ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД УКООПСПІЛКИ
«ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ»

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ БІЗНЕСУ
ТА СУЧАСНИХ ТЕХНОЛОГІЙ
ФОРМА НАВЧАННЯ ДЕННА

КАФЕДРА МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ ТА СОЦІАЛЬНОЇ ІНФОРМАТИКИ

Допускається до захисту

Завідувач кафедри _____О.О. Ємець
(підпис)
«_____»_____2021 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО БАКАЛАВРСЬКОЇ РОБОТИ
на тему
ШИФР ГРОНСФЕЛЬДА В КОДУВАННІ: ПРОГРАМУВАННЯ ТА ДОСЛІДЖЕННЯ

зі спеціальності 122 «Комп'ютерні науки та інформаційні технології»

Виконавець роботи Заливчий Михайло Вячеславович

_____ «___»_____2021 р.
(підпис)

Науковий керівник проф., к. ф.-м. н. Ємець Єлизавета Михайлівна

_____ «___»_____2021 р.
(підпис)

ПОЛТАВА 2021 р.

ЗМІСТ

ВСТУП

Процес написання програмного коду або скриптів з метою реалізації деякого алгоритму певною мовою програмування називається кодуванням.

Деякі люди плутають цей термін із поняттям програмування. Кодування є лише частиною останнього і знаходиться на ряду з аналізом, проектуванням, компіляцією, тестуванням, налагодженням і супроводженням.

Розібравшись із ключовими елементами розробки програмного забезпечення, можемо визначити основні задачі проекту.

Мета роботи – створення зручної програми для вирішення питань з теми «Шифр Гронсфельда в кодуванні: програмування та дослідження».

Об'єкт розробки – шифрування та дешифрування текстів методом Гронсфельда.

Предмет розробки – зручна програма для шифрування та дешифрування текстів методом Гронсфельда.

Головне завдання – проектування і розробка алгоритму для вирішення питань з теми «Шифр Гронсфельда в кодуванні: програмування та дослідження» та написання відповідної програми.

Методи розробки – написання програми для кодування текстів мовою JavaScript із використанням HTML і CSS оболонок.

Новизна роботи – створення онлайн-сторінки з можливістю вирішувати питання з теми «Шифр Гронсфельда в кодуванні: програмування та дослідження». Зручний сервіс, який дозволяє кодувати тексти онлайн, без завантаження сторонніх програм.

Структура пояснювальної записки. Записка складається з 4 розділів. В першому розділі описується постановка задачі, інформаційний розділ ознайомлює з аналогічними до наведеної теми програмами, третій розділ несе інформацію про теоретичні відомості, останній, четвертий розділ відповідає за практичну частину проекту.

Обсяг пояснювальної записки – 53 сторінки.

1. ПОСТАНОВКА ЗАДАЧІ

Мета дипломного проектування – створення зручної програми для вирішення питань з теми «Шифр Гронсфельда в кодуванні: програмування та дослідження».

В зв’язку з цим слід:

1. Ознайомитися з теоретичним матеріалом за темою дипломного проектування;
2. Ознайомитися з програмами, аналогічними визначеній темі;
3. Визначити переваги та недоліки розглянутих програм;
4. Підібрати кілька прикладів для реалізації у бакалаврській роботі;
5. Визначити мову для реалізації проекту та вибір програмного забезпечення;
6. Створити блок-схему алгоритму.
7. Написати та протестувати програму.
8. Описати процес створення програми та надати пояснення до коду програмного продукту.
9. Скласти інструкцію по роботі з програмою.

2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1 Переваги та недоліки інших програм з наведеної теми

Оскільки нас цікавлять лише онлайн сервіси для кодування шифром Гронсфельда, звернемось до Інтернету і спробуємо знайти деякі приклади.

Перший сайт має назву “Інформаційний портал Криптографія” (рис. 2.1) [1].

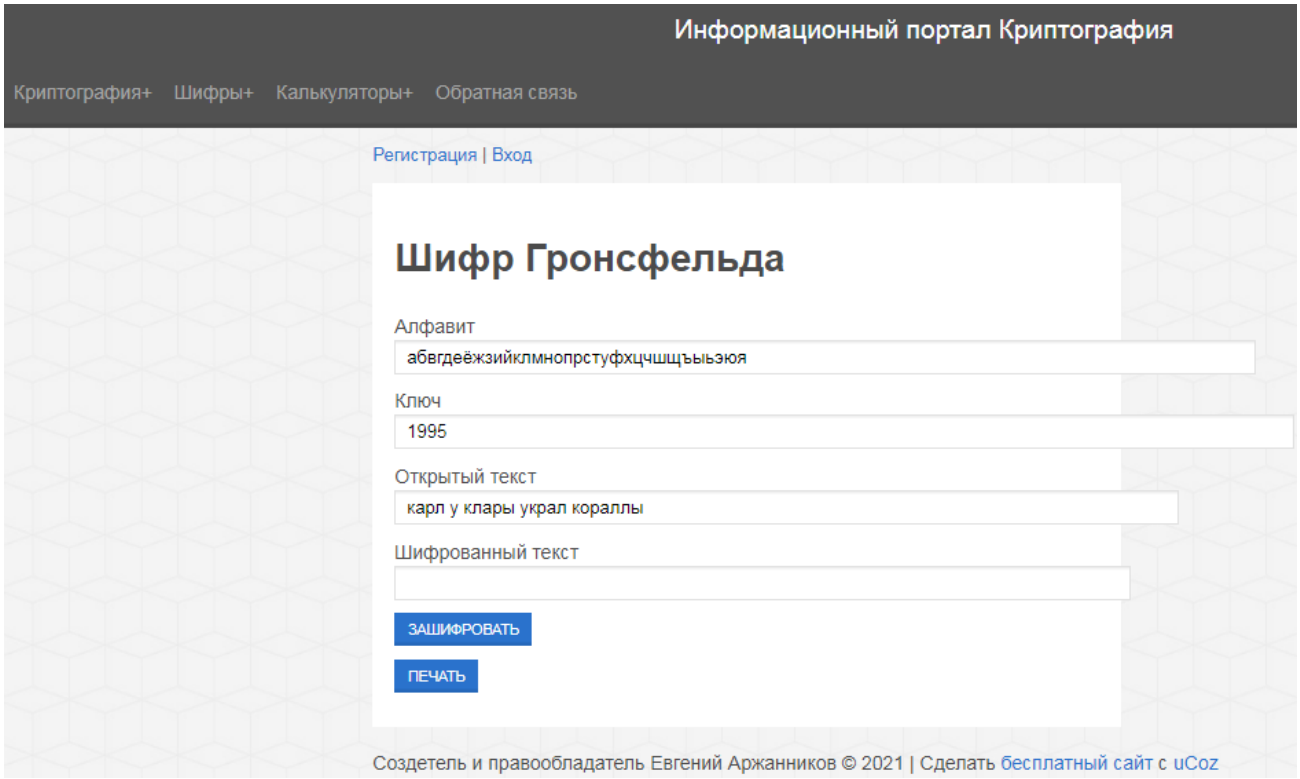


Рис. 2.1 – Сайт “Інформаційний портал Криптографія”

На сторінці, присвяченій кодуванню текстів шифром Гронсфельда, є поле для створення власного алфавіту, з якого буде створюватись шифрування. Поле для ключа та вставки тексту. Нижче знаходиться кнопка зашифрувати.

Головним недоліком представленого сервісу є відсутність можливості дешифрування. Можна лише отримати шифрований текст, а розшифрувати текст – ні.

Наступний сайт, який було оглянуто – “dCode” (рис. 2.2) [2]. На ньому, як і на попередньому сервісі, існує безліч різних сторінок не лише із шифруванням текстів, але й із великою кількістю теоретичного матеріалу. Скориставшись функцією

пошуку на сайті, було знайдено розділ, який присвячений шифру Гронсфельда. На відміну від попереднього сервісу, сайт “d.Code” дає можливість дешифрування текстів. На сторінці є два великих вікна для вставки тексту, одне з яких призначене для шифрування, а інше для дешифрування. Під кожним із них знаходяться поля для ключів, алфавіту та інші допоміжні функції. На відміну від раньше розглянутого нами сайту “Інформаційний портал Криптографія”, сайт “dCode” налічує більше можливостей та додаткових функцій для шифрування і дешифрування.

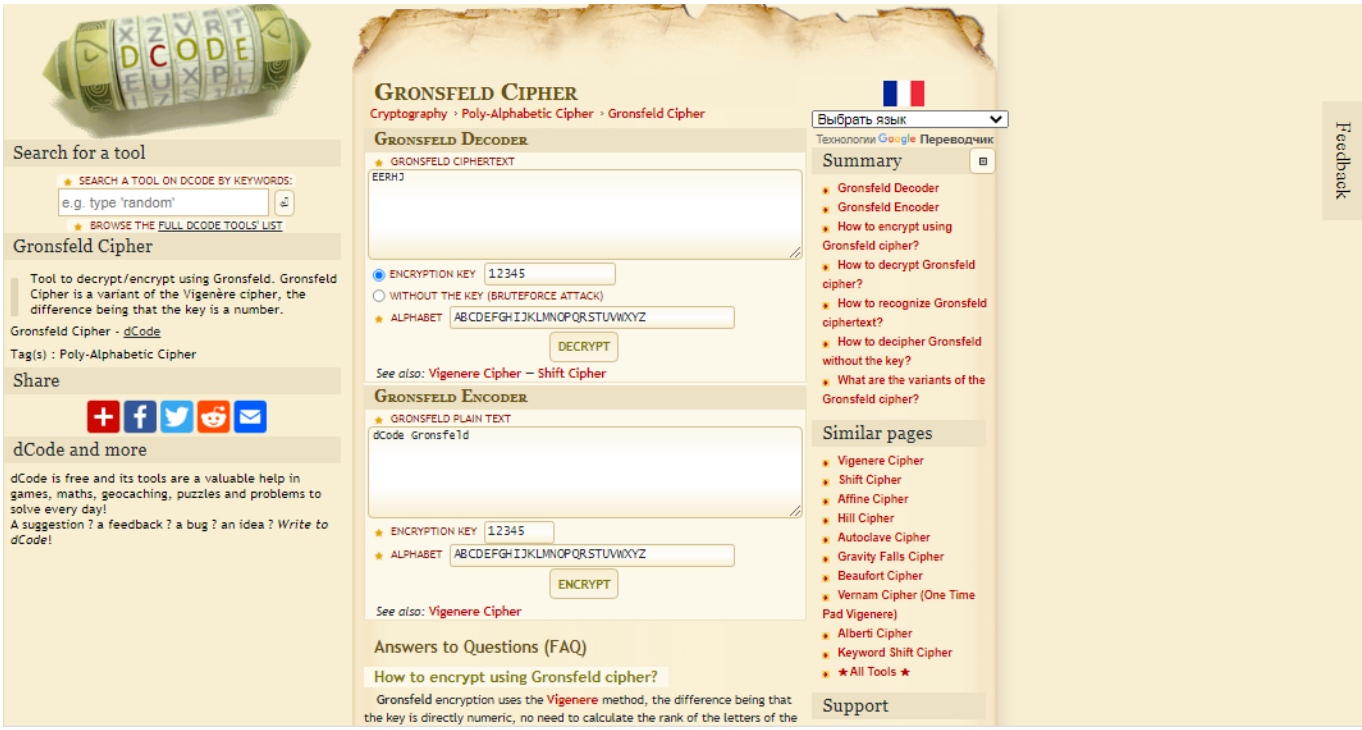


Рис. 2.2 – Сайт “dCode”

Третій сайт називається “Boxentriq” (рис. 2.3) [3].

Представлений сервіс спеціалізується на вирішенні питань кодування. В меню сайта є розділи, які присвячені словниковим програмам, математиці, стенографії і найголовніше – сучасним та класичним шифрам.

Серед наведених у останньому розділі пунктів є сторінка з шифруванням та дешифруванням текстів методом Гронсфельда.

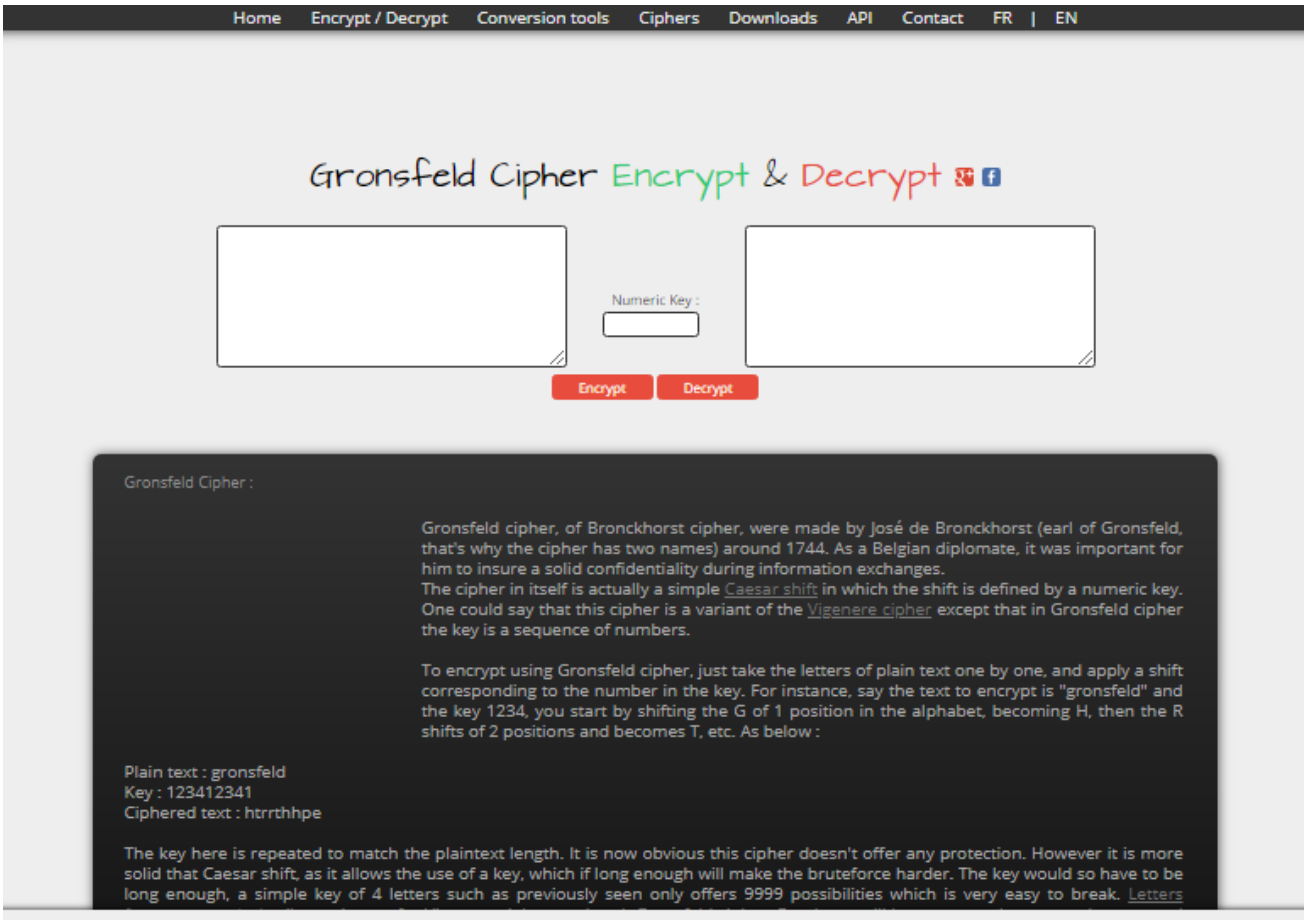


Рис. 2.3 – Сайт “Boxentriq”

В центрі сторінки знаходиться велике поле для вставки тексту. Нижче – три кнопки з функціями копіювання, вставки та налаштування тексту. Ще нижче – поле для введення ключа. Важливим являється короткий опис і принцип роботи самого методу Гронсфельда. Буде корисно та цікаво ознайомитись з основами шифру, який збираємось використовувати. Також особливістю цього сайта є можливість обирати мову тексту, який збираємось зашифрувати або дешифрувати. На відміну від попередніх сервісів, Boxentriq має зручний інтерфейс і не дуже велику кількість реклами.

Четвертий та останній сайт має назву “md5decrypt” (рис. 2.4) [4].

Аналогічно попереднім, цей сервіс присвячений шифруванню та дешифрування різних текстів. Сторінка, яка нас цікавить – Шифрування методом Гронсфельда. Вона має простий інтерфейс, який складається з двох полів для тексту, та поля для ключа між ними.

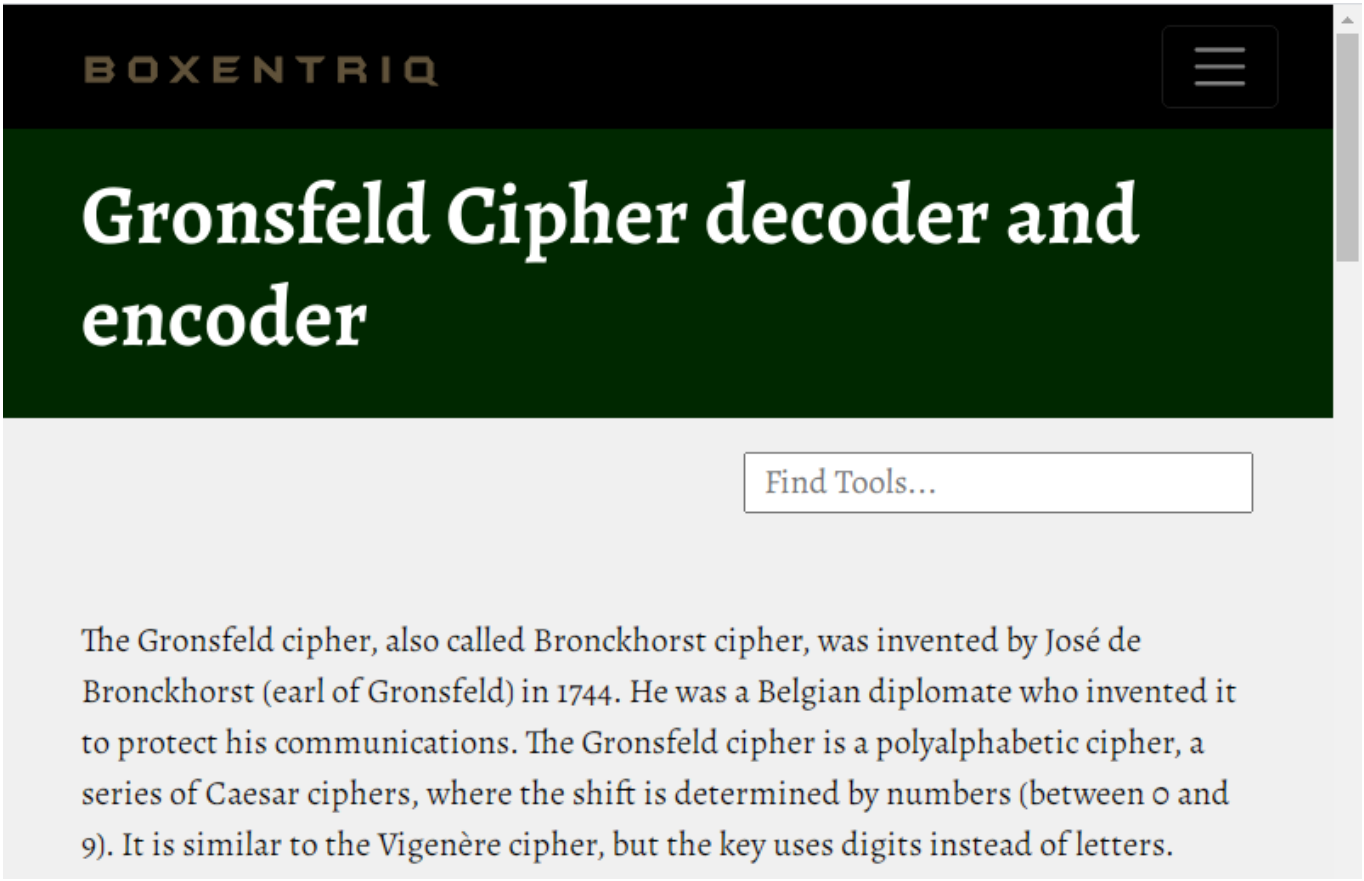


Рис. 2.4 – Сайт “md5decrypt”

На цьому сайті немає можливості вибору мови, але є функція не тільки шифрування, але й дешифрування. Як і на попередньому сайті, нижче знаходиться інформація про сам метод та його особливості. Але великим мінусом є вікна з рекламою, які відволікають від роботи та привертають зайву увагу до себе.

Підводячи підсумки після розглянутих сервісів, можна зробити висновок, що онлайн кодування не є досконалим і йому є куди розвиватись. Деякі з них рясніють зайвою інформацією, а деяким не вистачає навіть елементарних функцій.

Головною схожістю розглянутих сайтів є не дуже зручна реалізація інтерфейсу та складне оформлення, яке іноді відволікає увагу та заважає працювати.

В Інтернеті існує велика кількість інших сайтів, які не встигли розглянути, але більшість із них страждають проблемою не дуже вдалого інтерфейсу та захараченістю. Треба враховувати всі ці фактори задля того, щоб створити зручну сторінку, на якій користувач почував би себе комфортно і також розробити надійний алгоритм, аби кодування шифром Гронсфельда було правильним та технічним. Тому, створюючи сторінку з теми “Шифр Гронсфельда в кодуванні:

програмування та дослідження”, будемо опиратись на зручний інтерфейс сайту “md5decrypt”, а також на справність і широкий спектр функцій сайту “Boxentriq”.

Таким чином розглянуто сайти з можливістю шифрування методом Гронсфельда та визначено їх переваги і недоліки.

2.2 Необхідність та актуальність теми

В наш час не можна уявити своє життя без Інтернету, комп'ютера або інших гаджетів, які об'єднують усіх користувачів всесвітньої павутини. Інтернет - це дуже важливий ресурс інформації для людей. У наші дні в Інтернеті можна знайти будь-яку необхідну інформацію для навчання або для роботи.

Сьогодні дуже складно знайти людину, яка не знайома з Інтернетом. Усілякі інформаційні технології, до яких відноситься і Всесвітня Павутина, міцно увійшли в повсякденне життя. Зараз можна навіть вчитися за допомогою Інтернету, не виходячи з дому. Щодня ряди користувачів мережі поповняються тисячами нових користувачів.

Інтернет - це можливість розвивати власну справу, не дивлячись на кордони. Навряд чи хтось стане заперечувати, що бізнес в мережі відкриває велику кількість можливостей навіть для тих компаній, які функціонують в оффлайн режимі. Всесвітня павутина - не тільки джерело інформації, але і джерело можливості цю інформацію безмежно поширювати.

З огляду на постійну нестачу часу, багато людей поступово звикають до того, що велику кількість дій вони можуть виконати, не відриваючись від монітора: сходити в магазин, "побувати" на прес-конференції, зробити телефонний дзвінок, почитати газети. Звичайно, скептики тут же кинуться засуджувати такий спосіб організації життя: мовляв, скоро люди взагалі перестануть виходити з дому. Однак не можна заперечувати той факт, що можливість економити власний час таким способом дозволяє виділити його для чогось більш важливого: спілкування з родиною, походу в парк, відпочинку з друзями.

Інтернет здатний зробити наше життя комфортним. Важливо тільки не зловживати тими перевагами, які має Всесвітня павутина і раціонально їх використовувати. Інтернет є скарбницею інформації; він містить знання з будь-якої теми. Пошукові системи роблять інформацію доступною. Стало звичайною практикою звертатися за допомогою до Інтернету в рішенні будь-яких питань і пошуку відповідей. В Інтернеті також можна дізнаватися новини про останні досягнення в області медицини, техніки та інших областях науки.

Інтернет став невід'ємним розповсюджувачем знань, як через безкоштовне навчання, так і через надання платних послуг. Довіра до цієї форми навчання і чи є вона безпечною, надійною, як правило, відноситься до кожного сайту окремо. Всесвітня павутина стала чудовою можливістю для академічно непривілейованих людей, накопичити більше знань із потрібних їм предметів. Є також сайти, такі як Вікіпедія, Coursera, Babbel і TeacherTube, які присвятили себе мистецтву передачі знань для всіх людей.

Завдяки численним послугам можна виконати всі фінансові операції в Інтернеті, забронювати квитки на літак, переказати кошти, оплатити комунальні послуги і податки, не залишаючи свої домівки або офіси. Деякі веб-сайти пропонують швидкі схеми бронювання і планування маршрутів відповідно до уподобань своїх клієнтів. Електронна комерція використовується для всіх типів ведення бізнесу, яка включає переказ грошей через Інтернет. Онлайн угоди стали нормою майже всіх видів бізнесу. Електронна комерція, з її величезною досяжністю для різних товарів і послуг, дає можливість доставляти клієнту його замовлення на поріг його будинку. Сайти, такі як eBay дозволяють клієнтам брати участь в торгах, купувати, продавати, і навіть проводити онлайн аукціони.

У світлі останніх подій, через пандемію коронавірусу, багато навчальних закладів перейшли на дистанційну форму навчання.

На сьогоднішній день перспективність розвитку дистанційного навчання набирає обертів не тільки в Європі, але і в Україні. Такий вид отримання знань і навичок дозволяє вирішити багато проблем сучасної освітньої дійсності, а саме дати можливість здобути освіту не виходячи з дому, скоротити час і витрати на навчання.

Дистанційне навчання - це особливий вид отримання знань, в якому взаємодія учня і викладача здійснюється за допомогою інформаційно-комунікаційних технологій. Модернізація освіти дозволяє розширити звичні межі способу передачі і отримання інформації в освітньому процесі, при цьому зберігаючи всі цілісні компоненти.

Варто відзначити позитивні сторони даного процесу: можливість вивчати матеріал в індивідуальному темпі, свобода і гнучкість у виборі запропонованих курсів і факультативів, доступність навчання незалежно від тимчасового і географічного положення, швидкість здійснення зворотного зв'язку між педагогом і учнем, соціальна рівноправність, можливість творчого самовираження, а також сприятливі умови для самостійного отримання знань.

До негативних сторін, як правило, відносять відсутність реального контакту між педагогом і учнем, наявність постійного доступу до комп'ютера та Інтернету, відсутність лабораторних і практичних занять, а також відсутність регулярного контролю за виконанням поставлених завдань.

Безсумнівно, отримання знань за допомогою дистанційного навчання є досить корисним і зручним. Можливо, незабаром, дистанційне навчання стане доповненням до традиційного, але повністю замінити звичний процес отримання знань не зможе.

Підсумовуючи все сказане раніше, можна зробити висновок, що Інтернет налічує безліч корисних можливостей і робить наше життя простішим. Але в ньому, як і у всіх інших речах, є свої недоліки. Тому розглянемо деякі із них.

До найбільшого недоліку Інтернету можна віднести крадіжку особистої інформації. При використанні всесвітньої павутини ви можете зіткнутися з крадіжкою особистої інформації. Наприклад, злочинці можуть заволодіти паролем від вашої електронної пошти, а що ще гірше - номером вашої кредитної картки, допустимо, якщо ви розплачувалися нею в Інтернет магазині.

Вірусна загроза - важливий негативний фактор при використанні глобальної мережі.

Вірус - це програма, яка порушить нормальне функціонування вашого комп'ютера, вірусний напад на ваш комп'ютер може закінчитися поломкою жорсткого диска, що в свою чергу принесе вам багато неприємностей.

Спам відноситься до відправки небажаних повідомлень електронної пошти, які не несуть ніякої мети і засмічують ваш ящик. Такі незаконні дії можуть бути дуже докучаючими.

Спамери зазвичай використовують ботів, які бомбардують приймач з нескінченною лінією рекламних оголошень. Це може виявитися неприємним, так як спам змішується з нашими більш важливими листами. На щастя, постачальники послуг електронної пошти мають системи безпеки для захисту від спаму. Також, можна помітити лист як спам, і тоді всі електронні листи від того ж електронної адреси або IP, будуть блокуватися.

Одна з найбільш дратівливих проблем з Інтернетом є легкість, з якою будь-яка шкідлива програма може заразити наш комп'ютер. Інтернет користувачі часто страждають від вірусних атак, які завдають шкоди їх комп'ютерам і важливим файлам. Вірусні програми можуть непомітно активізуватися, якщо ви просто натиснули, на перший погляд, на нешкідливе посилання. Комп'ютери, підключені до Інтернету, надзвичайно схильні до вірусних атак, які в кінцевому підсумку можуть привести до збою системи. Звертаючи увагу на останній пункт, слід сказати декілька слів про способи, які можуть захистити користувачів від вірусів та витіку особистої інформації.

Шифрування інформації

Шифрування інформації - це її кодування, яке ускладнює або навіть унеможливлює прочитання цієї інформації без відповідної програми або ж апарату. Шифрування є одним з найнадійніших способів захисту інформації.

Шифрування ділиться на: «сильне» і «слабке». «Сильне» шифрування - це таке шифрування, яке практично неможливо вгадати або розкрити без знання пароля. «Слабке» шифрування - ускладнює доступ до інформації, але шифр можна розкрити за певний час без знання пароля, наприклад:

- за допомогою підбору пароля або ключа перебором;
- вгадати пароль;
- підібрати або вгадати пароль, якщо відома його частина;
- за допомогою злому алгоритму шифрування.

Якщо довжина шифру недостатньо велика, його можна розкрити за короткий час, тому він називається «слабким». Суть шифрування полягає в тому, що зашифрована інформація не представляє ніякої цінності без ключа доступу.

Шифрування направлено на досягнення чотирьох основних цілей:

1. Статичний захист інформації, який зберігається на жорсткому диску комп'ютера або дискетах (шифрування файлів, фрагментів файлів або всього дискового простору) виключає або серйозно ускладнює доступ до інформації особам, які не володіють паролем. Тобто захищає дані від стороннього доступу за відсутності власника інформації. Статичне шифрування застосовується з метою інформаційної безпеки на випадок викрадення файлів або жорстких дисків комп'ютерів і виключення можливості прочитання даних будь-якими сторонніми особами, які не володіють паролем. Наприклад, якщо був викрадений жорсткий диск комп'ютера, то за допомогою шифру на ньому зловмисники не зможуть ним скористатися шляхом переміщення його на інший комп'ютер. Шифруватися можуть всі файли, які додаються на зашифрований диск, при їх копіюванні на інший диск, вони автоматично дешифруються.

2. Поділ прав і контроль доступу до даних. Користувач може володіти своїми особистими даними (різними комп'ютерами, фізичними або логічними дисками одного комп'ютера), не доступними іншим користувачам. Прикладом можуть служити звичайні паролі при вході в певну програму або ж список користувачів і права доступу до даної програми.

3. Захист даних, що передаються через треті особи, в тому числі по електронній пошті або в рамках локальної мережі. Наприклад, при передачі листа через електронну пошту можна його захистити за допомогою пароля і підказки до нього, але одержувач повинен знати цей пароль, щоб прочитати зашифрований лист.

4. Ідентифікація дійсності і контроль цілісності переданих через треті особи документів. Наприклад, кожен документ може мати унікальну мітку (ідентифікатор).

Найбільш поширеними методами шифрування в даний час є алгоритми симетричного і асиметричного шифрування. У першому варіанті використовується один і той же ключ, як для шифрування, так і для дешифрування. Класичним

прикладом такого методу можна назвати «блоковий» і «потоковий» алгоритми, які шифрують інформацію, що надходить блоками або в міру надходження відповідно. У асиметричному алгоритмі ситуація дещо інша - тут для шифрування застосовується відкритий ключ, а для процесу дешифрування - закритий, який відомий тільки користувачеві. При цьому, відкритий і закритий ключі створюються, як правило, одночасно і доповнюють один одного.

Існує багато методів та способів для шифрування/дешифрування текстів, задля забезпечення безпеки інформації та даних. Тема дипломної роботи «Шифр Гронсфельда в кодуванні: програмування та дослідження», тому саме на цьому методі буде реалізоване наше програмне забезпечення.

В наш час дуже актуальною темою є збереження персональної інформації в безпеці та підтримці конфіденційності. Багато навчальних закладів перейшли на дистанційну форму навчання, через це онлайн лекції потребують надійності та безпеки.

Проблеми захисту інформації

Інтернет та інформаційна безпека несумісні за своєю природою. Інтернет родився як чисто корпоративна мережа, однак, в даний час за допомогою єдиного стека протоколів TCP / IP та єдиного адресного простору об'єднує не тільки корпоративні та відомчі мережі (освітні, державні, комерційні, військові та ін.), але і рядовим користувачам, які мають можливість отримати прямий доступ до Інтернету від своїх домашніх комп'ютерів за допомогою модемів та телефонної мережі загального користування.

Як відомо, чим простіше доступ у мережу, тим гірше її інформаційна безпека, тому з повною підставою можна сказати, що споконвічна простота доступу в Інтернет - гірше злодійства, тому що користувач може навіть і не дізнатися, що у нього були скопійовані - файли і програми , не кажучи вже про можливість їхнього псування і коректування.

Платою за користування Інтернетом є загальне зниження інформаційної безпеки, тому для запобігання несанкціонованого доступу до своїх комп'ютерів всі корпоративні і відомчі мережі, а також підприємства, що використовують технологію intranet, ставлять фільтри (fire-wall) між внутрішньою мережею і

Інтернетом, що фактично означає вихід з єдиного адресного простору. Ще велику безпеку дасть відхід від протоколу TCP / IP і доступ в Інтернет через шлюзи.

Цей перехід можна здійснювати одночасно з процесом побудови всесвітньої інформаційної мережі загального користування, на базі використання мережевих комп'ютерів, які за допомогою мережевої карти 10Base-T і кабельного модему забезпечують високошвидкісний доступ (10 Мбіт / с) до локального Web-сервера через мережу кабельного телебачення.

Для вирішення цих та інших питань при переході до нової архітектури Інтернету потрібно передбачити наступне:

1. ліквідувати фізичний зв'язок між Інтернетом та корпоративними і відомчими мережами, зберігши між ними лише інформаційний зв'язок через систему World Wide Web;

2. замінити маршрутизатори на комутатори, виключивши обробку в вузлах IP-протоколу і замінивши його на режим трансляції кадрів Інтернет, при якому процес комутації зводиться до простої операції порівняння MAC-адрес;

3. перейти в новий єдиний адресний простір на базі фізичних адрес доступу до середовища передачі (MAC-рівень), прив'язане до географічного розташування мережі і дозволяє в рамках 48-біт створити адреси понад 64 трильйонів незалежних вузлів.

Безпека даних є однією з головних проблем в Інтернеті. З'являються все нові і нові страшні історії про те, як комп'ютерні зловмисники, що використовують усе більш витончені прийоми, потрапляють в чужі бази даних. Зрозуміло, все це не сприяє популярності Інтернету в ділових колах. Одна тільки думка про те, що хулігани або, що ще гірше, конкуренти, зможуть отримати доступ до архівів комерційних даних, змушує керівництво корпорацій відмовлятися від використання відкритих інформаційних систем. Фахівці стверджують, що подібні побоювання безпідставні, тому що в компаній, що мають доступ і до відкритих, і приватних мереж, практично рівні шанси стати жертвами комп'ютерного терору.

Дилема безпеки така: доводиться робити вибір між захищеністю вашого майна і його доступністю для вас, а значить, і можливістю корисного використання.

Це справедливо і щодо інформації. Наприклад, база даних, що містить конфіденційні відомості, лише тоді повністю захищена від зазіхань, коли вона знаходиться на дисках, знятих з комп'ютера і прибраних в охороняєме місце. Як тільки ви встановили ці диски в комп'ютер і почали використовувати, з'являється відразу кілька каналів, по яких зловмисник, в принципі, має можливість отримати до вашим таємниць доступ без вашого відома. Іншими словами, ваша інформація або недоступна для всіх, включаючи і вас, або не захищена на сто відсотків.

В області інформації дилема безпеки формулюється так: слід вибирати між захищеністю системи і її відкритістю. Правильніше, втім, говорити не про вибір, а про баланс, так як система, яка не володіє властивістю відкритості, не може бути використана.

Засоби захисту інформації

Зараз навряд чи комусь треба доводити, що при підключенні до Інтернету Ви піддаєте ризику безпеку Вашої локальної мережі і конфіденційність міститься в ній інформації. За даними CERT Coordination Center в 1999 році було зареєстровано 2421 інцидентів - зломів локальних мереж і серверів. За результатами опитування, проведеного Computer Security Institute (CSI) серед 500 найбільш великих організацій, компаній і університетів з 1995 число незаконних вторгнень зросло на 48.9%, а втрати, викликані цими атаками, оцінюються в 66 млн. Доларів США.

Одним з найбільш поширених механізмів захисту від Інтернет бандитів - "хакерів" є застосування міжмережевих екранів - брендмауерів (firewalls).

Варто відзначити, що в слідстві непрофесіоналізму адміністраторів і недоліків деяких типів брендмауерів близько 30% зломів відбувається після встановлення захисних систем.

Інформаційна безпека в Інтернеті

Архітектура Інтернету має на увазі підключення до зовнішніх відкритих мереж, використання зовнішніх сервісів і надання власних сервісів зовні, що висуває підвищені вимоги до захисту інформації.

В Інтернет-системах використовується підхід клієнт-сервер, а головна роль на сьогоднішній день приділяється Web-сервісу. Web-сервери повинні підтримувати традиційні захисні засоби, такі як аутентифікації і розмежування доступу; крім того,

необхідне забезпечення нових властивостей, особливо безпеки програмного середовища і на серверної, і на клієнтської сторонах.

У еволюціях технологій захисту можна виділити три основні напрями.

Перше - розробка стандартів, які імплементують в мережу певні засоби захисту, перш за все адміністративної. Прикладом є IP security option і варіанти протоколів сімейства TCP / IP, що використовуються в Міністерстві оборони США.

Другий напрямок - це культура міжмережевих екранів (firewalls), давно застосовуваних для регулювання доступу до підмереж.

Третє, найбільш молодий напрямок - це так звані технології віртуальних захищених мереж (VPN, virtual private network, або intranet).

2.3 Мова для реалізації проекту

У наш час IT-індустрія продовжує розвиватися. З'являються нові мови, покращуються старі [5].

Python

Python - безкоштовна мова програмування з відкритим вихідним кодом і зручними структурами даних. Він запускається на будь-яких ОС і підтримує безліч сервісів, середовищ розробки і фреймворків. До того ж він підходить для новачків і його просто вивчити.

Python підходить для створення веб-сервісів і мобільних додатків, на зразок YouTube, Quora, Pinterest та Instagram, а також програм Blender, Inkscape і Autodesk. Крім того, Python використовували для створення відеоігор, включаючи Civilization IV і Vegas Trike.

Java

Java - одна із самих практичних мов програмування для вивчення. Її популярність не можна переоцінити, оскільки більшість (90%) компаній зі списку Fortune використовують Java для розробки бекенда-систем і десктопних додатків. Кросплатформеність досягнута завдяки JVM.

Java - став стандартом для додатків, які запускаються на будь-яких платформах, включаючи Mac, Windows, Android, iOS і так далі. Також його використовують в системах великих даних.

На Java написані веб-додатки великих компаній, таких як Twitter, LinkedIn, Amazon і eBay. Він також є офіційною мовою для створення додатків на Android.

Javascript

JavaScript - це одна з основ фронтенд-розробки. Її використовують, щоб зробити сайти інтерактивними: додавати спливаючі вікна, ефекти і навіть невеликі ігри.

Крім того, з випуском ECMAScript 6 і таких фреймворків, як Angular, Node, Express і React, розробники почали використовувати JavaScript для створення клієнтського і серверного програмного забезпечення.

За статистикою, зібраною сайтом Stackoverflow (Stack Overflow Developer Survey), який охопив аудиторію складом більше, ніж 64000 розробників з 173 країн, JavaScript стала найбільш використовуваною мовою програмування в світі. Вона надає можливість створювати інтерактивні сайти і є однією з основних веб-технологій поряд з HTML і CSS, оскільки більшість браузерів в тому чи іншому вигляді можуть використовувати JS.

JavaScript відмінна мова, щоб почати свій шлях в веб-розробці. Вона підійде для створення інтерактивних сайтів в Інтернеті. Останнім часом JavaScript розширився і тепер на ньому можна писати мобільні додатки, ігри, а також десктопні програми. Це виразно вплинуло на популярність мови.

C #

C # - об'єктно-орієнтована мова програмування, розроблена Microsoft. Це одна з найпотужніших мов для платформи .NET Framework.

C # часто використовують бекенд-розробники, в тому числі в Bing, Dell, Visual Studio і MarketWatch., Розробники ігор на Unity, творці додатків для Windows, Android і iOS.

Крім того, вона є рекомендованою мовою для розробки ігор з використанням Unity Game.

Сі і С ++

Сі був розроблений ще в 1973 році і до цих пір залишається однією з найпоширеніших мов програмування. С ++ дуже близький до Сі: у них схожий синтаксис, до того ж велика частина коду, написана Сі, буде справедлива і для С ++.

На Сі і С ++ написані Microsoft Windows, Linux, macOS, ядра iOS і Android. А також Oracle Database, MySQL і MS SQL Server.

Сі і С ++ вважаються високопродуктивними мовами. Тому їх використовують в розробці додатків, для яких важлива продуктивність. Це, наприклад, Firefox, програмами Adobe, а також відеоігри.

Знання С ++ дозволить з легкістю писати ігри і складні комерційні системи поряд з простими додатками. Вона є однією з найпотужніших мов програмування, який надає чимало корисних функцій.

PHP

PHP - одна з найпопулярніших мов програмування для бекенд. Її використовують для створення багатьох сайтів в Інтернеті, включаючи Facebook і Yahoo.

PHP вважається відносно доступною мовою для початківців програмістів. До того ж у PHP-розробників є багато спеціалізованих онлайн-спільнот, де можна отримати відповіді на будь-які питання.

PHP використовується повсюдно завдяки Wordpress. 80% сайтів з відвідуваністю понад 10 млн. користувачів використовують PHP. Прикладами таких сайтів можуть стати Facebook і Wikipedia. У PHP не існує яких-небудь строгих правил в написанні коду, а також він гнучкий у вирішенні різних проблем. PHP - це відмінний вибір для веб-розробників, оскільки він є серверним скриптовою мовою і для Worldpress, і для Facebook.

У більшості випадків вивчення PHP для веб-розробника є запорукою успіху, так як його знання дозволяє вам створювати приголомшливі динамічні веб-сайти. Ви можете використовувати PHP для різних веб-проектів. Це досить проста мова з відкритим вихідним кодом, хорошою підтримкою багатьох баз даних, а також численними інструментами і різними напрямками для використання.

R

Програми, написані на R, використовують великі компанії для аналізу статистики та обробки і структурованих і неструктурованих даних. Також вона підходить для машинного навчання.

R досить складно вивчити, але у неї активна онлайн-спільнота, яка допомагає новачкам.

Objective-C

Objective-C - використовують для створення програмного забезпечення OS X і iOS з початку 1980-х років. Вона досить гнучка, з простим синтаксисом і її легко освоїти. Особливо, якщо ви знайомі з C і Java.

У 2014 році на заміну Objective-C Apple представила Swift. Втім, Objective-C досі затребувана і краще підходить для великих проектів.

Swift

Swift - одна з найпопулярніших мов програмування для розробки додатків на iOS. У неї відкритий вихідний код і простий синтаксис, до того ж, Swift сумісна з Objective-C.

За останні роки Swift стала більш популярна, ніж Objective-C. Це мова програмування для розробки нативних додатків для iOS або Mac OS. Також можна сказати, що це мова програмування з найбільшим потенціалом для зміни майбутнього. Було виявлено, що нативні додатки перевершують крос-платформні гібридні програми, а движок SpriteKit при цьому спрощує створення 2D-ігор. На ділі Swift спирається на успіхи C і Objective-C, але при цьому без обмежень сумісності.

Велику роль в становленні Swift вплинуло на нього таких мов програмування, як Ruby і Python. Вона вважається зручноюдля користувача і цікава у використанні. Swift - це високорівнева мультипарадигмальна мова, розроблена Apple для iOS. Якщо робота з продуктами Apple є вашою метою, то це мова для вас. Swift - статично типізований мова. Це означає, що Xcode перевіряє ваші помилки за вас, тому їх легше відстежувати.

На Swift написані популярні сервіси, такі як WordPress, Mozilla Firefox, SoundCloud і Flappy Bird.

Go

Go - мова 2009 року - епохи багатоядерних процесорів, тоді як мови на зразок Python і Java з'явилися в роки одно поточного середовища розробки. Саме тому мова Go враховує багатозадачність і працює відповідно до неї. Замість всім відомих потоків (Thread), які у більшості мов займають багато пам'яті (наприклад, в Java це 1 Мб на кожен потік), в Go передбачені горутіни, «з'їдають» усього 2 КБ пам'яті. Можна створити хоч тисячу або мільйон горутин, і це практично не позначиться на роботі програми.

Швидкий час запуску, використання пам'яті, тільки якщо це необхідно (сегментовані, але розгортаються стеки горутини), і інші переваги роблять Go надзвичайно затребуваним в рішенні багатопоточних завдань. Це без перебільшень серверна мова майбутнього, і в 2021 вона точно не здасть свої позиції.

Kotlin

Kotlin стрімко розвивається і має низку переваг. Серед них:

- лаконічність мови програмування;
- сумісність з Java;
- підтримується Google.

Цілком можливо, що скоро додатки для Android-девайсів будуть писатися виключно на Kotlin.

Rust

У 2016 році Rust зайняв перше місце в опитуванні "що розробники люблять найбільше" на Stack Overflow. Rust виявився мовою програмування, яку розробники дійсно цінують (79.1% голосів). Мова розробки з відкритим вихідним кодом, розроблена Mozilla Foundation, працює як низькорівнева.

Враховуючи все вищесказане необхідно ще раз чітко визначити особливості та фактори, які впливають на зручність користуванням того чи іншого сайту в Інтернеті. Складаючи цей список, будемо опиратись на аргументи, які навели раніше:

– в Інтернеті дуже багато інформації, тому наша сторінка має носити лише необхідні дані. В нашому випадку це кодування текстів шифром Гронсфельда;

– на сторінці не може бути жодної реклами або інформації, яка могла б відволікти користувача;

– проект буде написаний мовою JavaScript та реалізований за допомогою HTML та CSS, аби користувачі не хвилювалися через шкоду їх комп’ютеру за допомогою різних програм.

3. ТЕОРЕТИЧНА ЧАСТИНА

3.1 Теоретичні відомості з теми «Шифр Гронсфельда в кодуванні»

Перш ніж перейти до самого методу Гронсфельда, необхідно розібратися із загальними поняттями кодування та шифрування.

Коди і шифри - не одне й те саме: в коді кожне слово замінюється іншим, в той час як в шифрі замінюються всі символи повідомлення.

Стеганографія

Стеганографія - це метод організації зв'язку, який власне приховує само наявність зв'язку. На відміну від криптографії, де ворог точно може визначити чи є передане повідомлення зашифрованим текстом, методи стеганографії дозволяють вбудовувати секретні повідомлення в нешкідливі послання так, щоб неможливо було запідозрити існування вбудованого таємного послання.

Слово "стеганографія" в перекладі з грецької буквально означає "тайнопис" (steganos - секрет, таємниця; graphy - запис). До неї належить величезна безліч секретних засобів зв'язку, таких як невидиме чорнило, мікрофотознімкі, умовне розташування знаків, таємні канали та засоби зв'язку на плаваючих частотах тощо.

Найкраще для стеганографії підходять повсякденні об'єкти. Колись в Англії використовувався такий метод: під деякими буквами на першій сторінці газети стояли крихітні точки, майже невидимі неозброєним оком. Якщо читати тільки помічені букви, то вийде секретне повідомлення.

Деякі писали повідомлення першими літерами складових його слів чи використовували невидиме чорнило. Була поширена практика зменшення цілих сторінок тексту до розміру буквально одного пікселя, так що їх було легко пропустити при читанні чогось відносно нешкідливого.

Стеганографія займає свою нішу в забезпеченні безпеки: вона не замінює, а доповнює криптографію. Приховування повідомлення методами

стеганографії значно знижує ймовірність виявлення самого факту передачі повідомлення. А якщо це повідомлення до того ж зашифровано, то воно має ще один, додатковий, рівень захисту.

В даний час у зв'язку з бурхливим розвитком обчислювальної техніки і нових каналів передачі інформації з'явилися нові стеганографічні методи, в основі яких лежать особливості подання інформації в комп'ютерних файлах, обчислювальних мережах тощо. Це дає нам можливість говорити про становлення нового напряму - комп'ютерної стеганографії.

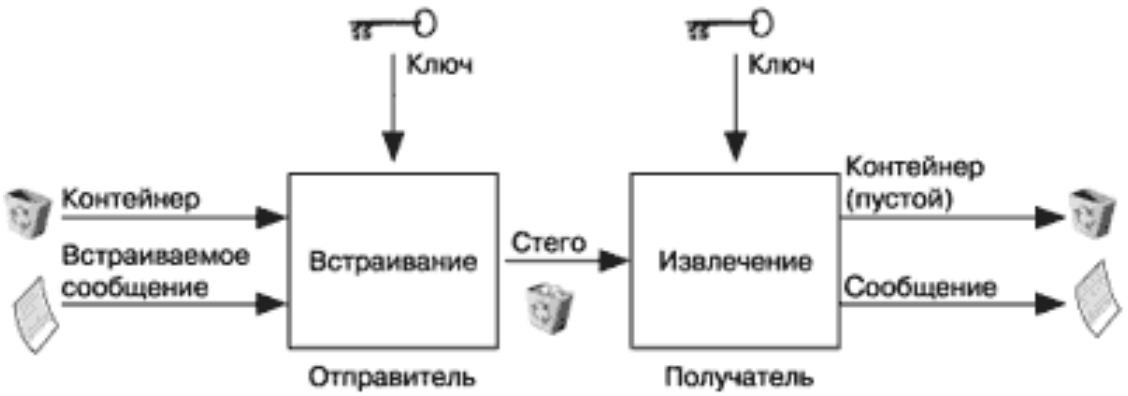


Рис. 3.1 – Схема стеганографії

Транспозиція

У транспозуючих шифрах букви переставляються за заздалегідь визначеним правилом. Наприклад, якщо кожне слово пишеться задом наперед, то з «all the better to see you with» виходить «lla eht retteb ot ees joy htiw». Інший приклад - міняти місцями кожні дві букви. Таким чином, попереднє повідомлення стане «la tl eh eb tt re ot es ye uo iw ht».

Подібні шифри використовувалися в Першу Світову і Американську Громадянську Війну, щоб посилати важливі повідомлення. Складні ключі можуть зробити такий шифр досить складним на перший погляд, але багато повідомлень, закодованих подібним чином, можуть бути розшифровані простим перебором ключів на комп'ютері.

Азбука Морзе

Всі телеграфи використовують так званий телеграфний код - прийняту умовну систему позначень, в якій кожній літері (або знаку) відповідає своя комбінація елементарних посилок електричного струму. Елементарна посилка (елемент коду) - найліпша, з них складаються всі інші. Кількість елементарних посилок для позначення кожного знака в коді може бути різним (нерівномірні коди, наприклад, код Морзе) або однаковим (рівномірні коди, наприклад, код Бодо). Число значень, яке може набувати елементарна посилка в процесі передачі, називається підставою коду - за цією ознакою коди діляться на бінарні (двійкові), потрійні і якісь ще. Залежно від числа елементарних посилок для передачі знаків розрізняють рівномірні коди 5-елементні, 6-елементні і так далі.

«Код Морзе» (він же «морзянка») - ні що інше, як нерівномірний телеграфний код, в якому знаки позначаються комбінаціями з посилок струму різної тривалості (рис. 3.2). За одиницю тривалості приймається тривалість точки, а тривалість «тире» дорівнює тривалості трьох крапок. Пауза між знаками в букві - одна точка, а між буквами в слові - 3 точки. Пауза між словами становить 7 точок. Всі знаки в коді Морзе утворюють так звану азбуку Морзе:

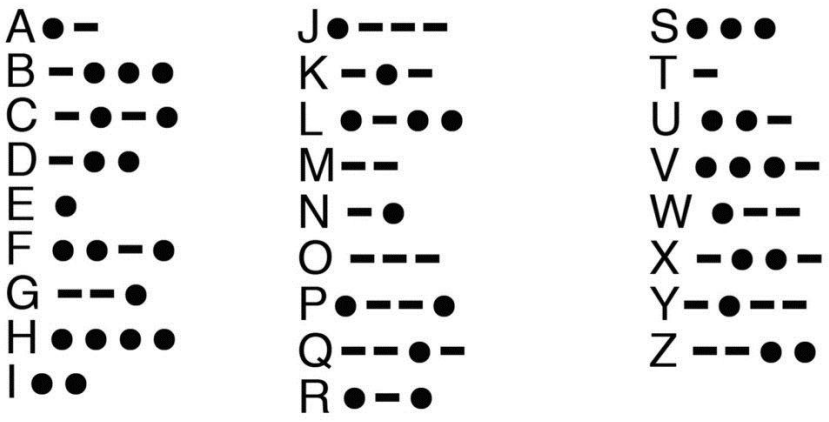


Рис. 3.2 – Азбука Морзе

Примітно, що вихідна таблиця «коду Морзе» відчутно відрізнялася від тих кодів, що сьогодні звучать на аматорських діапазонах. По-перше, в ній використовувалися посилки трьох різних тривалостей (крапка, тире і довге тире). По-друге, деякі символи всередині своїх кодів мали паузи. Кодування сучасної й вихідної таблиць збігаються тільки для половини букв (A, B, D, E, G, H, I, K, M, N, S, T, U, V і W) і не збігаються ні для однієї цифри. Більш того, для побудови коду деяких символів в оригінальній «морзянці» взагалі використовувалися інші принципи. Так, крім «точок» і «тире», були поєднання "подвійне тире» (буква L) і навіть «потрійне тире» (цифра 0), а деякі символи включали в себе паузу. Латинська літера C, наприклад, передавалася раніше як «дві точки-пауза-точка», тобто як передані послідовно букви I та E - подібні нюанси помітно ускладнювали прийом радіограм. У тому числі з цієї причини незабаром з'явилися різні варіанти телеграфної абетки, що не містили кодів з паузами між посилками (Філліпса, Бална, «морський», «континентальний» тощо) (рис. 3.3).

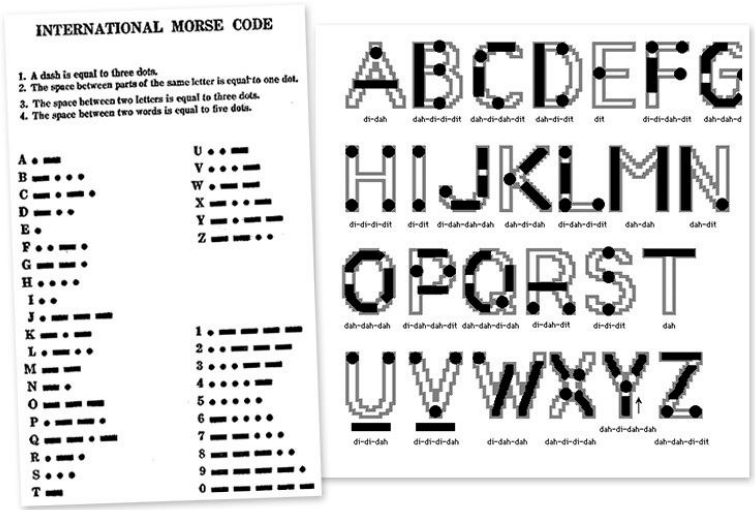


Рис. 3.3 – Міжнародний код Морзе

В азбуці Морзе кожна буква алфавіту, всі цифри і найбільш важливі знаки пунктуації мають свій код, що складається з низки коротких і довгих сигналів, що їх називають «точками і тире». Так, А - це «• -», В - «- •••», і т.д. На відміну від більшості шифрів, азбука Морзе використовується не для утруднення читання повідомлень, а навпаки, для полегшення їх передачі (за допомогою телеграфу).

Довгі й короткі сигнали посилаються за допомогою включення і виключення електричного струму.

Телеграф і азбука Морзе назавжди змінили світ, зробивши можливою блискавичну передачу інформації між різними країнами, а також сильно вплинули на стратегію ведення війни, адже тепер можна було можна здійснювати майже миттєву комунікацію між військами.

Шифр Цезаря

Шифр Цезаря називається так, тому що його використовував сам Юлій Цезар. Насправді шифр Цезаря - це не один шифр, а цілих двадцять шість, що використовують один і той же принцип. Так, ROT1 - всього один з них. Одержувачу потрібно сказати, який з шифрів використовується. Якщо використовується шифр «G», тоді А замінюється на G, В на H, С на I тощо. Якщо використовується шифр «Y», тоді А замінюється на Y, В на Z, С на A тощо.

На шифрі Цезаря базується величезна кількість інших, більш складних кодів і шифрів, але сам по собі він не представляє з себе інтересу через легкість дешифрування. Перебір 26 можливих ключів не займе багато часу. Li bra ghflskhu wklv dqg bra nqrz lw, fods brxu kdqgv.

Шифр Віженера

Цей шифр складніший, ніж моноалфавітний. Уявімо, що у нас є таблиця, побудована за тим же принципом, що і наведена вище, і ключове слово, припустимо, «CHAIR». Шифр Віженера використовує той же принцип, що і шифр Цезаря, за тим винятком, що кожна буква змінюється

відповідно до кодовим словом. У нашому випадку перша буква послання буде зашифрована згідно шифрування алфавітом для першої літери кодового слова (в нашому випадку «С»), друга буква - відповідно до алфавіту для другої літери кодового слова («Н»), і так далі.

У разі, якщо послання довше кодового слова, то для (k * n + 1) -ої літери (де n - це довжина кодового слова) знову буде використаний алфавіт для першої літери кодового слова, і так далі. Дуже довгий час шифр Віженера вважався таким, що його неможливо взламати. Щоб його розшифрувати, для початку вгадують довжину кодового слова і застосовують частотний аналіз до кожної n-ної букві послання, де n - передбачувана довжина кодового слова. Якщо довжина була вгадана вірно, то і сам шифр розкриється з більшою або меншою часткою ймовірності.

Якщо передбачувана довжина не дає вірних результатів, то пробують іншу довжину кодового слова, і так далі до переможного кінця.

Для зашифрування можна використовувати таблицю алфавітів, звана таблиця прямокутник або квадрат (таблиця) Віженера (рис.3.4). Якщо на підставі взяти латинський алфавіт, для таблиці Віженер складатиметься з рядків по 26 символів, котрий кожна наступна рядок з'являється на 1 позиції. Таким чином, у таблиці вийде 26 різних шифрів Цезаря.

Шифр Гронсфельда

Шифр Гронсфельда, був виготовлений Хосе де Бронкхорстом (граф Гронсфельд, тому шифр має дві назви) близько 1744 року. Як бельгійський дипломат, йому було важливо забезпечити конфіденційність під час обміну інформацією.

Сам по собі шифр насправді є простим зсувом Цезаря, в якому зсув визначається цифровою клавішею. Можна сказати, що цей шифр є різновидом шифру Віженера, за винятком того, що в шифрі Гронсфельда ключ - це послідовність чисел.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	D
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	C
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	B
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Рис. 3.4 – Таблиця Віженера

Щоб зашифрувати за допомогою шифру Гронсфельда, просто візьміть букви простого тексту по черзі та застосуйте зсув, що відповідає номеру в ключі. Наприклад, скажімо, що текст для шифрування - "gronsfeld", а клавіша 1234 починається зі зміщення G на 1 позицію в алфавіті, стає H, потім R зміщується на 2 позиції і стає T тощо.

Звичайний текст: gronsfeld

Ключ: 123412341

Зашифрований текст: htrrthhpe

Ключ тут повторюється, щоб відповідати довжині відкритого тексту. Таким чином, ключ повинен бути достатньо довгим, простий ключ із 4 літер, такий, як вище, пропонує лише 9999 можливостей, які дуже легко зламати. Аналіз частоти літер дозволяє, як і для шифру Віженера, зламати шифр Гронсфельда. Для цього вам доведеться отримати досить довгий зашифрований текст, а потім з’ясувати повторення в ньому. Ці повторення можуть бути результатом того, що однакові літери шифруються однією і тією ж частиною ключа. Наприклад, слово "зашифровано цифрами" 12. Аналіз частоти також дозволяє вгадати довжину ключа. Якщо у вас достатньо довгий текст, ви можете, знайшовши всі ці повторення, об’єднати їх та проаналізувати результат, щоб відгадати букви.

3.2. Приклади кодування методом Гронсфельда

Приклад 1. Припустимо, необхідно зашифрувати слово «Ukraine», використовуючи ключ «4789351». Записуємо під словом Ukraine наш ключ, після чого зрушуємо за алфавітом кожну букву на стільки букв вперед, скільки вказано нижче.

U	k	r	a	i	n	e
4	7	8	9	3	5	1
Y	r	z	j	l	s	f

Приклад 2. Запишемо англійське речення: Sweet Home. Яке складається із 9 літер і одного пропуску. Всього маємо 10 символів. Ключем стане число 2475, котре необхідно повторити 2,5 рази, аби його кількість дорівнювала кількості літер.

Повідомлення	S	w	e	e	t		H	o	m	e
Ключ	2	4	7	5	2	4	7	5	2	4
Шифрування	U	a	l	j	v		J	s	t	j

Приклад 3. Наступні приклади спробуємо зашифрувати українською мовою. Візьмемо прізвище українського письменника Івана Петровича Котляревського і зашифрує мого його користуючись ключем «1368», який необхідно буде повторити до тих пір, поки кількість цифр ключа не буде дорівнювати кількості літер. Отримаємо наступний результат:

К	о	т	л	я	р	е	в	с	ь	к	и	й
1	3	6	8	1	3	6	8	1	3	6	8	1
Л	с	ш	у	а	у	Ї	и	т	а	р	о	к

Приклад 4. До цього моменту були наведені зразки шифрування коротких слів та словосполучень. В останньому прикладі спробуємо зашифрувати невеликий текст про місто Полтава.

Текст, який будемо шифрувати*:* *Полтава – одне з найдавніших руських міст, засноване сіверянами у IX ст. задля оборони Русі від кочівників.*

Згідно з першою літописною згадкою міста, у давньоруському Іпатіївському літописі, під назвою «Лтава» сучасна Полтава походить від найменування річки Лтава, правої притоки Ворскли (згодом похідне «По-Лтава» себто По Лтаві) що текла Мазурівським яром на Поділ. Назві приписують слов'янське походження.

На початку XX століття щодо міста також вживали термін «Духовний центр України», чому сприяли діяльність цілої плеяди видатних діячів культури і мистецтва, значні церковні та історичні пам'ятки. Окрім цього Полтава була найбільшим центром розвитку української культури того часу.

Ключ для шифрування:

765329487628965734286871264861374253423462347862348652394623478523487231462738945236438753908127482364782354267584319765124623472026423854236845236403571026487849394830265748371633947578928347463728 72.

Ключ складається із 200 цифр, які слід повторити тричі, аби довжина ключа дорівнювала кількості символів в тексті.

Результат шифрування: *Цфрхвіґ – ціуж н цеоїгеппбоя схчбтмц похф, кгхпсеепз хоиїтвсжтл х IX фю. йеєог хзутссо Чхфї енє ссвретїнне.*

Їіоирч з чєтвте нкштцоурув ізєиттб нршшд, ф єґжпатчхсяртоц Пфґфкоихвмстч лкчхриун, упї хґодчв «Ухаґе» цьюжффб Хсоюґзд ццбрййцґ енж фвсужфщєгпцг шоамо Фшдзг, утжжцп ртмцирі Дхфупок (іетірп уханєри «Чф-Рфгіґ» чжґцх Чу Нхґио) юс уирнє Пжочхйдчбноу ґучм хб Схзпн. Рейзп сулуїчьгьб фмчз'дттяої ссцхєжжусб.

Рж фтщгштч XX цфсслтхґ гпдр тлщщж цзнчї иіиґерн цкууїу «Жцбтзтнс вжххф Ьоцгппо», брущ цттпгун ійєфгтофця бнухй ссиєіі днзвчркч жлдщке сюсяхчшм м ойьціїшхеє, нтвьсп ажултжпо хж рхчрумюрп цдп'жтті. Рсфпо щгтиц Ссрцвжє еюпг озрєммяямо щиффррт фрїилцмц щтфдкрчбксн сфляшчшн ьтїс гґщц.

3.3. Алгоритм створення програми

Створення програмного забезпечення з теми «Шифр Гронсфельда в кодуванні: програмування та дослідження» потребує чітко визначених кроків дій, аби остаточний результат відрізнявся оригінальністю, технічністю, справністю, зручнітю та достовірністю.

Шифрування представляє собою процес введення тексту в поле, вставлення цифрового ключа і отримання готового результату у вигляді зашифрованого тексту. Протилежною процедурою є процес дешифрування, який працює навпаки. Зашифрований текст зіставляється з ключем і отримується готовий текст повідомлення.

Онлайн-сторінка повинна містити в собі поля для введення тексту, поле для вставлення ключа і кнопок, для вибору бажаної операції, тобто шифрування або дешифрування.

Макет для сторінки буде створений за допомогою HTML. На цьому кроці буде створена основна розмітка та додана інформація про призначення сайту.

Наступне – оформлення дизайну сторінки при використанні CSS. Даний етап додасть кольори сторінці, визначить розміри всіх полів та кнопок.

Останнє – сама програма шифрування методом Гронсфельда, яка буду реалізована мовою програмування Javascript.

Отже, визначивши основні етапи створення програмного забезпечення на тему «Шифр Гронсфельда в кодуванні: програмування та дослідження» можна чітко зіставити послідовність дій:

Крок 1. Ознайомлення з методом Гронсфельда та інформативна підготовка.

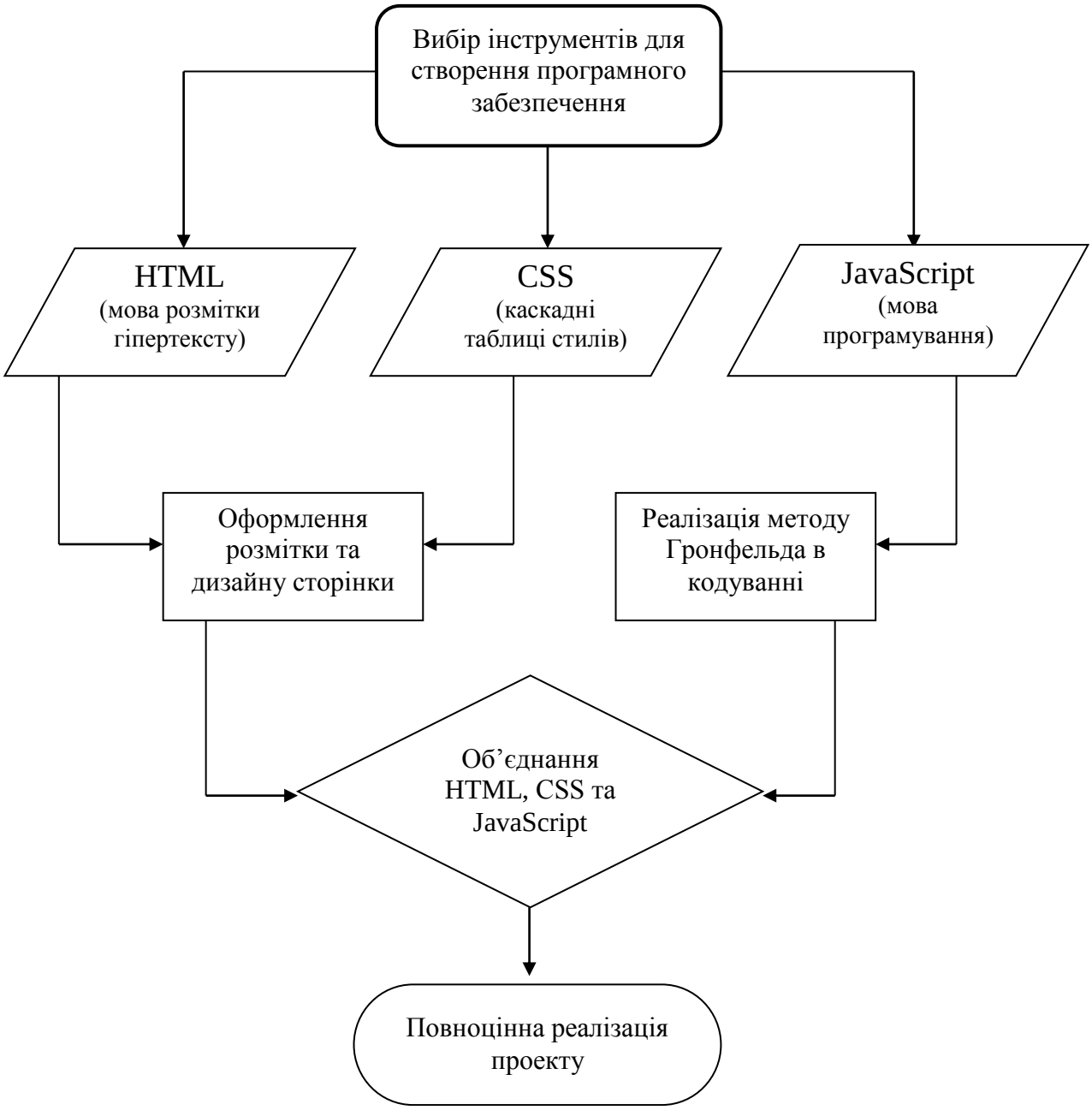
Крок 2. Створення макету сторінки.

Крок 3. Дизайнерське оформлення сайту.

Крок 4. Реалізація програмного забезпечення.

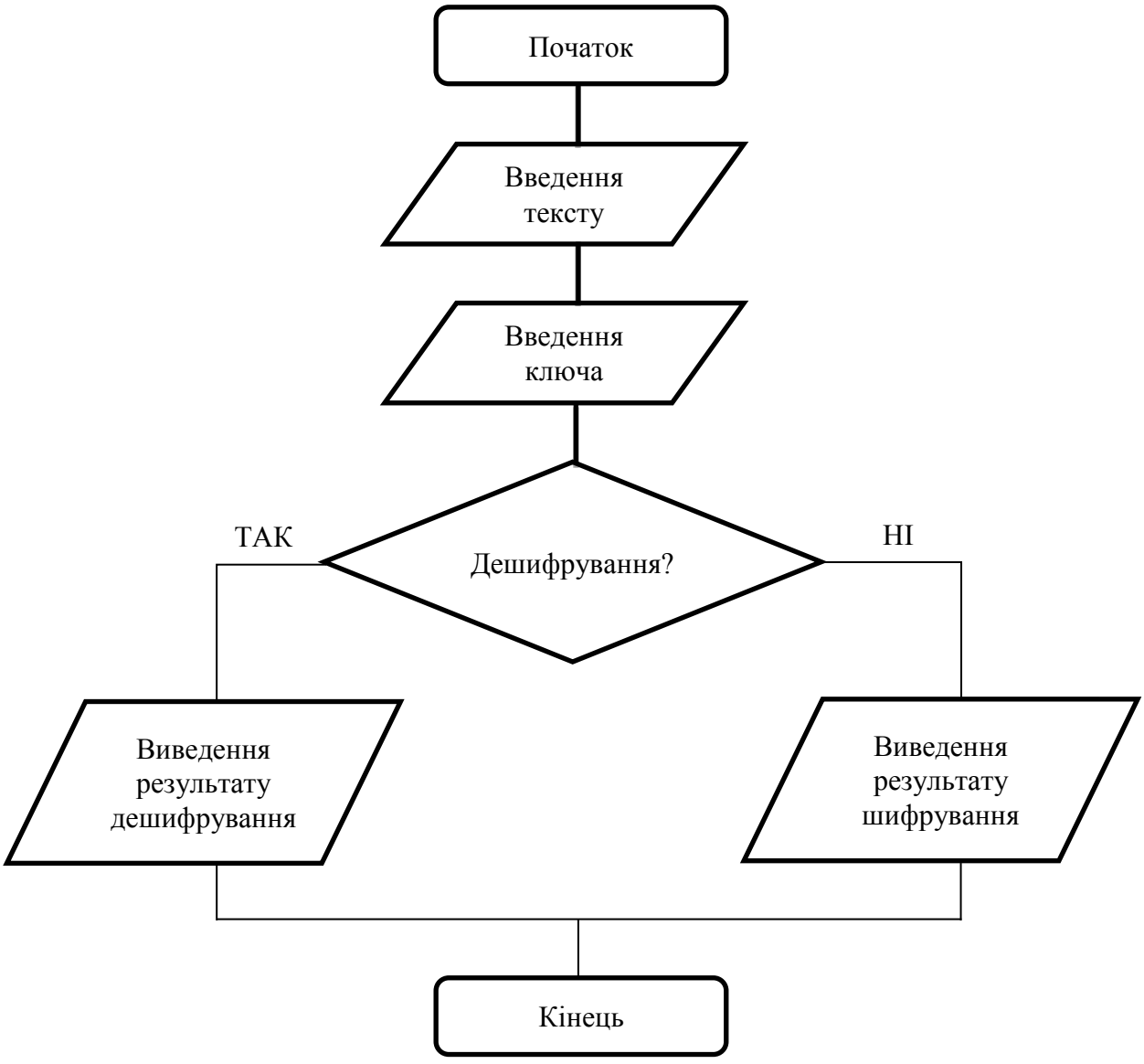
3.4. Блок-схема для створення програми

Для детальнішого огляду створення програмного забезпечення з теми «Шифр Гронсфельда в кодуванні: програмування та дослідження» складемо блок-схему на відповідну тему:



Наступним кроком є створення блок-схеми з описом принципу роботи самої програми.

Робота програмного забезпечення розпочинається з введення повідомлення для шифрування або дешифрування. Далі вводиться ключ та отримується результат.



4. ПРАКТИЧНА ЧАСТИНА

4.1 Обґрунтування вибору програмних засобів

Для розробки програмного забезпечення з теми “Кодування текстів шифром Гронсфельда” мною була обрана мова програмування JavaScript.

JavaScript - це мова програмування, що є прототипно-орієнтованою. Вона відображає мову ECMAScript, чиїм прототипом спочатку і була. Перша варіація з'явилася ще в 1995 році і з тих пір постійно вдосконалювалася, поки не прийшла до нинішнього вигляду.

Найчастіше ця мова використовується в розробці додатків і браузерах з метою надання їм інтерактивності і «жвавості».

JavaScript - це кращий друг HTML і CSS. HTML задає розмітку сайту, CSS відповідає за зовнішній вигляд, а JavaScript все це оживляє. За допомогою коду на JavaScript програміст визначає, як сторінка відреагує на дії користувача.

Зараз JavaScript - єдина мова програмування для браузерів. Вона працює під Windows, macOS, Linux і на мобільних платформах, тобто скрізь. Якщо не знаєш JavaScript, робити в програмуванні інтерактивних сайтів нічого.

Вона досить проста але містить всі фундаментальні речі: алгоритми, об'єктно-орієнтовану модель, структури даних. Якщо традиційні мови для навчання - Pascal і Basic - несуть мало практичної користі, то JavaScript - робоча конячка.

Програма на JavaScript - це простий текст. Писати на JavaScript можна в будь-якому текстовому редакторі.

Частою помилкою зустрічається плутанина між Java і JavaScript - це дві різні мови, незважаючи на схожі назви.

Базовою особливістю цієї мови відзначається те, що на неї вплинули інші (Python, Java та ін.) Мови програмування з метою надання максимального комфорту JavaScript і легкості в розумінні її тими користувачами, які не мають відповідної освіти і глибинних знань - не програмістами. JavaScript - офіційно зареєстрована торгова марка компанії Oracle.

За допомогою неї доступні до виконання наступні функції:

- можливість змінювати сторінки браузерів;
- додавання або видалення тегів;
- зміна стилів сторінки;
- інформація про дії користувача на сторінці;
- запит доступу до випадкової частини вихідного коду сторінки;
- внесення змін до цього код;
- виконання дії з cookie-файлами.

Область застосування цієї мови дивно обширна і нічим не обмежена: серед програм, які використовують JS, присутні і тестові редактори, і додатки (як для комп'ютерів, так і мобільні і навіть серверні), і прикладне ПЗ.

В якості оболонки для коду було вирішено використовувати HTML та CSS задля створення простого та зручного інтерфейсу. Декілька слів про них.

HTML - це мова розмітки гіпертексту. Вона дозволяє користувачеві створювати і структурувати розділи, параграфи, заголовки, посилання і блоки для веб-сторінок і додатків[6].

HTML не є мовою програмування, тобто вона не має можливості створювати динамічні функції. Замість цього вона дозволяє організовувати і форматувати документи, аналогічно Microsoft Word.

HTML була винайдена Тімом Бернерс-Лі, фізиком з дослідницького інституту ЦЕРН в Швейцарії. Він придумав ідею Інтернет-гіпертекстової системи.

Hypertext означає текст, що містить посилання на інші тексти, які глядачі можуть отримати негайно. Він опублікував першу версію HTML в 1991 році, яка складається з 18 тегів HTML. З тих пір кожна нова версія мови HTML з'явилася з розміткою нових тегів і атрибутів (модифікаторів тегів).

Згідно із Довідником HTML Element Reference від Mozilla Developer Network, в даний час існує 140 тегів HTML, хоча деякі з них вже застаріли (не підтримуються сучасними браузерами).

У порівнянні з іншими мовами програмування, HTML найбільш простий у вивченні і зрозумілий навіть для новачків.

Сьогодні, коли в Інтернеті багато всіляких платформ з шаблонами і підказками для швидкого створення сайту будь-якої тематики, деякі просто не бачать сенсу вникати в тонкощі програмування. Однак знання HTML-розмітки знадобиться вам під час налаштування і просуванні сторінки, якщо ви захочете хоча б трохи відступити від базових шаблонів і вийти за їх межі, щоб перетворити свій ресурс в оригінальний.

Навіть початкові знання зазначеної мови дозволять красиво і креативно оформити самий звичайний текст, зробивши його більш привабливим для відвідувачів сайту.

HTML зрозумілий більшості додатків, навіть текстового редактора Word, тому може стати в нагоді, якщо ви хочете, щоб вони підтримували роботу з вашим сайтом.

За допомогою даної розмітки можна створити на сторінці прості, але оригінальні динамічні елементи і анімацію.

Саме HTML лежить в основі багатьох інших, більш складних, мов програмування. Якщо ви раптом захочете їх освоїти, то без базових знань доведеться туго.

HTML - основа будь-якої веб-сторінки. Якщо ви бажаєте зробити свій сайт барвистим, оригінальним і цікавим для потенційного відвідувача, то вам доведеться вносити в початковий шаблон певні зміни, а без необхідних знань

це не вдасться. Для будь-якого, хто хоча б раз стикався з необхідністю створення власного ресурсу, знання даної розмітки не буде зайвим і точно обійдеться дешевше, ніж послуги програміста.

Через швидке зростання популярності HTML тепер вважається офіційним веб-стандартом. Специфікації HTML підтримуються і розробляються консорціумом World Wide Web (W3C).

CSS - це мова таблиць стилів, яка використовується для стилізації елементів, написаних на мові розмітки, наприклад HTML. Вона відокремлює контент від візуального представлення сайту. Давайте розберемося, що таке CSS більш детально[7].

CSS була розроблена W3C (World Wide Web Consortium) в 1996 році за досить простої причини. В HTML не були розроблені теги, які допомогли б відформатувати сторінку. Потрібно було тільки написати розмітку для сайту.

Відносини між HTML і CSS сильно пов'язані між собою. Оскільки HTML - це мова розмітки (сама основа сайту), а CSS підкреслює стиль (всю естетику сайту), вони йдуть рука об руку. З технічної точки зору CSS не є необхідністю, але ви, ймовірно, не захочете дивитися на сайт, який містить тільки HTML, оскільки він буде виглядати абсолютно голим.

Різниця між сайтом, який реалізує CSS, і тим, який не використовує, величезна і, безумовно, помітна.

Можливо, ви бачили веб-сайт, який не завантажується повністю і має білий фон з більшою частиною синього і чорного тексту. Це означає, що CSS-частина сайту не була завантажена правильно або не існує взагалі.

Перед використанням CSS вся стилізація повинна була бути включена в HTML-розмітку. Це означає, що ви повинні були окремо описати весь фон, кольору шрифту, вирівнювання і т. д.

CSS дозволяє стилізувати все в іншому файлі, створюючи там стиль, а потім інтегруючи файл CSS поверх розмітки HTML. Це робить реальну HTML-розмітку набагато чистіше і простіше в обслуговуванні.

Коротше кажучи, за допомогою CSS вам не потрібно багато разів описувати зовнішній вигляд окремих елементів. Це економить час, зменшує код і робить його менш схильним до помилок.

До того, як з'явився CSS, оформлення web-сторінок могло здійснюватися безпосередньо всередині вмісту документа, поява ж технології CSS дало можливість поділу змісту та подання документа. Завдяки цій можливості стало реальним просте застосування єдиного стилю оформлення для більшості подібних документів, і стало доступно швидка зміна їх оформлення.

Переваги CSS:

Простота самої мови CSS разом з принципом відділення оформлення від змісту дає можливість скоротити час на розробку і підтримку сайту.

Є кілька варіантів дизайнів сторінки для перегляду на різних пристроях. Наприклад, дизайн на екрані комп'ютера розрахований на одну ширину, і буде повністю виводитися на екран, а на мобільних пристроях він буде підлаштовуватися до розмірів екрану і деякі елементи будуть виключені від показу, також і при друці, буде друкуватися потрібний текст, без зайвого (наприклад , без шапки меню).

Зменшується час завантаження сторінок web-сайту за рахунок перенесення правил представлення даних в окремий CSS-файл. Завдяки цьому браузер завантажує тільки структуру документа, а також дані, що зберігаються на сторінці, а представлення цих даних завантажується браузером тільки один раз і можуть бути закешировані, - завдяки цьому зменшується трафік, час завантаження, а також навантаження на сервер.

Простота зміни дизайну. Один CSS управляє відображенням безлічі HTML-сторінок. Коли виникає необхідність змінити дизайн сайту, то нема чого правити кожну сторінку. Для подальшої зміни дизайну всього лише потрібно змінити CSS-файл, і як результат, зміна дизайну робиться швидше.

CSS надає додаткові можливості форматування, про яких при використанні тільки самих атрибутів навіть і не мріяли.

Підвищення сумісності з різними платформами за рахунок використання web-стандартів.

Недоліки CSS:

Різна відображення верстки в різних браузерах. Якщо браузери застарілі, то можливо, що одні й ті ж дані CSS по-різному ними інтерпретуються.

Необхідність виправляти не тільки один CSS-файл, але і теги HTML. Іноді це значно збільшує час редагування, а також і тестування.

Отже, CSS - це найсильніший інструмент і один з основних складових практично будь-якого web-сайту. Сьогодні CSS - це загальноприйнятий стандарт розробки, який приймається усіма без винятку компаніями-розробниками, що явно показує його значимість і необхідність.

4.2 Опис процесу програмної реалізації

В першу чергу потрібно було розробити приємний та зручний інтерфейс програми. Для цього були використана мова розмітки гіпертексту HTML та каскадні таблиці стилів CSS. Завдяки HTML створили сторінку з основною інформацією про програму шифрування. CSS відповідав за оформлення дизайну сайту, тому з його допомогою на сторінці з’явились наступні об’єкти:

- контейнер для тексту;
- контейнер для ключа;
- кнопки шифрування та дешифрування;

Наступним кроком було підключення CSS файлу до HTML документу однією командою, завдяки чому наша сторінка отримала наступний вигляд:

```
<link rel="stylesheet"href="gc.css">
```

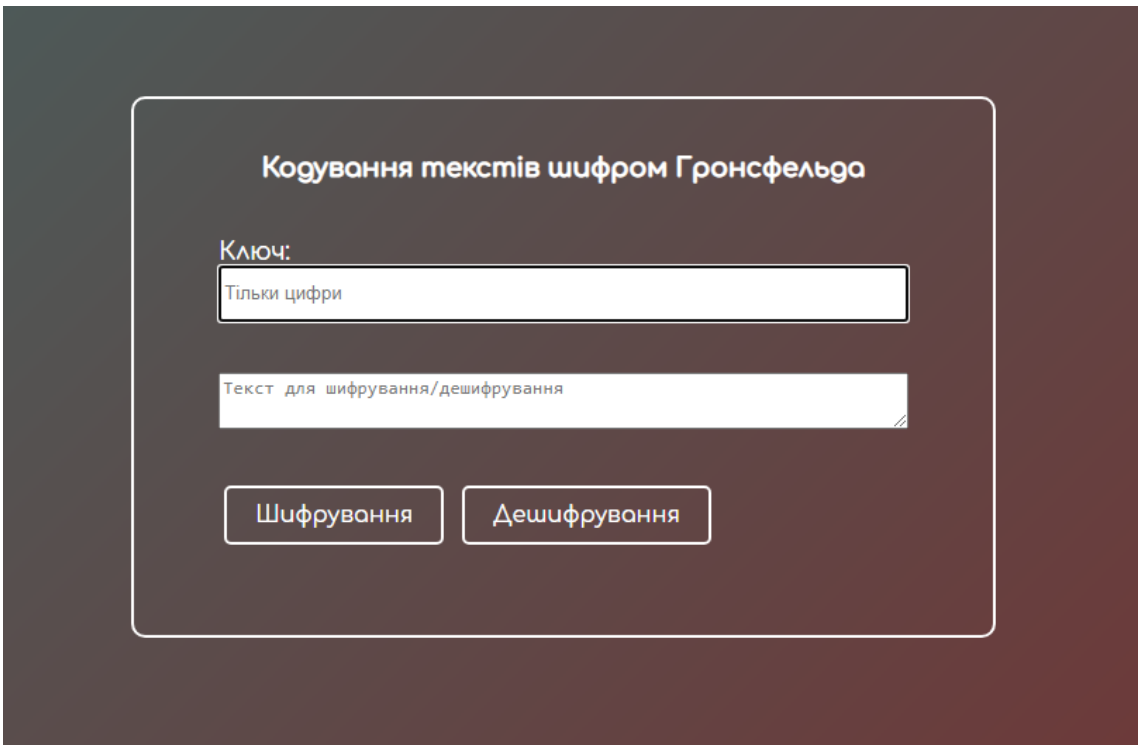


Рис. 4.1 – Представлення графічного інтерфейсу програмної реалізації

Нижче представлений код створення сторінки за допомогою каскадної таблиці стилів CSS:

```
body {
background: linear-gradient(-45deg, #377631, #763131,
#317674, #653176);
background-size: 400% 400%;
animation: gradient 15s ease infinite;
font-family: 'Comfortaa', cursive;
```

```
color: #fff;
}

@keyframes gradient {
0% {
background-position: 0% 50%;
}
50% {
background-position: 100% 50%;
}
100% {
background-position: 0% 50%;
}
}

.container {
background-color: rgba(117, 190, 218, 0.0);
margin-top: 200px;
margin-left: auto;
margin-right: auto;
width: 40%;
border-radius: 10px;
border: 2px solid #fff;
padding: 20px;
text-align: center;

}

textarea,
```

```
input {
-webkit-box-sizing: border-box;
-moz-box-sizing: border-box;
box-sizing: border-box;
width: 100%;
height: 40px;
}

.button {
  font-family: 'Comfortaa', cursive;
  color: #fff;
  background-color: rgba(117, 190, 218, 0.0);
  border-radius: 5px;
  border: 2px solid #fff;
  padding: 10px 20px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 16px;
  margin: 4px;
  margin-top: 40px;
  cursor: pointer;
}

.button:hover {
  cursor: pointer;
  background-color: #fff;
  color: #000;
}
```

```
.button:active {
  cursor: pointer;
}

.fieldContainer {
  padding: 20px;
  margin: 20px;
  font-size: 17px;
  text-align: left;
}

@media screen and (max-width: 900px) {
  .container {
    width: 90%;
  }
}
```

Нижче представлений код створення програми за допомогою прототипної мови програмування JavaScript :

Шифрування

```
function encrypt() {
  var key = document.getElementById("keyValue").value;
  var keyArr = key.split("");
  var text = document.getElementById("text").value;
  var result = "";
  var counter = 0;

  for (var i = 0; i < text.length; ++i) {
```

```
    var c = text.charCodeAt(i);

    if (c === 32) {
      counter = -1;
      result += String.fromCharCode(c)
    } else if (c < 65 || c > 122) {
      result += String.fromCharCode(c)
    } else if (c > 90 && c < 97) {
      result += String.fromCharCode(c)
    } else if (c >= 97 && (parseInt(c) +
parseInt(keyArr[counter])) > 122) {
      result += String.fromCharCode(parseInt(96) +
parseInt((parseInt(c) + parseInt(keyArr[counter]) -
parseInt(122))));
    } else if (c <= 90 && (parseInt(c) +
parseInt(keyArr[counter])) > 90) {

      result += String.fromCharCode(parseInt(64) +
parseInt((parseInt(c) + parseInt(keyArr[counter]) -
parseInt(90))));
    } else {
      result += String.fromCharCode(parseInt(c) +
parseInt(keyArr[counter]));
    }

    ++counter;

    if (counter === keyArr.length) {
      counter = 0;
```

```
    }
  }

  document.getElementById("text").value = result;
}
```

Дешифрування

```
function decipher() {
  var key = document.getElementById("keyValue").value;
  var keyArr = key.split("");
  var text = document.getElementById("text").value;
  var result = "";
  var counter = 0;

  for (var i = 0; i < text.length; ++i) {
    var c = text.charCodeAt(i);

    if (c === 32) {

      counter = -1;
      result += String.fromCharCode(c)
    } else if (c >= 65 && c <= 90) {
      if ((parseInt(c) - parseInt(keyArr[counter])) <
65) {
        result += String.fromCharCode(parseInt(91) -
parseInt((parseInt(65) - (parseInt(c) -
parseInt(keyArr[counter])))));
      }
      else {
```

```
        result += String.fromCharCode(parseInt(c) -
parseInt(keyArr[counter]));
      }
    } else if (c >= 97 && c <= 122) {
      if ((parseInt(c) - parseInt(keyArr[counter])) <
97) {
        result += String.fromCharCode(parseInt(123) -
parseInt((parseInt(97) - (parseInt(c) -
parseInt(keyArr[counter])))));
      }
      else {
        result += String.fromCharCode(parseInt(c) -
parseInt(keyArr[counter]));
      }
    } else {
      result += String.fromCharCode(c);
    }

    ++counter;

    if (counter === keyArr.length) {
      counter = 0;
    }
  }
```

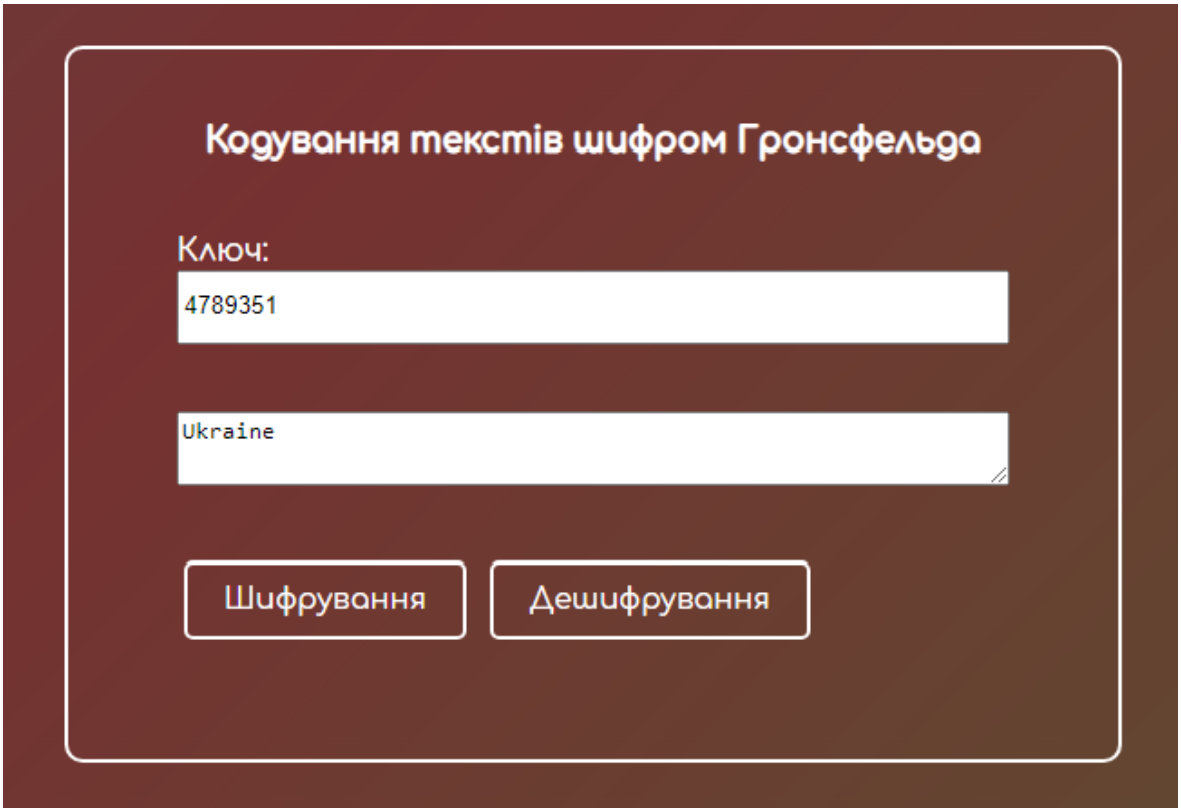


Рис. 4.2 – Шифрування слова Ukraine

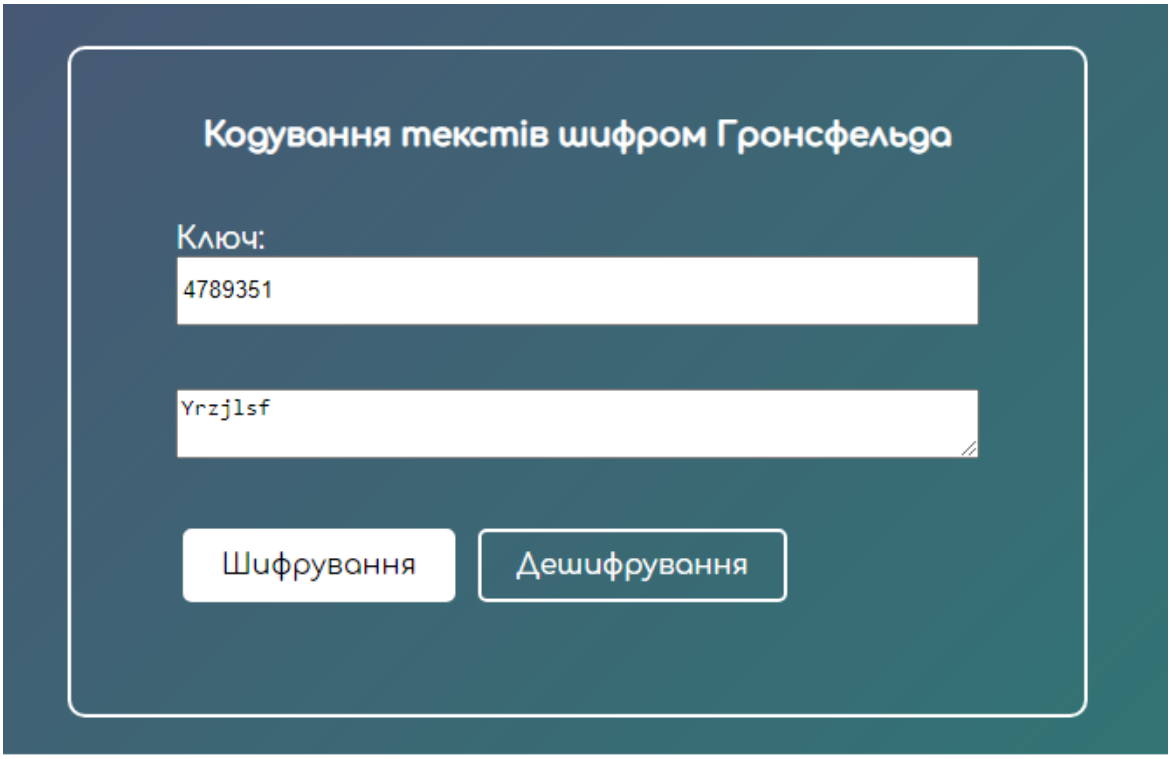


Рис. 4.3 – Результат шифрування слова Ukraine

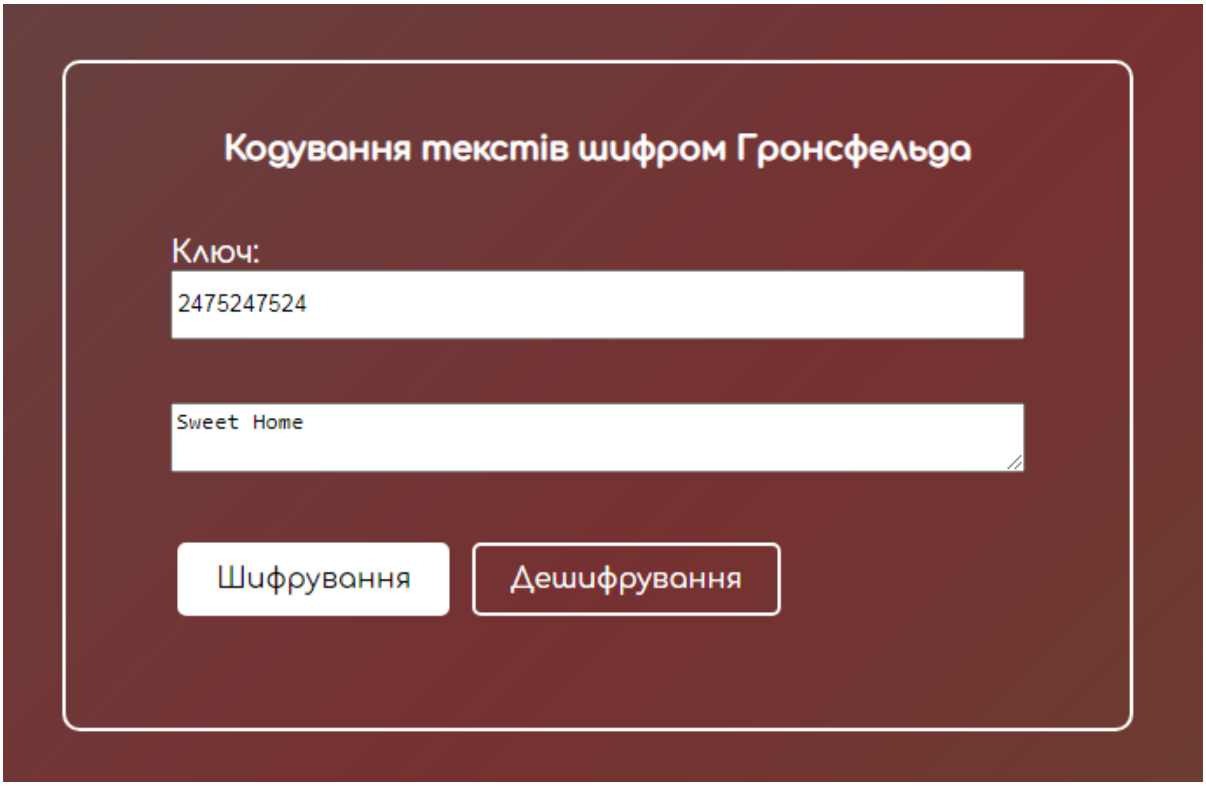


Рис. 4.4 – Шифрування фрази «Sweet Home»

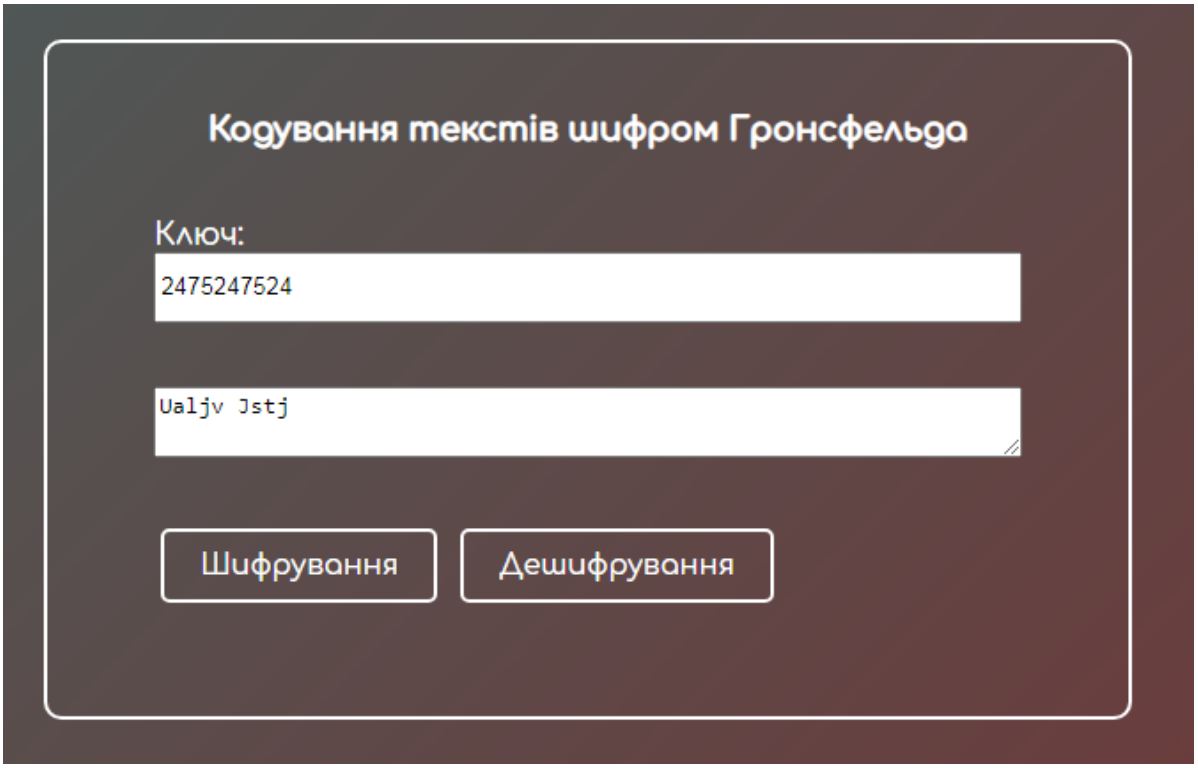


Рис. 4.5 – Результат шифрування фрази "Sweet Home»

4.3 Інструкція користувача по роботі з програмою

Для початку роботи потрібно відкрити файл з розширенням .html і перейти на сторінку програми. Перед користувачем відкриється вікно програми.

Перед шифрування або дешифрування в поле для тексту (рис. 4.6) необхідно ввести слово або речення.

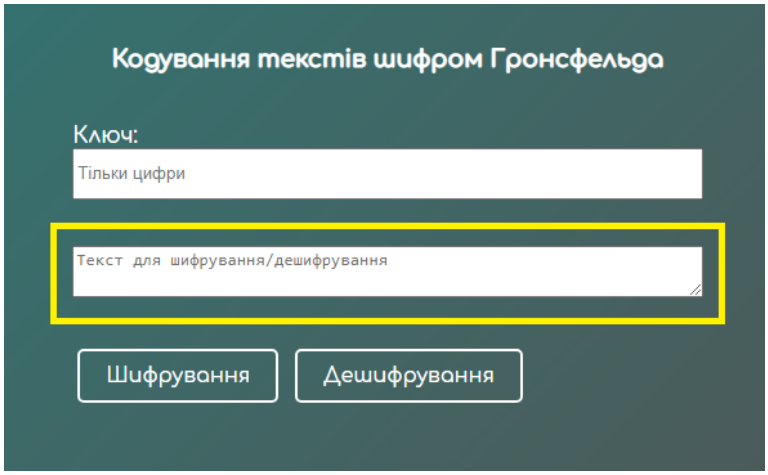


Рис. 4.6 – Поле для тексту

Далі в поле ключ (рис. 4.7) треба ввести цифри для шифрування або дешифрування.

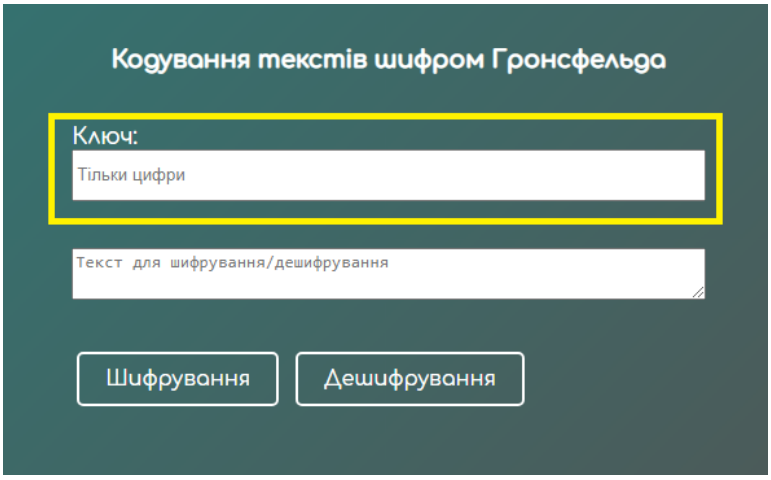


Рис. 4.7 – Поле для ключа

Наступним кроком потрібно натиснути необхідну кнопку: шифрування або дешифрування, залежного від потреби та отримати результат (рис. 4.8).

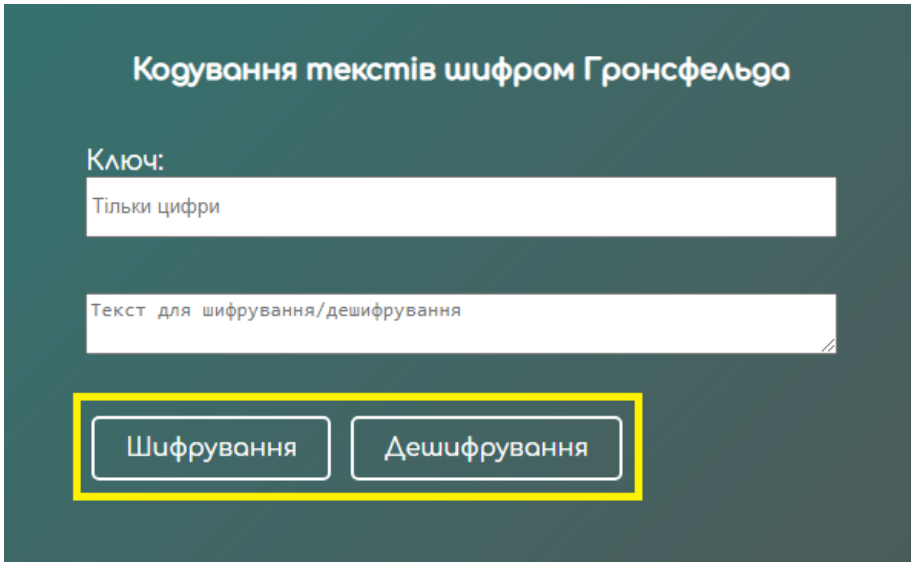


Рис. 4.8 – Кнопки "Шифрування" та "Дешифрування"

ВИСНОВКИ

В бакалаврській роботі було досліджено теоретичний матеріал за темою «Шифр Гронсфельда в кодуванні: програмування та дослідження». Було підібрано кілька прикладів для реалізації в дипломній роботі.

Було переглянуто деякі з наявних у мережі Інтернет програм на аналогічну тематику. Проаналізовано їх переваги та недоліки.

В процесі роботи ознайомились з мовами програмування та прийшли до висновку, що проект доцільно написати мовою JavaScript за допомогою HTML та CSS.

Розроблено алгоритм програми, блок-схему, створено програмну реалізацію.

Програма перевірена на тестових прикладах та описана.

Розроблена інструкція по роботі з програмою.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Інформаційний портал Криптографія для шифрування та обчислень різними шифрами та методами. Режим доступу: https://cryptography.ucoz.net/index/shifr_gronsfelda/0-20

2. dCode - це універсальний сайт для декодування повідомлень, листівських ігор, розгадування головоломок, геокеш-пам'яті, пошуку скарбів тощо. Режим доступу: https://www.dcode.fr/gronsfeld-cipher

3. Md5 – сайт з великою кількістю сторінок для шифрування різними методами. Режим доступу: https://md5decrypt.net/en/Gronsfeld-cipher/

4. Boxentriq – сайт з різними онлайн сервісами для вирішення головоломок та дешифрування текстів. Режим доступу: https://www.boxentriq.com/code-breaking/gronsfeld-cipher

5. Популярні мови програмування. Режим доступу: https://tproger.ru/articles/top-10-jazykov-programmirovanija-v-2020-godu-po-versii-github/

6. Основи мови розмітки гіпертексту. Режим доступу: https://www.hostinger.com.ua/rukovodstva/shto-takoje-html/

7. Каскадні таблиці стилів для початківців. Режим доступу: https://www.hostinger.com.ua/rukovodstva/shto-takoje-css/