_sc_author_ru_ru_email&txGid=030a5fa3ef0a593fa1705f0c73130f01

21. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256. (Scopus). Режим доступу: https://www.scopus.com/record/display.uri?eid=2-s2.0-85096919335&origin=SingleRecordEmailAlert&dgcid=raven_sc_author_ru_ru_email&txGid=612e931a8e3eb73102c95ce1ccc90d0d

22. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114. (Scopus). Режим доступу: https://www.scopus.com/record/display.uri?eid=2-s2.0-85096412796&origin=SingleRecordEmailAlert&dgcid=raven_sc_author_ru_ru_email&txGid=feb5eedf8c0626618743ca09212f9cd6

23. Smirnov O., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». Journal of theoretical and applied information technology Vol.98. No 21, 2020, P. 3334-3346. (Scopus). Режим доступу: https://www.scopus.com/record/display.uri?eid=2-s2.0-85096438117&origin=SingleRecordEmailAlert&dgcid=raven_sc_author_ru_ru_email&txGid=1e91df71a9e62824506812d4d2f72e33

24. Smirnov O., Kuznetsov A., Kiian A., Babenko V., Perevozova I.,Chepurko I. «New Approach to the Implementation of Post-Quantum Digital Signature Scheme». 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 166-171. (Scopus). Режим доступу: https://www.scopus.com/record/display.uri?eid=2-s2.0-85087899476&origin=resultslist&sort=plf-

f&src=s&sid=3b1b7490cfd07f8a6eb2e90ad30c8c6d&sot=autdocs&sdt=autdocs&sl=18&s=AU-ID%2857208667815%29&relpos=2&citeCnt=0&searchTerm

25. Smirnov O., Kuznetsov A., Kiian A., Babenko V., Perevozova I.,Chepurko I. «New Approach to the Implementation of Post-Quantum Digital Signature Scheme». 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 166-171. (Scopus). Режим доступу: https://www.scopus.com/record/display.uri?eid=2-s2.0-85087899476&origin=resultslist&sort=plf-f&src=s&sid=3b1b7490cfd07f8a6eb2e90ad30c8c6d&sot=autdocs&sdt=autdocs&sl=18&s=AU-ID%2857208667815%29&relpos=2&citeCnt=0&searchTerm

26. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 172-177. (Scopus). Режим доступу: https://www.scopus.com/record/display.uri?eid=2-s2.0-85087876353&origin=resultslist&sort=plf-f&src=s&sid=3b1b7490cfd07f8a6eb2e90ad30c8c6d&sot=autdocs&sdt=autdocs&sl=18&s=AU-ID%2857208667815%29&relpos=4&citeCnt=0&searchTerm

27. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies, vol 48. Springer, Cham. 2021. pp 557-587. (Scopus). Режим доступу: https://www.scopus.com/record/display.uri?eid=2-s2.0-85087208231&origin=resultslist&sort=plf-f&src=s&sid=c4094ccaebdad4549a0820b2d8742aa3&sot=autdocs&sdt=autdocs&sl=18&s=AU-ID%2857208667815%29&relpos=0&citeCnt=0&searchTerm

28. Smirnov O. Kuznetsov A., Zaicenko Yu., Pastukhov M., Oleshko O., Kuznetsova K., «Formation of Discrete Signals with Special Correlation Properties». International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2019; Odessa; Ukraine; 9-13 September 2019. P.22-28. (Scopus). Режим доступу: https://www.scopus.com/record/display.uri?eid=2-s2.0-85091704115&origin=AuthorNamesList&txGid=6047f73642b838afa9b36c54ad7e29d5

29. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». International Journal of Computing; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407. (Scopus). Режим доступу: https://www.scopus.com/record/display.uri?eid=2-s2.0-85084440832&origin=resultslist&sort=plf-f&src=s&sid=78e9700b01a40be3c0799a1567340a7f&sot=autdocs&sdt=autdocs&sl=18&s=AU-ID%2857208667815%29&relpos=11&citeCnt=0&searchTerm

30. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), Kyiv, Ukraine, 8 – 11 October 2019 . P.517-522. (Scopus). Режим доступу: https://www.scopus.com/record/display.uri?eid=2-s2.0-85083667464&origin=resultslist&sort=plf-f&src=s&sid=2b6a0139fad18bb19a964441b5bded76&sot=autdocs&sdt=autdocs&sl=18&s=AU-ID%2857208667815%29&relpos=2&citeCnt=0&searchTerm

31. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». CEUR Workshop Proceedings, Vol 2588, P. 90-106, 2019. (Scopus). Режим доступу: https://www.scopus.com/record/display.uri?eid=2-s2.0-

85083237488&origin=resultslist&sort=plf-f&src=s&sid=4e89c5e5e6bd68a6310e60ba77c04b42&sot=autdocs&sdt=autdocs&sl=18&s=AU-ID%2857208667815%29&relpos=9&citeCnt=0&searchTerm

32. Smirnov, O., Kuznetsov, A., Kiian, A., Pushkar'ov, A., Mialkovskyi, D ., Kuznetsova, T., «Code-Based Schemes for Post-Quantum Digital Signatures», 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019; Metz; France; 18-21 September 2019. P. 707-712. Режим доступу: https://www.scopus.com/record/display.uri?eid=2-s2.0-85077116930&origin=resultslist&sort=plf-f&src=s&sid=e66ec7ff6625e5acea5827784acaead6&sot=autdocs&sdt=autdocs&sl=18&s=AU-ID%2857208667815%29&relpos=0&citeCnt=0&searchTerm.

33. Smirnov, O., Kuznetsov, A., Kiian, A., Babenko, B., Zhosan, H., Prokopovych-Tkachenko, D., «Soft Decoding Method for Turbo-Productive Codes», 2019 3rd International Conference on Advanced Information and Communications Technologies, AICT 2019, Lviv, Ukraine, 2-6 July, 2019, P. 129-134. Режим доступу: https://www.scopus.com/record/display.uri?eid=2-s2.0-85073344541&origin=resultslist&sort=plf-f&src=s&sid=d686ad0e756d5334e61f7258a32f58c1&sot=autdocs&sdt=autdocs&sl=18&s=AU-ID%2857208667815%29&relpos=1&citeCnt=0&searchTerm

34. Smirnov, O., Kuznetsov, A., Kiian, A., Zamula, A., Rudenko, S., Hryhorenko, V., «Variance Analysis of Networks Traffic for Intrusion Detection in Smart Grids», 2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS), Kyiv, Ukraine April 17-19, 2019 P. 353-358. Режим доступу: https://www.scopus.com/record/display.uri?eid=2-s2.0-85069931997&origin=resultslist&sort=plf-f&src=s&sid=d686ad0e756d5334e61f7258a32f58c1&sot=autdocs&sdt=autdocs&sl=18&s=AU-ID%2857208667815%29&relpos=2&citeCnt=0&searchTerm

35. Smirnov, O., Kuznetsov, A., Kavun, S., Babenko, B., Nakisko, O., Kuznetsova, K., «Malware Correlation Monitoring in Computer Networks of

Promising Smart Grids», 2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS), Kyiv, Ukraine April 17-19, 2019 P. 347-352. Режим доступу: https://www.scopus.com/record/display.uri?eid=2-s2.0-85069931008&origin=resultslist&sort=plf-f&src=s&sid=d686ad0e756d5334e61f7258a32f58c1&sot=autdocs&sdt=autdocs&sl=18&s=AU-ID%2857208667815%29&relpos=3&citeCnt=0&searchTerm

36. Smirnov, O., Kuznetsov, A., Kiian, A., Kuznetsova, K., Ivko, T., Prokopovych-Tkachenko, D., «Soft Decoding Based on Ordered Subsets of Verification Equations of Turbo-Productive Codes», CEUR Workshop Proceedings Volume 2353, CEUR-WS 2019, Pages 873-884. Режим доступу: https://www.scopus.com/record/display.uri?eid=2-s2.0-85065482781&origin=resultslist&sort=plf-f&src=s&sid=d686ad0e756d5334e61f7258a32f58c1&sot=autdocs&sdt=autdocs&sl=18&s=AU-ID%2857208667815%29&relpos=5&citeCnt=0&searchTerm

37. Smirnov A.A., Kuzne tsov A.A., Danilenko D.A., Berezovsky A., «The statistical analysis of a network traffic for the intrusion detection and prevention systems», Telecommunications and Radio Engineering. – Volume 74, Issue 1. – Begel House Inc. – 2015. – P. 61-78. Режим доступу: https://www.scopus.com/record/display.uri?eid=2-s2.0-84938096221&origin=resultslist&sort=plf-f&src=s&sid=d686ad0e756d5334e61f7258a32f58c1&sot=autdocs&sdt=autdocs&sl=18&s=AU-ID%2857208667815%29&relpos=6&citeCnt=33&searchTerm

38. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Поліщук Л.І., Смірнов С.А. «Дослідження статистичної стійкості та швидкісних характеристик запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» Вісник Хмельницького національного університету. Серія: «Технічні науки», № 2 (307). С. 46-52. 2022. Режим

доступу: http://journals.khnu.km.ua/vestnik/?cat=65 (Фахове видання. Категорія «Б»)

39. Смірнов О.А., Смірнова Т.В., Константинова Л.В., Смірнов С.А., Якименко Н.М., «Дослідження стійкості до лінійного криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» Системи управління, навігації та зв'язку, 2022, № 1(67). С. 84-89. Режим доступу: http://journals.nupp.edu.ua/sunz/article/view/2449/1918 (Фахове видання. Категорія «Б»)

40. Смірнов О.А., Смірнов С.А., Поліщук Л.І., Смірнова Т.В., Конопліцька-Слободенюк О.К. Метод формування антивірусного захисту даних з використанням безпечної маршрутизації метаданих. Кібербезпека: освіта, наука, техніка. – Том 3 № 3. – Київ: КУ ім. Бориса Грінченка. – 2019. – С.

41. Смірнов О.А., Смірнов С.А., Поліщук Л.І., Конопліцька-Слободенюк О.К., Смірнова Т.В. GERT-моделі технології хмарного антивірусного захисту. Кібербезпека: освіта, наука, техніка. – Том 2 № 2. – Київ: КУ ім.. Бориса Грінченка. – 2018. – С. 7-30. https://doi.org/10.28925/2663-4023.2018.2.730 .Режим доступу: http://nbuv.gov.ua/UJRN/cest_2018_2_3 63-87. https://doi.org/10.28925/2663-4023.2019.3.6387. Режим доступу: http://nbuv.gov.ua/UJRN/cest_2019_3_7

42. Смірнов О.А., Мелешко Є.В., Хох В.Д. Дослідження методів аудиту систем управління інформаційною безпекою. Системи управління, навігації та зв’язку. – Випуск 1 (41). – Полтава: ПолтНТУ. – 2017. – С. 38-42.. Режим доступу: http://nbuv.gov.ua/UJRN/suntz_2017_1_12

43. Смирнов А.А., Смирнов С.А., Дидык А.К., Дреев А.Н. Способ контроля линий связи телекоммуникационной системы облачного антивируса. Способ контроля линий связи телекоммуникационной системы облачного антивируса. Збірник наукових праць Харківського університету

Повітряних Сил. Випуск 2 (47). – Харків: ХУПС. – 2016. – С. 121-127. Режим доступу: http://nbuv.gov.ua/UJRN/ZKhUPS_2016_2_32

44. Смирнов А.А., Смирнов С.А. Дидык А.К., Дреев А.Н. Модели системы нейросетевых экспертов безопасной маршрутизации в облачных антивирусных системах. Збірник наукових праць "Системи обробки інформації". – Випуск 3(140). – Х.: ХУПС – 2016. – С. 36-39.Режим доступу: http://www.hups.mil.gov.ua/periodic-app/article/16443