ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ ІМЕНІ СЕМЕНА КУЗНЕЦЯ

Лабораторна робота №2

з курсу «Безпека банківських систем»

ВИВЧЕННЯ СИСТЕМИ ЗАХИСТУ ДАНИХ TRUECRYPT

Харків 2023

Мета: вивчити алгоритми симетричного шифрування та одностороннього хешування. Ознайомитись з можливостями сучасних програм шифрування даних на прикладі програми TrueCrypt.

1 ТЕОРЕТИЧНІ ВІДОМОСТІ

1.1 Криптографічні алгоритми

Криптографічний алгоритм, також званий шифром, являє собою математичну функцію, що використовується для шифрування та дешифрування. Під шифруванням розуміють процес перетворення відкритих даних на зашифровані за допомогою ключа. Дешифрування – зворотний процес шифрування. На основі ключа шифрований текст перетворюється на вихідний. Ключ – це інформація, необхідна для безперешкодного шифрування та дешифрування даних. Ключ може бути будь-яким значенням. Зазвичай він є послідовним рядом цифр і букв алфавіту.

Множину можливих ключів називають простором ключів.

Криптосистема є алгоритм плюс всі можливі відкриті тексти, шифротексти та ключі.

Існує два основних типи алгоритмів, заснованих на ключах: симетричні та з відкритим ключем (асиметричні). Симетричні алгоритми (рис. 1), є алгоритми, у яких ключ шифрування може бути розрахований за ключом дешифрування і навпаки. У більшості симетричних алгоритмів ключі шифрування і дешифрування одні й ті самі. Ці алгоритми, також звані алгоритмами з секретним ключем або алгоритмами з одним ключем, вимагають, щоб відправник і одержувач узгодили ключ, що використовується перед початком безпечної передачі повідомлень. Розкриття ключа означає, що будь-хто зможе шифрувати та дешифрувати повідомлення.

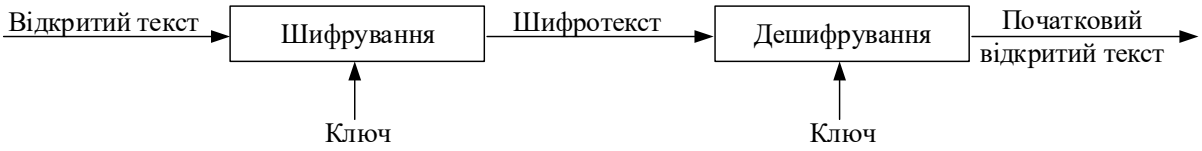

Рисунок 1 – Шифрування та дешифрування з ключем

Безпека симетричних алгоритмів повністю ґрунтується на ключах, а не на деталях алгоритмів. Це означає, що алгоритм може бути опублікований та проаналізований.

Симетричні алгоритми поділяються на дві категорії. Одні алгоритми обробляють відкритий текст побітно (іноді побайтно), вони називаються потоковими алгоритмами чи потоковими шифрами. Інші працюють із групами бітів відкритого тексту. Групи бітів називаються блоками, а алгоритми – блоковими алгоритмами чи блоковими шифрами.

При поточному шифруванні шифрується потік бітів (рис. 2). Такі шифри працюють у два етапи.

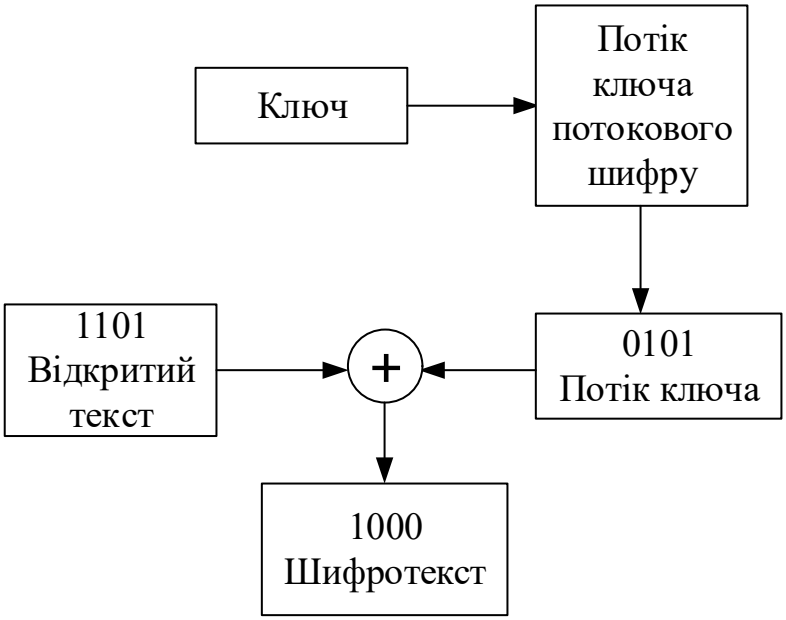

Рисунок 2 – Алгоритм потокового шифрування

На першому етапі алгоритм використовує секретний ключ генерації потоку випадкових бітів. На другому етапі алгоритм шифрує біти даних,

об'єднуючи їх по одному з потоком випадкових бітів, використовуючи операцію «Виключне АБО». Якщо змінити один біт у відкритому тексті, це вплине лише на той самий біт у шифрованому тексті. Характерна риса цього способу шифрування – висока продуктивність. Прикладами таких шифрів є Enigma та RC4.

При шифруванні блоків кодування даних виконується після розбиття їх на блоки фіксованої довжини, кожен з яких шифрується окремо за допомогою одного і того ж ключа (рис. 3). Якщо розмір вихідної інформації не кратаний розміру блоку, то останній блок доповнюється символами-заповнювачами, які при розшифруванні відкидаються. Прикладами блокових шифрів можуть бути DES, ГОСТ 28147-89, RC5, AES.

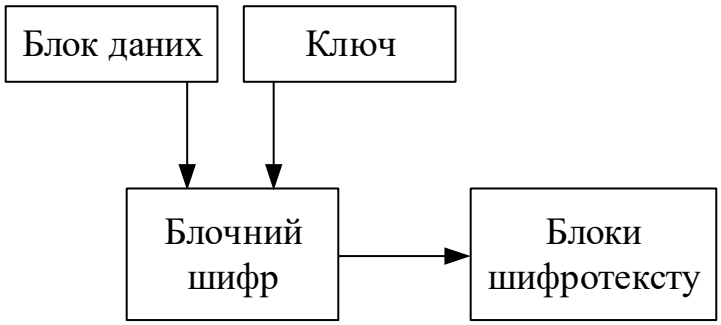

Рисунок 3 – Алгоритм блочного шифрування

Хешувальна функція – це багатозначна функція H, що ставить у відповідність до свого аргументу M довільної довжини значення $h=H(M)$ фіксованої довжини. Хеш-функції, що використовуються в криптографічних застосунках, мають такі властивості:

– для будь-якого M легко обчислити його хеш-значення $h=H(M)$;

– за відомим значенням хеш-функції практично неможливо визначити повідомлення M;

– для повідомлення M практично неможливо знайти повідомлення M' таке, щоб значення хеш-функцій цих повідомлень були однакові;

– практично неможливо знайти два випадкові повідомлення M і M' таких, щоб значення хеш-функцій цих повідомлень були однакові (властивість сильної стійкості до колізій).

Хеш-значення повідомлення може використовуватися як його унікальний ідентифікатор. Якщо повідомлення змінити, зміниться і хеш-значення. Знайти інше повідомлення з таким самим хеш-значенням практично неможливо. Завдяки цій властивості хеш-функції можна використовувати для перевірки цілісності документів та цифрового підпису. Очевидно, що найбільш важливим параметром хеш-функції є довжина значення, що нею повертається. Якщо ця довжина n біт, то для того, щоб знайти повідомлення M', що має таке ж хеш-значення, як і M, доведеться перебрати в середньому 2^{n-1} різних повідомлень.

Прикладами хеш-функцій можуть бути MD2, MD4, MD5, SHA.

1.2 TrueCrypt

TrueCrypt – це криптографічний застосунок для створення зашифрованих файл-контейнерів і шифрування розділів дисків, а також знімних пристроїв зберігання інформації.

Можливості TrueCrypt:

– запис та читання зашифрованих дисків;

– створення звичайних зашифрованих дисків;

– створення прихованих зашифрованих дисків;

– монтування зашифрованих фізичних дисків;

– можливість встановлювати (крім пароля) ключові файли;

– набір алгоритмів (на вибір користувача) шифрування;

– версії TrueCrypt доступні під Windows та Linux (можливість використовувати один і той же том, як по мережі, так і з різних операційних систем).

TrueCrypt (рис. 4) дозволяє зашифровувати та розшифровувати дані «на льоту». Це означає, що дані автоматично зашифровуються або розшифровуються, перш ніж вони будуть завантажені або збережені, без будь-якого втручання користувача.

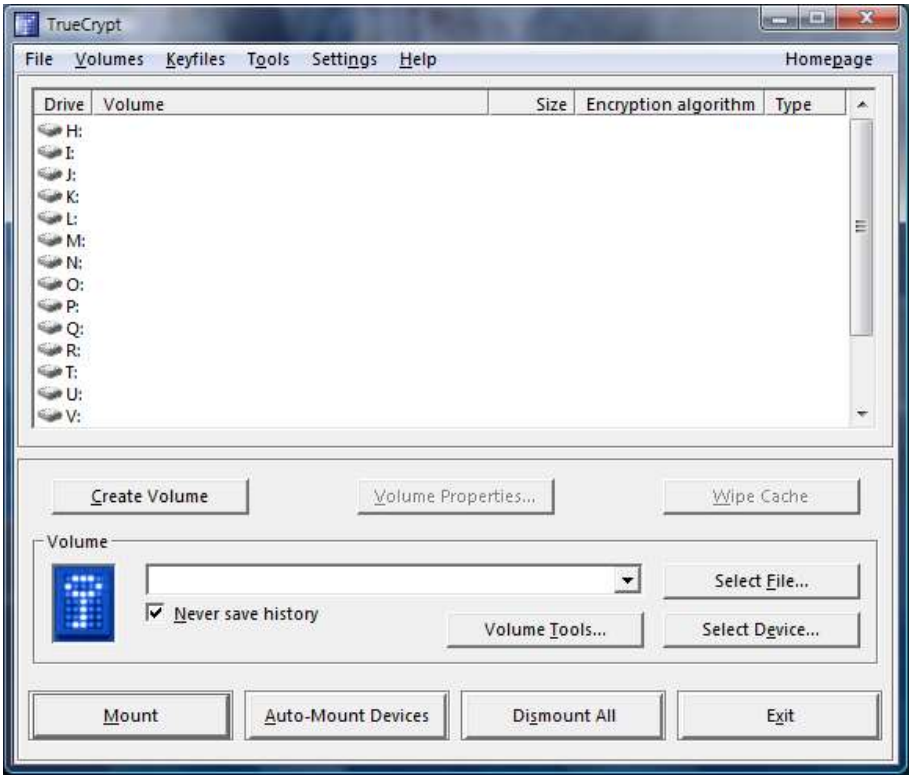

Рисунок 4 – Вікно програми TrueCrypt

Розшифрування відбувається в оперативній пам'яті при зчитуванні файлів з монтованого тома, при цьому TrueCrypt ніколи не зберігає жодних тимчасових даних на диск – всі операції відбуваються в оперативній пам'яті. Так само відбувається і шифрування. Дані, що зберігаються на зашифрованому томі, неможливо розшифрувати без правильного секретного ключа. Програма повністю шифрує файлову систему тому (імена файлів, назву папок та їх зміст, вільний простір тощо). Для зберігання паролів у TrueCrypt використовується хешування.

1.2.1 Створення зовнішнього тому

Щоб створити том TrueCrypt, потрібно натиснути кнопку «Створити том» у головному вікні програми. У вікні можна вибрати тип створюваного тома:

– звичайний (зовнішній) том;

– прихований том усередині звичайного.

Останній варіант дозволить приховувати важливі дані.

Програма створює томи таким чином, що неможливо дізнатися, чи є у звичайному томі прихований.

Розглянемо процес створення звичайного тому TrueCrypt. Пізніше, за бажання, у створеному звичному томі можна буде створити прихований.

Як том TrueCrypt може використовуватися або файл, або пристрій (рис. 5). При виборі пристрою шифрується весь вміст розділів жорстких дисків або змінних носіїв, при цьому *вся інформація, що зберігається, буде знищена*.

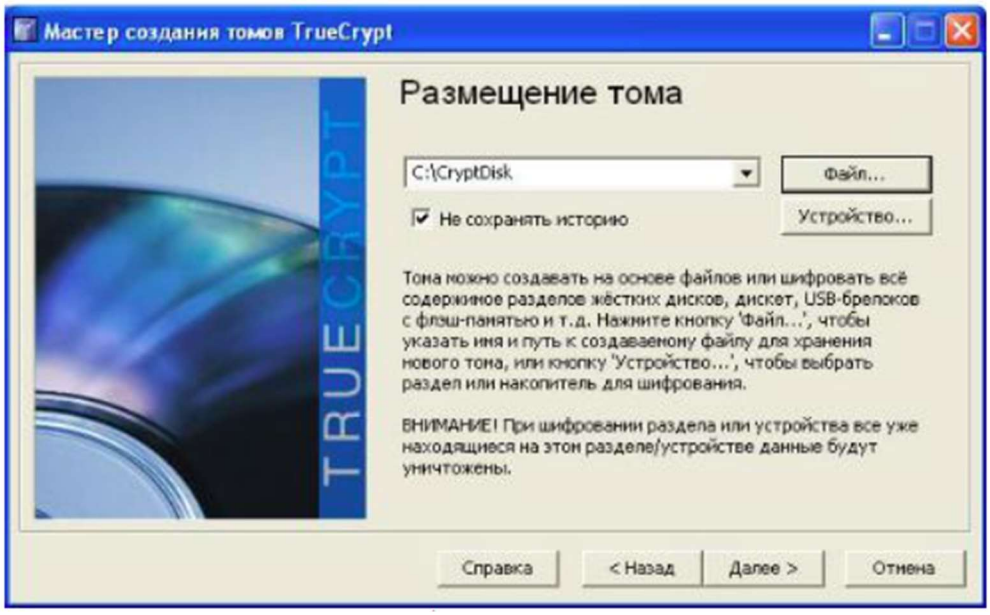

Рисунок 5 – Вибір розміщення тома

Створимо том на основі файлу. Для цього скористаємося кнопкою «Файл ...». У діалоговому вікні вкажемо ім'я створюваного файлу та його розміщення. У наступному діалоговому вікні буде запропоновано вибрати параметри шифрування. Тут можна протестувати алгоритми на швидкість шифрування і дешифрування.

Після цього буде запропоновано вибрати розмір тома. Необхідний розмір можна вказати у кілобайтах чи мегабайтах.

Далі потрібно встановити пароль та/або ключовий файл (рис. 6).

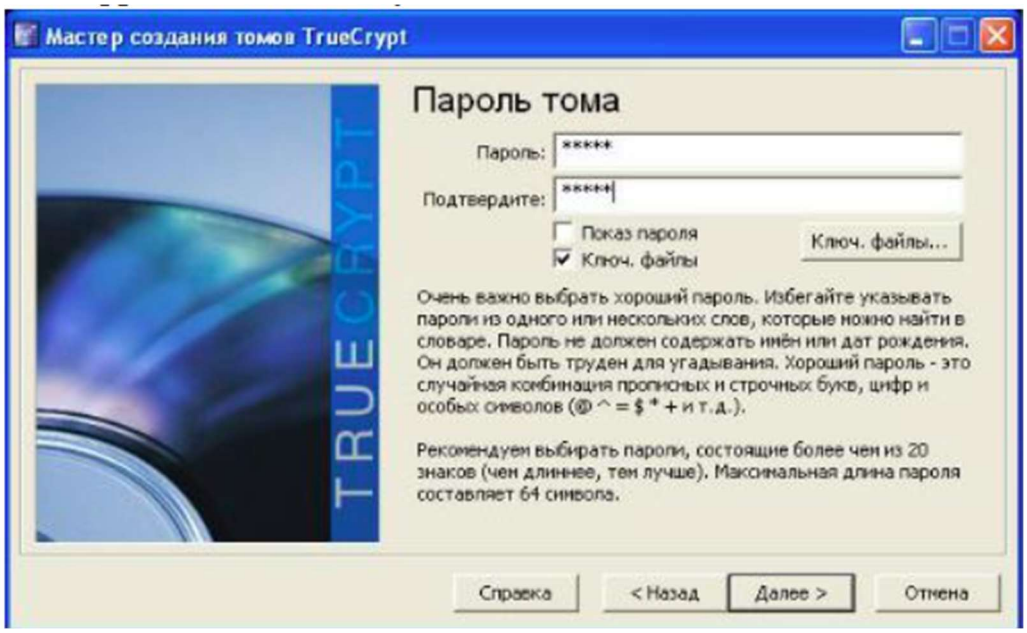

Рисунок 6 – Створення паролю

Виробник рекомендує вибирати пароль з більш ніж двадцяти символів. Ключовий файл – це файл, зміст якого використовується як ключ. Встановлення ключового фалу можна скомбінувати з установкою пароля. Навіть якщо пароль буде розсекречено, то без ключового файлу розшифрувати дані буде неможливо.

Як ключовий можна використовувати будь-який файл. TrueCrypt не змінює ключові файли. Дозволено вибирати кілька ключових файлів (порядок неважливий). При додаванні папки ключовими будуть всі наявні в ній файли. Програма також дозволяє генерувати випадкові ключові файли. Для цього слід скористатися кнопкою «Випадковий ключовий файл». При втраті ключового файлу або пошкодженні його перших 1024 кілобайт монтування томів, що використовують цей файл, буде неможливо, і вся інформація буде втрачена.

На наступному кроці буде запропоновано вибрати файлову систему, розмір кластера та тип створюваного контейнера.

Слід пам'ятати, що в NTFS-томі не можна створити прихований том. Крім того, у Windows не можна монтувати NTFS томи з опцією лише для читання. Тому при виборі файлової системи слід спиратися на конкретні потреби.

При створенні динамічного тома буде створено файл, фізичний розмір якого (місце, яке він займає на диску) збільшується в міру додавання до нього нових даних. Динамічні томи можуть бути створені лише з файловою системою NTFS.

При натисканні на кнопку «Виконати» після деякої паузи програма випадковим чином згенерує ключі. На цьому створення шифрованого диска буде завершено.

1.2.2 Створення прихованого тому

TrueCrypt поряд зі звичайними дозволяє створювати і приховані томи. Суть цього методу полягає в тому, що навіть коли змонтований зовнішній том, дізнатися чи є в ньому прихований том неможливо, тому що вільне місце в будь-якому томі TrueCrypt при створенні завжди заповнюється випадковими даними (якщо вимкнено швидке форматування), і відрізнити прихований том від випадкових даних не можна (рис. 7).

Це дозволяє навіть за умови розкриття пароля зовнішнього тому зберегти дані на прихованому томі в секреті. Паролі для прихованого та зовнішнього томів мають бути різними.

Для створення прихованого тому також використовується майстер створення томів. Тільки при виборі типу тома необхідно вибрати «Створити прихований том TrueCrypt». Використовуючи інструкції майстра, можна створити том усередині вже створеного. Якщо раніше звичайний том не було створено, можна скористатися пунктом «Створити том TrueCrypt і прихований том усередині нього». У наступному діалоговому вікні потрібно вказати шлях до створеного звичайного того.

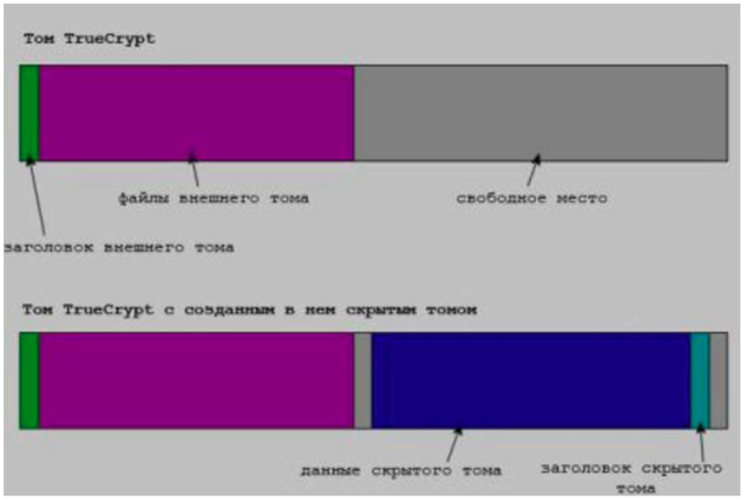

Рисунок 7 – Структура томів True Crypt

Далі програма запросить ввести пароль до зовнішнього тому. При правильному введенні пароля з'явиться майстер створення прихованого тома. Усе готово.

Дії при створенні прихованого тому аналогічні діям при створенні звичайного.

Для використання створеного шифрованого диска його потрібно змонтувати. Для цього в головному вікні програми потрібно натиснути кнопку «Файл…», щоб вказати розташування файлу, в якому зберігатиметься потрібний том. Тепер, коли вибрано файл, потрібно натиснути кнопку «Змонтувати». Якщо потрібно змонтувати прихований том, потрібно вибрати файл, в якому зберігається прихований том, і ввести пароль від прихованого тома. На вимогу програми введіть пароль та ключовий файл, вказаний раніше під час створення цього тома.

Перш ніж натиснути на «ОК», скористайтеся кнопкою «Параметри...». У діалоговому вікні, що з'явилося, позначте пункт «Захистити прихований том від пошкоджень при записі у зовнішній том» (рис. 8). Це потрібно для того, щоб захистити дані прихованого тому. Тепер при спробі запису даних в область прихованого тома, програма заборонятиме запис на весь том до його розмонтування.

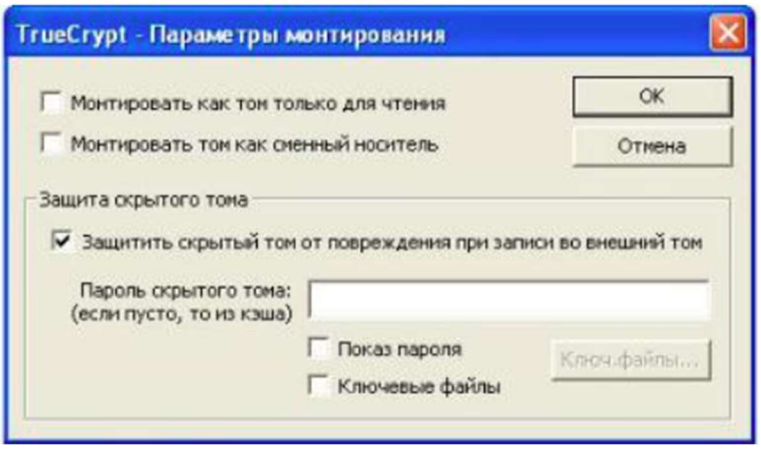

Рисунок 8 – Налаштування параметрів монтування

Після монтування зашифрованим диском можна буде користуватися як звичайним логічним. На диск можна поміщати будь-яку інформацію. Наприклад, монтуємо том TrueCrypt як диск Q (рис. 9). Він буде відображатись у провіднику у вигляді локального диска Q.

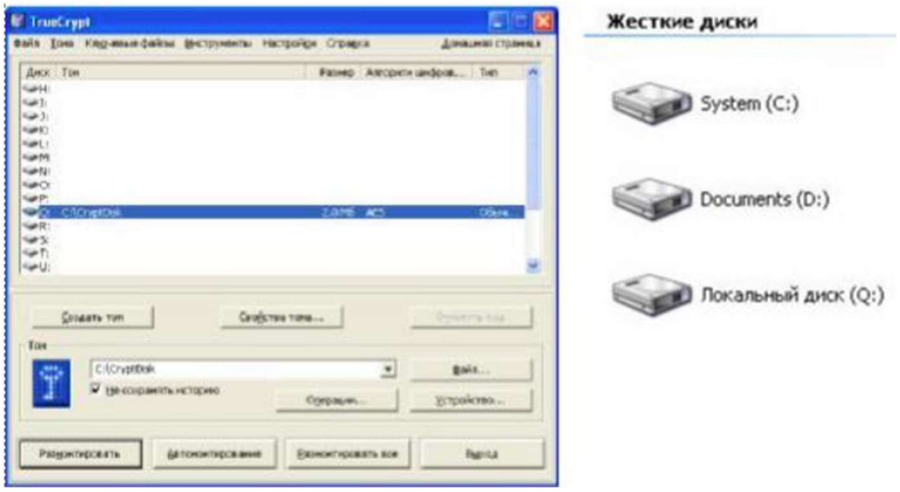

Рисунок 9 – Приклад монтування диска

Щоб вимкнути диск, потрібно скористатися пунктом контекстного меню «Розмонтувати», вибравши відповідний диск, або, за бажанням, можна розмонтувати всі змонтовані диски, скориставшись кнопкою «Розмонтувати всі».

3 ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

1. За допомогою програми TrueCrypt створити звичайний том, захищений звичайним паролем.

2. За допомогою програми TrueCrypt створити звичайний том, захищений ключовим файлом.

3. За допомогою програми TrueCrypt створити прихований том у вже створеному звичайному томі.

4. Створити текстовий файл із певною послідовністю символів, помістити файл на змонтований носій, розмонтувати, спробувати виявити послідовність під час перегляду файлу TrueCrypt звичайними засобами перегляду.

5. Змонтувати прихований розділ та скопіювати створений у п.4 файл на цей розділ.

6. Спробувати змонтувати основний розділ при неправильному паролі.

7. За допомогою програм md5sum або sha1sum отримати хеш файлу з п.4. Скопіюйте цей файл на змонтований диск і отримати хеш там. Порівняти отримані результати та зробити висновки.

8. Перенести том TrueCrypt на інший комп'ютер із встановленою програмою TrueCrypt та спробувати його змонтувати (*не обов'язково, при наявності технічної можливості*).

Контрольні питання:

1. Що таке симетричні алгоритми шифрування? Наведіть приклади.
2. Які типи симетричних алгоритмів є?
3. Що таке хеш-функції? Навіщо вони застосовуються?
4. Назвіть основні можливості TrueCrypt.
5. Який принцип роботи TrueCrypt?
6. Для чого потрібні приховані томи TrueCrypt?