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

Безпека банківських систем

Методичні вказівки до виконання лабораторних робіт
для студентів денної форми навчання за спеціальністю 125 “ Кібербезпека ”

ЗАТВЕРДЖЕНО
на засіданні кафедри кібербезпеки та
програмного забезпечення, протокол №13
від 31.03.2022.

Кропивницький
2022

Безпека банківських систем: методичні вказівки до виконання лабораторних робіт для студентів за спеціальністю 125 «Кібербезпека»/ М-во освіти і науки України, Центральноукр. нац. техн. ун-т; [уклад. П. С. Усік, С. П. Євсеєв, К. О. Буравченко] – Кропивницький: ЦНТУ, 2022. – 39 с.

Укладачі: Усік П. С., доктор філософії, викладач;
Євсеєв С. П., докт. техн. наук, професор;
Буравченко К. О., канд. техн. наук, ст. викладач.

Рецензенти: Смірнов О. А., докт. техн. наук, професор;
Коваленко О. В., докт. техн. наук, професор.

ЗМІСТ

ВСТУП

Мета: Основною метою є формування у здобувачів вищої освіти ґрунтовних теоретичних знань, практичних умінь та навичок в області захисту банківських інформаційних ресурсів, системами й методами визначення захищеності програмних продуктів в автоматизованих банківських системах.

Завдання:

– Вивчення теоретичних основ безпеки банківських системах.

– Вивчення способів забезпечення безпеки банківських системах.

– Набути практичних навичок роботи з програмними засобами забезпечення безпеки банківських інформаційних ресурсів.

У результаті вивчення навчальної дисципліни студент повинен:

– знати про джерела і способи дії загроз на об’єкти інформаційної безпеки банківських установ; про правові і нормативні акти, які визначають систему захисту інформації в автоматизованих банківських системах; про документи, що визначають ступінь захищеності комп’ютерних систем; методи аналізу надійності системи захисту інформації в автоматизованих банківських системах; основні методи, технологію, принципи і правила захисту транзакцій, у тому числі від сучасних загроз гібридного характеру;

– вміти працювати з програмними засобами забезпечення безпеки банківських інформаційних ресурсів в автоматизованих банківських системах за складовими безпеки: ІБ, КБ, та БІ, проводити комплексну оцінки дотримання вимог регуляторів щодо забезпечення інформаційної безпеки, налаштування програмних застосунків щодо електронного цифрового підпису, механізмів PKI.

Структурно логічна схема підготовки бакалавра.

Враховуючи послідовність накопичення знань та інформації, дисципліна вивчається після викладання наступних дисциплін: «Вступ до кібербезпеки».

Для опанування матеріалу дисципліни «Безпека банківських системах» окрім лекційних та лабораторних занять, тобто аудиторного навантаження, значна увага приділяється самостійній роботі.

До основних видів самостійної роботи студента відносимо:

1. Вивчення лекційного матеріалу.

2. Робота з літературними джерелами.

3. Розв’язання практичних задач.

4. Підготовка до модульних, підсумкового контролю, екзамену (денна) та заліку (заочна).

5. Виконання контрольної роботи для заочної форми навчання.

В ході викладання дисципліни викладачем застосовуються види занять, які згідно з програмою навчальної дисципліни передбачають лекційні, та лабораторні заняття, а також виконання самостійної роботи.

Основна мета лекції – дати систематизовані основи знань з навчальної дисципліни, зосередити увагу студентів на найбільш складних та ключових питаннях.

Основна мета лабораторної роботи – закріплення й деталізація знань, а головне – формування навичок і вмінь.

Шкала оцінювання: національна та ЄКТС

Сума балів за всі види навчальної діяльності	Оцінка ЄКТС	Оцінка за національною шкалою
		для екзамену
90-100	A	відмінно
82-89	B	добре
74-81	C	
64-73	D	задовільно
60-63	E	
35-59	FX	незадовільно з можливістю повторного складання
1-34	F	незадовільно з обов’язковим повторним вивченням дисципліни

Вибравши предметну область, над якою ви будете працювати, ви повинні виконати завдання до лабораторних робіт, а також відповісти на питання в кінці кожної лабораторної роботи. Звіт повинен містити хід виконання завдань а також графічні матеріали, що підтверджують виконання цих завдань.

Лабораторна робота №1 Вивчення системи захисту інформації GnuPG та Kleopatra

Мета: вивчити роботу системи шифрування та підписування даних з відкритим кодом *GnuPG* (*GNU Privacy Guard*) та менеджера сертифікатів *Kleopatra*.

Завдання:

1. Вивчити принципи та методи роботи з системою захисту даних *GnuPG* та *Kleopatra*.
2. Перевірити їх роботу.

Теоретична частина

Система захисту інформації *GPG* призначена для захисту файлів, папок та листування з використанням поштових клієнтів, до прикладу, Thunderbird.

Система GPG розроблена Linux-спільнотою на противагу комерційної системи PGP і виконує подібні до останньої функції: шифрування/ розшифрування даних та підписування ЕЦП разом з генерування пар ключів та відповідних сертифікатів. Система постачається разом з менеджером сертифікатів Kleopatra, однак у випадку Лінукс може вимагати додаткового встановлення демона scdaemon, який виконує команди системи. Для встановлення демона необхідно у терміналі подати команду:

sudo apt install scdaemon

В ОС Windows усі компоненти системи встановлюються автоматично з пакету gpg4win.

Робота з програмою.

При запуску *Kleopatra* на екрані з'явиться форма, яка запропонує створити чи імпортувати пару ключів асиметричної криптосистеми (рис.1)

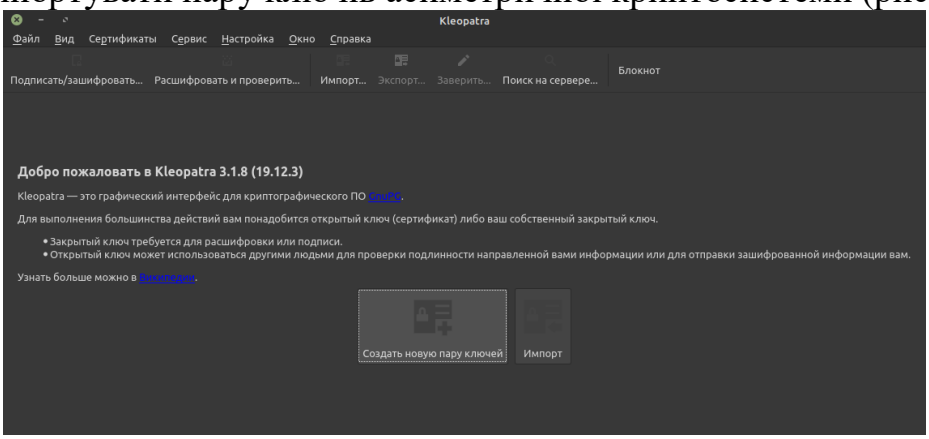


Рис.1. Форма для створення криптографічних ключів.

Якщо Ви запускаєте програму вперше, для її роботи необхідно натиснути кнопку «Создать новую пару ключей». На екрані з'являться формии майстра створення ключів, де Ви можете увести ім'я та прізвище, а також е-пошту (рис.2).

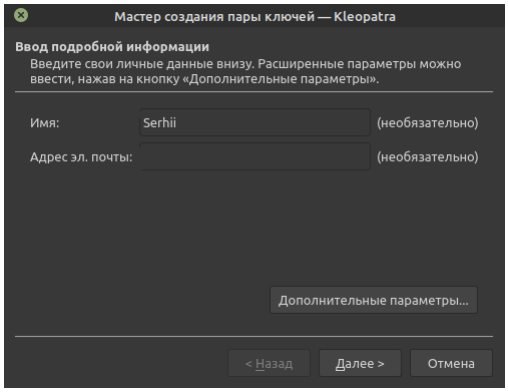


Рис.2. Форма для уведення параметрів користувача

Натиснувши кнопку «Дополнительные параметры», Ви можете налаштувати технічні параметри ключів: обрати потрібний алгоритм та довжину ключа, а також призначення та термін дії сертифікату (див. рис.3).

Після натискання на кнопку «ОК», майстер надасть Вам можливість перевірити правильність уведених даних, а після натискання на кнопку «Создать» (рис.4), - почне створювати потрібні ключі.

Для цього майстер попросить Вас хаотично порухати мишкою в межах поточної форми, накопичуючи при цьому випадкові дані у ключі (рис.5).

Коли ключі будуть сформовані, натисніть кнопку «Завершить», і майстер видасть на екран форму, де позначено створені криптографічні ключі (рис.6).

Виконавши ці процедури, Ви отримаєте дійсну пару ключів обраної асиметричної криптосистеми та сертифікат публічного ключа.

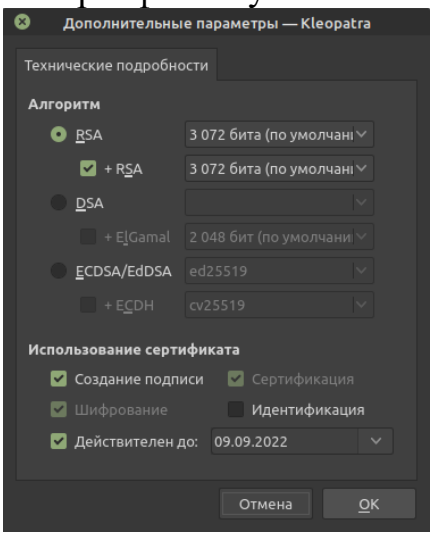


Рис.3. Форма для узгодження додаткових параметрів ключів.

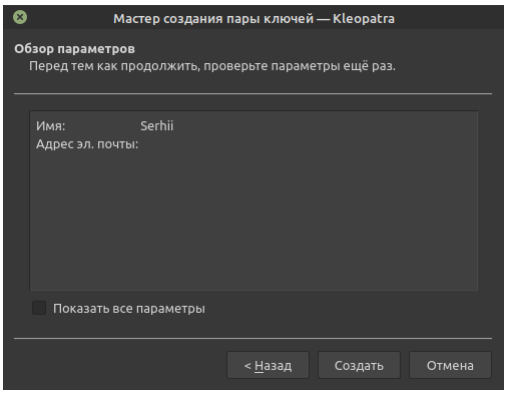


Рис. 4. Форма перевірки параметрів ключів.

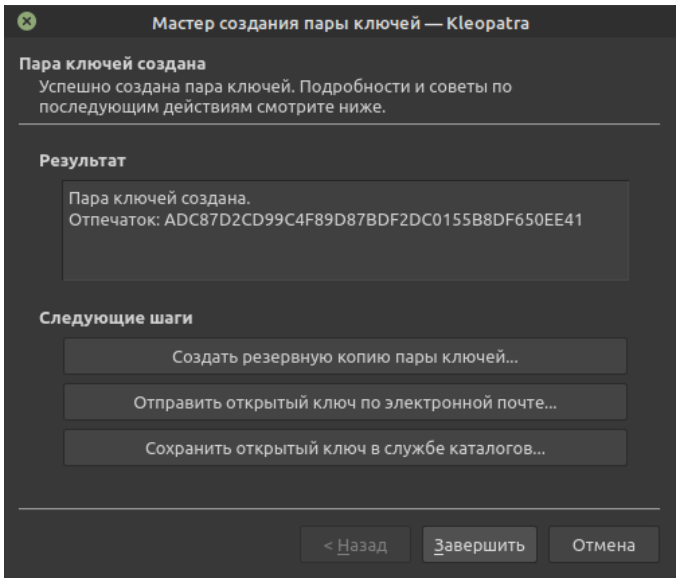


Рис. 5. Форма створення ключів.

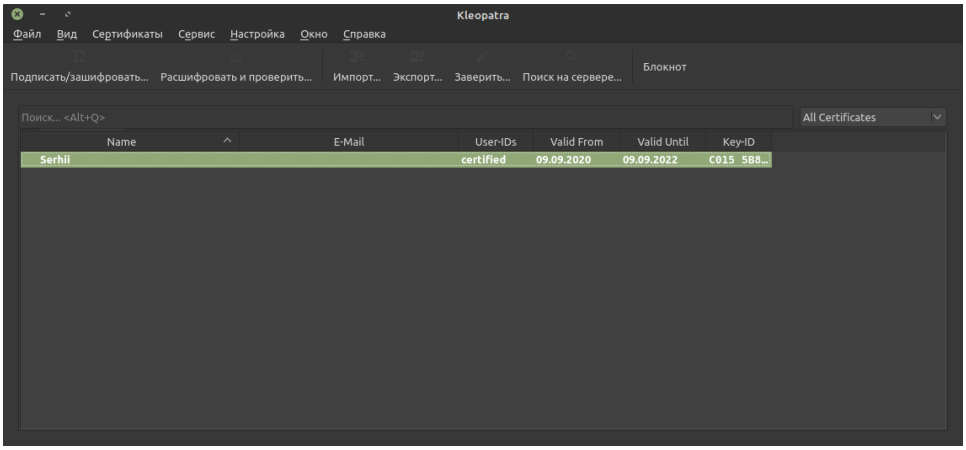


Рис. 6. Форма з назвами створених криптографічних ключів.

Шифрування та підписування файлів

Для зашифрування та накладання ЕЦП документів чи будь-яких файлів, на головній формі програми (рис.6) вибираємо закладку «Подписать/Зашифровать» і отримаємо форму створення ЕЦП та шифрування файлів (рис.7).

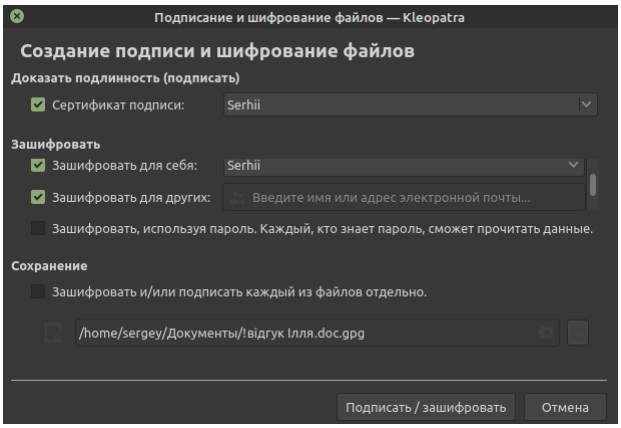


Рис.7. Форма для зашифрування/підписування файлів.

Тут можна вибрати файл/каталог для шифрування та ключ, на якому буде виконуватися шифрування/підписування. Після заповнення форми натисніть кнопку «Подписать/Зашифровать».

Розшифрування/Перевірка підпису відбуваються аналогічно. Для цього треба вибрати вкладку «Расшифровать и проверить подпись». У разі дійсного підпису та правильної розшифровки – отримаємо форму, подану на рис. 8.

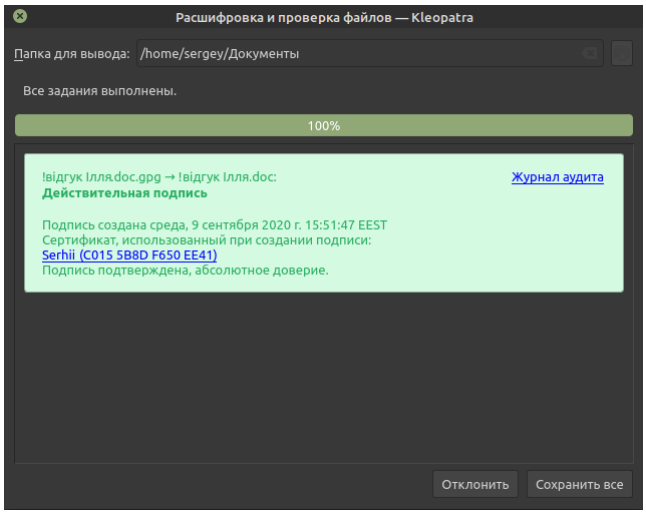


Рис.8. Перевірка підпису пройшла успішно.

Робота з буфером обміну.

Kleopatra також вміє працювати з буфером обміну: зашифровувати/розшифровувати та підписувати інформацію, що міститься у буфері обміну.

Після встановлення та запуску *Kleopatra* Ви побачите у системній корзині (System Tray) її іконку у вигляді стилізованої жіночої голови (див.рис. 9).

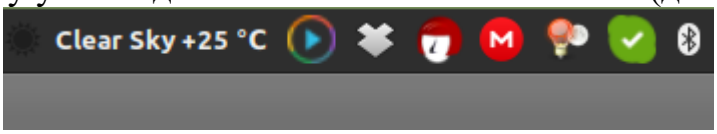


Рис. 9. Іконка *Kleopatra* у системній корзині

Натискаючи правою кнопкою мишки, Ви отримаєте меню, за допомогою якого можна виконати описані операції.

Розглянемо, як можна зашифрувати та підписати інформацію безпосередньо у буфері обміну.

Відкриваємо потрібний документ у текстовому редакторі (або в іншій програмі). Нехай це будуть оці методичні вказівки до лабораторної роботи. Вигляд документу у Microsoft Word подано на рис.10.

Виділимо увесь документ, натиснувши Ctrl+A та скопіюємо його до буферу обміну Ctrl+C.

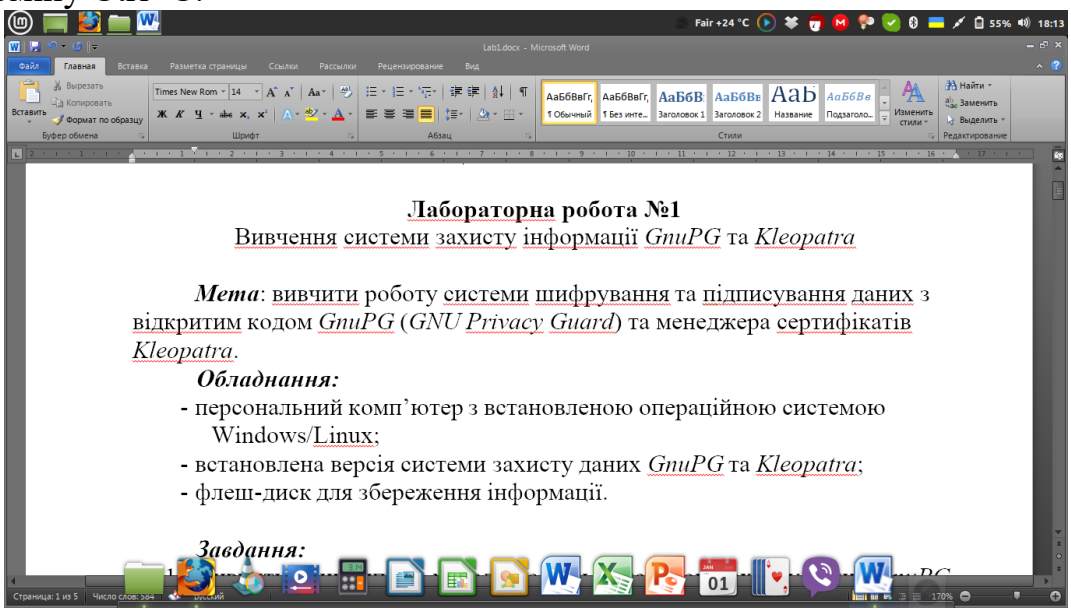


Рис.10. Відкритий документ у MS Word.

Клацнувши по іконці Kleopatra у системній корзині правою кнопкою мишки, вибираємо з меню «Буфер обмена» -> «Зашифровать…» і отримаємо таку форму для вибору ключа (рис.11).

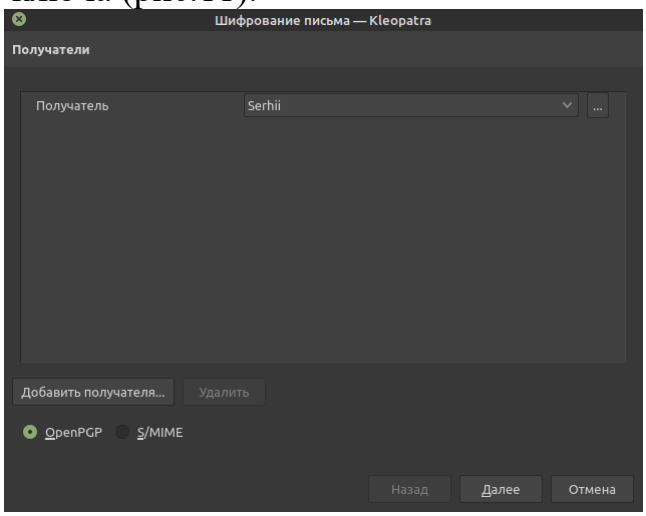


Рис.11. Форма для вибору ключа шифрування/підпису.

Тут ми маємо натиснути кнопку «Добавить получателя» і обрати ключ шифрування/підпису. Натискаємо кнопу «Далее» і отримаємо результат (рис.12). Натискаємо «ОК».

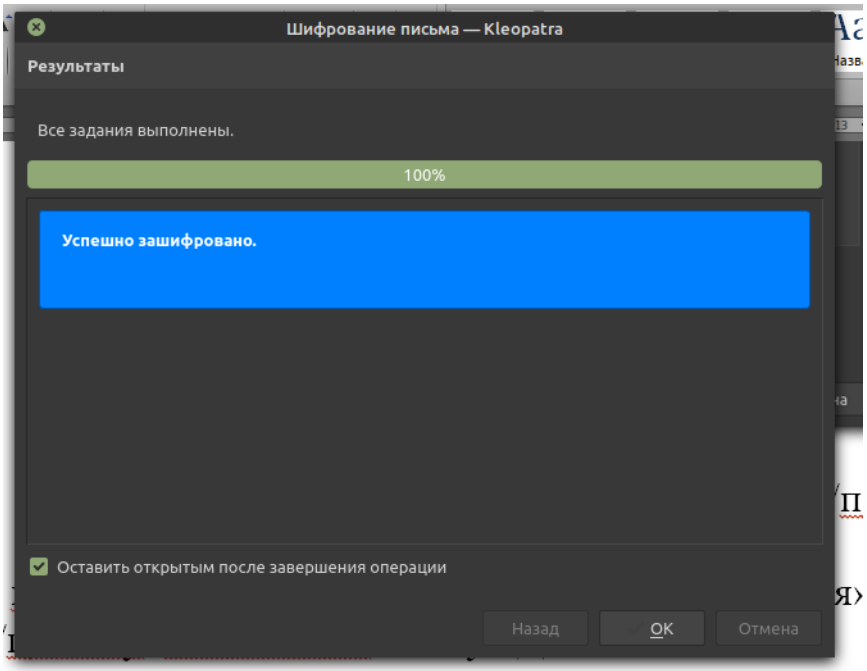


Рис. 12. Буфер успішно зашифровано обраним ключем.

Вставляємо зашифрований документ у нове вікно редактора Microsoft Word (рис.13).

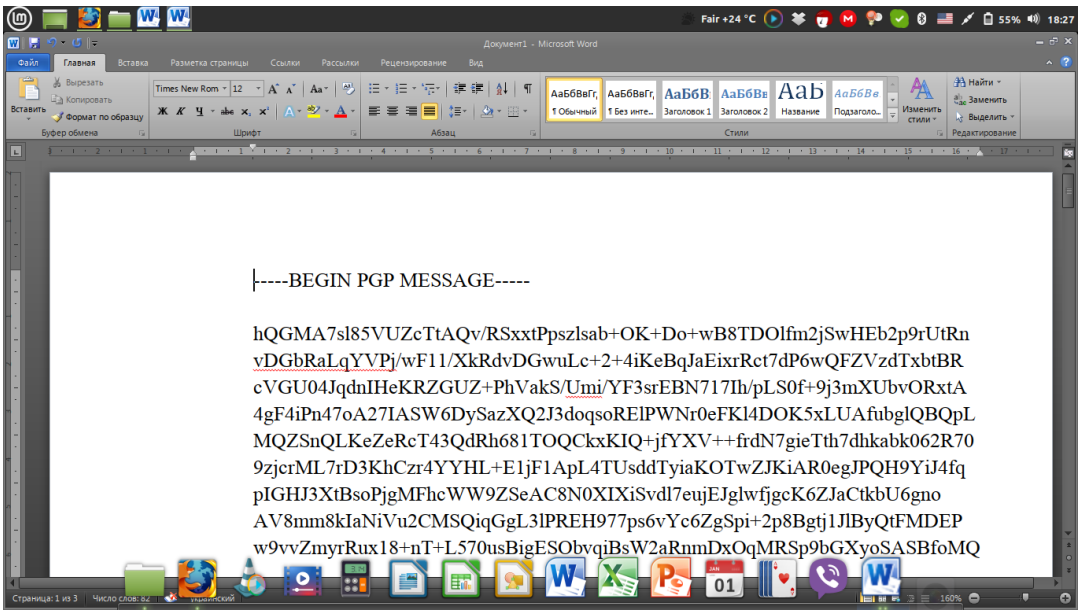


Рис.13. Так виглядає зашифрований документ.

Аналогічним чином можна створити електронний підпис документа, що міститься у буфері обміну, або зашифрувати та підписати документ одночасно та розшифровати/перевірити підпис.

Як видно з рис. 13, документ зашифровано асиметричним криптоалгоритмом, в даному випадку – RSA, оскільки саме його обуло обрано

під час створення ключової пари. Цифровий підпис також буде виконано цим алгоритмом.

Тим не менше, нагадаю, що для більшої безпеки необхідно використовувати різні ключі для підпису та шифрування, про що треба подбати на стадії генерування ключів.

Для використання цього програмного забезпечення у захищеному документообігу необхідно, щоби ключі шифрування/підпису були на обох кінцях системи інформаційного обміну. Для цього використовується імпорт/експорт ключів, що також передбачено у *Kleopatra*. Як виконати ці операції, розберіться, будь лакс, самостійно.

Для захисту е-пошти можна також використати *Kleopatra*, яка чудово інтегрується з поштовими клієнтами, наприклад, *Thunderbird.*

Практична частина.

1. Скачайте потрібні пакети з вказаних на початку методички ресурсів та встановіть *Kleopatra* разом з *GnuPG* на своєму комп'ютері.
2. Для захисту е-пошти скачайте та встановіть *Thunderbird* або аналогічний поштовий клієнт, з яким інтегрується *Kleopatra* (за бажанням!).
3. Запустіть *Kleopatra* та згенеруйте пару ключів для шифрування та окрему пару ключів для ЕЦП. Алгоритми та параметри – довільні.
4. Виконайте шифрування/розшифрування файлів та перевірте, що розшифрований файл ідентичний оригінальному. Зробіть висновок про коректність операцій шифрванні/розшифрування.
5. Підпишіть документи за допомогою ключів підпису та перевірте коректність процедури підпису.
6. Виконайте шифрування/розшифрування та підписування/перевірку підпису буферу обміну та поясніть отримані результати.
7. Складіть звіт з лабораторної роботи, який має містити протокол Ваших дій, висновки та відповіді на контрольні запитання.

Лабораторна робота №2 Вивчення системи захисту даних VeraCrypt 1.24.

Мета: Вивчити роботу системи захисту даних „*VeraCrypt 1.24*”.

Завдання:

1. Вивчити принципи та методи роботи з системою захисту даних „*VeraCrypt 1.24*”.
2. Перевірити її роботу.

Теоретична частина

Система захисту інформації *VeraCrypt 1.24* призначена для захисту даних на жорстких дисках комп‘ютера, а також на знімних носіях інформації типу USB-дисків, дискет та CD-ROM. *VeraCrypt 1.24* може використовуватися як на домашніх комп‘ютерах, так і робочих станціях або серверах, які обслуговують багато користувачів.

Принцип роботи *VeraCrypt 1.24* полягає у створенні в системі захищеного логічного диску. Захист цього диску ґрунтується на двох речах: а) паролі користувача; б) ключовому файлі; в) шифруванні даних диска у цілому. Дозволяється також створення прихованих дисків.

Інформація на утвореному захищеному диску шифрується за допомогою алгоритмів AES, Serpent, Blowfish та інших або комбінаціями цих алгоритмів. При монтуванні диску в систему порівнюються хеш-перетворення ключів за допомогою алгоритмів RipeMD-160, SHA-512 або Wirlpool.

Ключова послідовність генерується системою при так званій ініціалізації ключа. В сеансі ініціалізації ключ записується у файл. Після цього генерований ключ можна використовувати для криптографічних цілей.

Подальша робота з зашифрованим логічним диском звичайна, як зі звичайним логічним диском Windows.

Фізично створений диск являє собою звичайний файл на диску, який виступає логічним диском після монтування його в систему. При монтуванні система питає пароль і, якщо вказано, ключовий файл.

Якщо диск не підмонтований в систему, він виступає як файл. Його можна скопіювати, знищити і т.ін., але прочитати інформацію, яка зберігається в ньому, можна лише знаючи ключ.

Робота з програмою

1. *Встановлення та налаштування програми*

Встановлення *VeraCrypt* виконується стандартним чином, як і більшість програм Windows/Linux.

Для створення захищеного диску необхідно запустити встановлену програму VeraCrypt і натиснути кнопку «Create Volume» (рис.1).

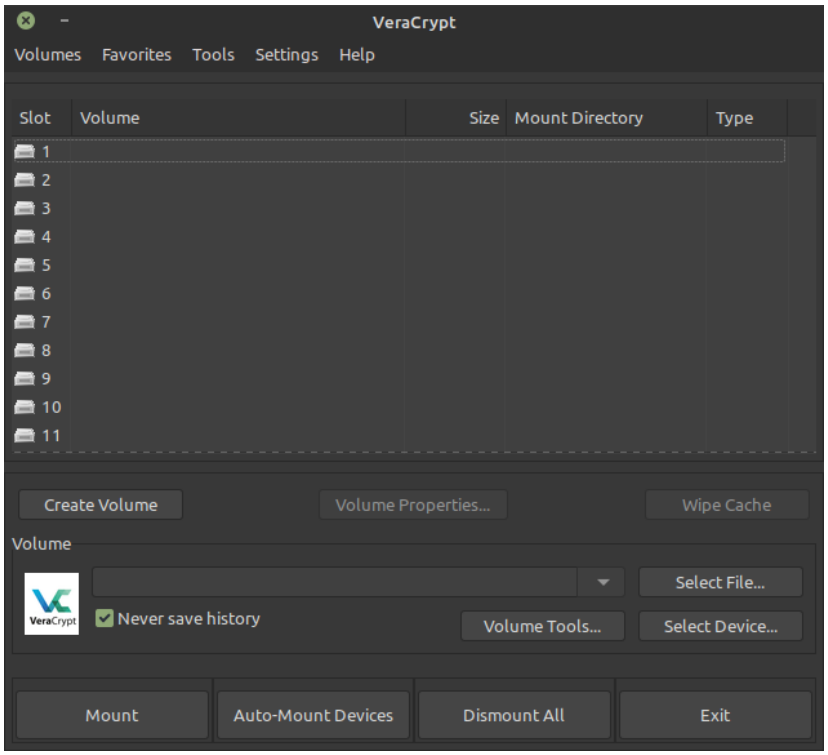


Рис.1. Головне вікно програми.

Обробник цієї кнопки запускає Майстра створення томів VeraCrypt (див. рис.2).

Рис.2. Майстер створення томів VeraCrypt.

Як бачимо з рисунку, VeraCrypt може створити віртуальний зашифрований диск, використовуючи файл на диску, зашифрувати несистемний або системний диск. Додатково можна створити приховану операційну систему.

Наступним кроком буде вибір типу зашифрованого диска. Виберіть «Standard VeraCrypt volume», як це показано на рис.3.

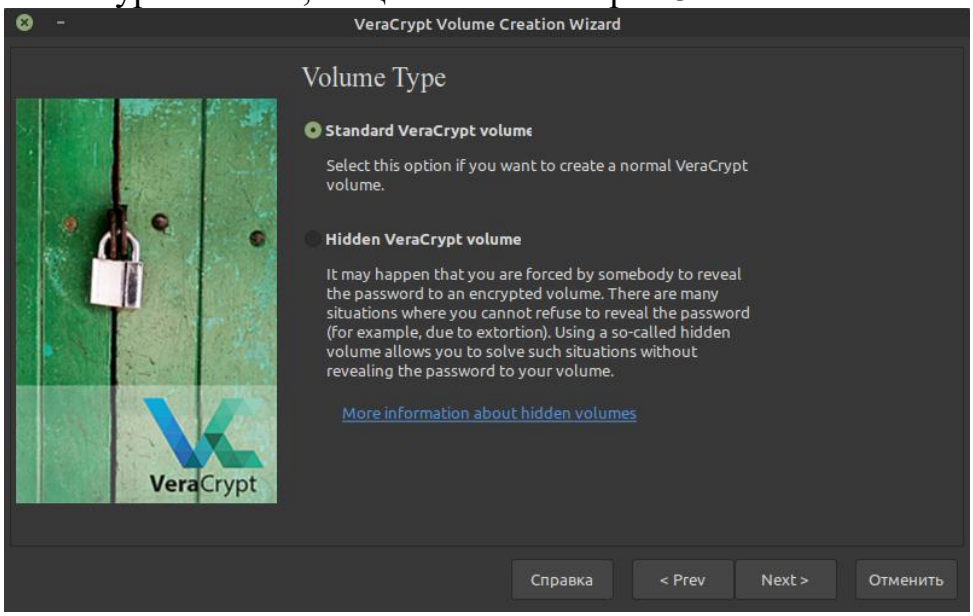


Рис.3. Вибір типу тома VeraCrypt.

Після натискання кнопки «Next» майстер надасть Вам можливість вибрати розміщення файлу для шифрованого диска, налаштувати алгоритм шифрування та хешування. Оберіть місцем розміщення захищеного диска Вашу флешку. Алгоритми шифрування та хешування можете обрати на свій розсуд (рис.4).

Рис.4. Вибір розміщення захищеного диску

Наступний крок – встановлення розміру захищеного диску (рис.5). Ви можете обрати довільний розмір контейнера, не менший визначеної величини для різних файлових систем.

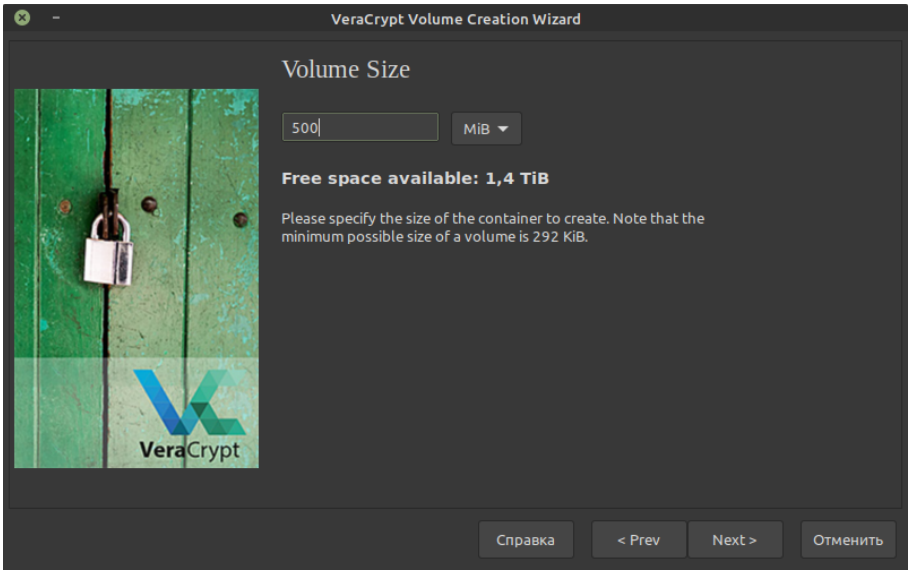


Рис.5. Вибір розміру тома.

Наступна форма майстра надає користувачу можливість встановити пароль для доступу до захищеного диску. Необхідно для більшої безпеки обрати пароль довжиною, більшою 20 символів. Додатковим засобом захисту може бути т.зв. ключовий файл. Його можна задати, якщо на формі відмітити «галочкою» чекбокс «Ключ. файлы» и натиснувши однойменну кнопку (див. рис.6), після чого майстер відкриє додаткову форму для вибору ключових файлів.

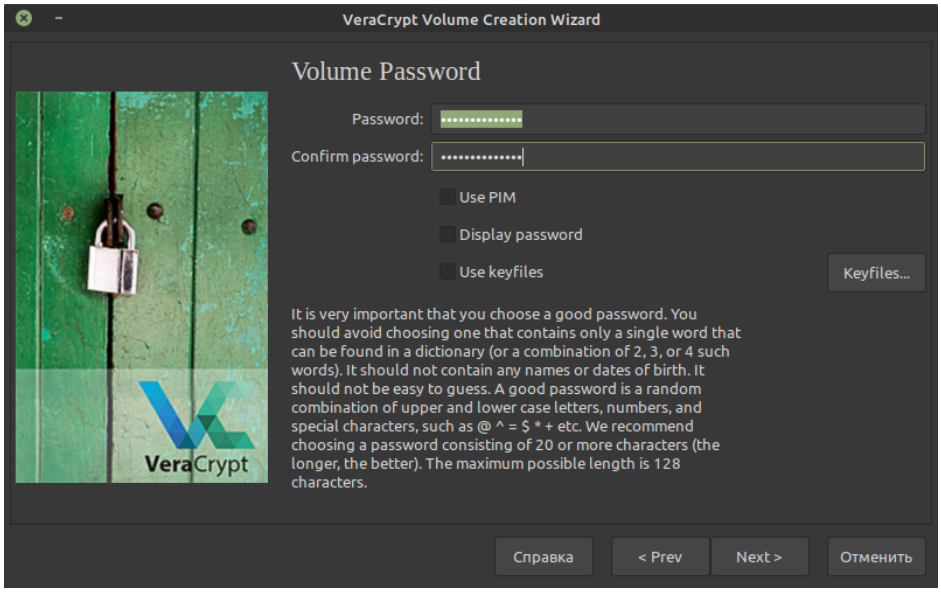


Рис. 6. Форма для встановлення пароля.

Після встановлення пароля та ключових файлів майстер запропонує Вам згенерувати ключі шифрування та відформатувати том. Для створення ключів

шифрування необхідно хаотично рухати мишкою в межах форми (див. рис.7) поки система буде генерувати ключі. Після обчислення ключів майстер відформатує том, і він буде готовий для використання.

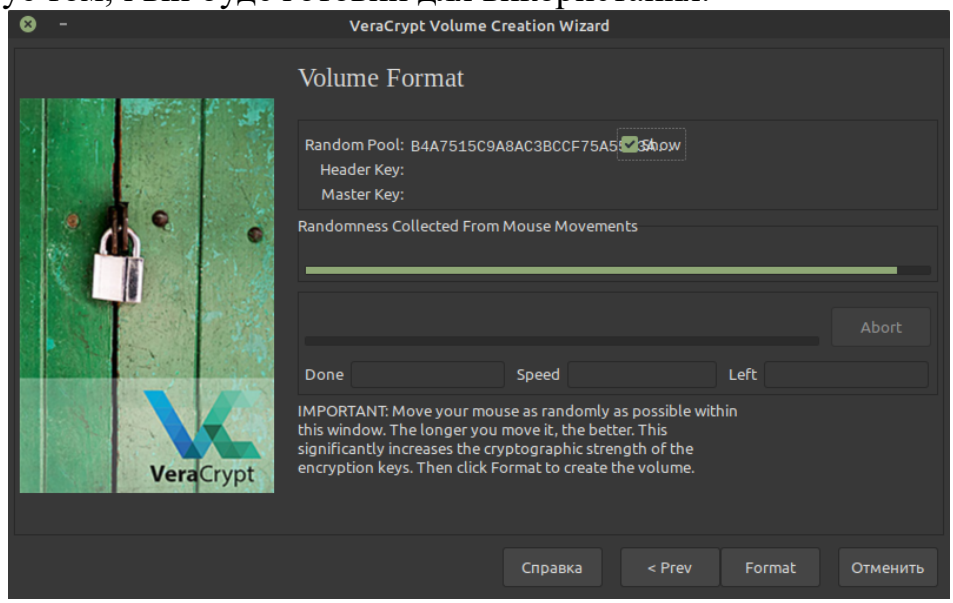


Рис.7. Форма для генерування ключів шифрування та форматування захищеного диска.

2. *Використання захищеного диска.*

Для використання створеного захищеного диска необхідно запустити VeraCrypt та виконати наступні дії:

- вибрати вільну літеру/вільний слот (Linux) для подальшого монтування диска;
- натиснути кнопку «Select File» і вибрати підготовлений файл, підготовлений при налаштуванні системи;
- натиснути кнопку «Mount»;
- система спитає пароль до файлу, а також ключові файли, якщо їх було задано при налаштуваннях захищеного диска (див.рис.8).

Тепер захищений диск готовий до роботи.

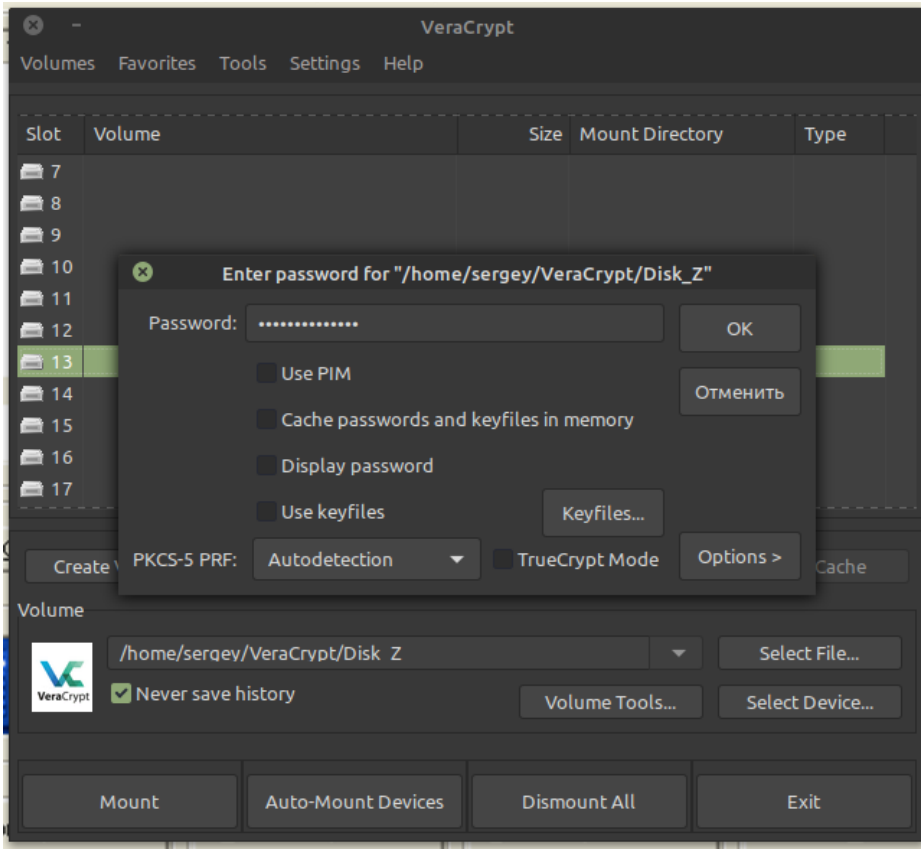


Рис.8. Форма для підключення захищеного диска до системи.

Практична частина

1. Встановіть програму *VeraCrypt 1.24* на Вашому комп'ютері, якщо її ще не встановлено для Вашого користувача (Посилання для скачування: https://www.veracrypt.fr/en/Downloads.html)
2. Згідно з вказівками теоретичної частини створіть на Вашому флеш-носієві захищений диск. Для захисту використайте пароль та ключовий файл.
3. Перезавантажте комп'ютер та підмонтуйте захищений диск, користуючись паролем та ключовим файлом.
4. Створіть текстовий файл у Microsoft Word (наприклад, звіт з лабораторної роботи) та збережіть його на захищеному диску. Копію файлу збережіть на відкритому диску. Відкрийте файли за допомогою TextView (HEX-mode) у FAR, порівняйте їх вміст та зробіть відповідний висновок. Чому отримано такий результат?
5. Спробуйте прочитати файл, який „символізує" захищений диск, спробуйте його витерти або змінити. Поясніть отриманий результат.
6. Відмонтуйте диск та спробуйте продивитися файл захищеного диску в HEX-редакторі. Поясніть отриманий результат.
7. Підготуйте звіт з лабораторної роботи. Звіт повинен містити: а) протокол Ваших дій; б) результати Ваших дій та пояснення цих результатів у пп. 4-6; в) відповіді на контрольні запитання. Для

отримання відповідей на контрольні запитання використайте довідку VeraCrypt та сайт програми.

Лабораторна робота №3 Дослідження захисту інформації у спрощених EDI-системах

Мета: створити спрощену систему off-line обміну захищеною банківською інформацією та дослідити її роботу

Завдання:

1. Будь-якою мовою програмування розробити спрощену систему захищеного обміну банківською інформацією з використанням Microsoft CryptoAPI.
2. Перевірити її роботу.

Теоретична частина

EDI-системи (Electronic Data Interchange) – один з перших засобів захисту банківської інформації, яка існує вже багато десятиліть. Вперше такі системи захисту були застосовані для захисту повідомлень у off-line обміні банківської інформації.

Сьогодні в системах EDI широко використовуються біля 20 стандартів, основними серед яких є два: UN/EDIFACT и ANSI X-12. Перший використовується, в основному у Європейських країнах, а останній – у Сполучених Штатах Америки.

Для впорядкування різних стандартів у 1996 році Економічною та Соціальною Радою ООН було розроблено Рекомендацію №25 щодо використання стандарту EDIFACT, де рекомендовано модернізувати існуючі системи згідно з стандартом UN/EDIFACT, а нові системи будувати тільки на основі цього стандарту.

Акронім UN/EDIFACT розшифровується так: *United Nations rules for Electronic Data Interchange for Administration, Commerce and Transport – Правила ООН електронного обміну документами для державного управління, торгівлі та транспорту.*

Детальніше ці стандарти описано у книзі.

У цій лабораторній роботі ми будемо використовувати спрощену систему сегментів українською мовою «власної» розробки. Це суто навчальна система, яка у реальному житті не використовується. Однак вона полегшить розуміння значення кожного сегмента та загалом усієї системи захисту.

Назви та значення сегментів цієї «навчальної» системи подано у табл.1. У табл. 2 подано назви банківських документів, як вони повинні подаватися у відповідних сегментах повідомлень.

Таблиця 1. Назви та значення сегментів

ЗАГ	Заголовок пакета	КІН	Кінець пакета
ПОЧ	Початок повідомлення	КІП	Кінець повідомлення
НАЗ	Назва	ДЧП	Дата/Час/Період
ПОК	Покупець	ОПП	Опис пункту

ПЛА	Платник	ВАЛ	Валюта оплати
ОТР	Отримувач	ЦІН	Ціна замовлення
КІЛ	Кількість товару	+	Розділовий символ
'	Знак закінчення сегмента	ДОК	Назва документа
ТОВ	Товар	ОДВ	Одиниця виміру
СУМ	Сума	БКВ	Банк відправника
РХВ	Рахунок відправника	БКО	Банк одержувача
РХО	Рахунок одержувача	МФВ	МФО банку відправн.
МФО	МФО банку одержувача	ВІН	Вектор ініціалізації
ШИФ	Алгоритм шифрування	КЛШ	Ключ шифрування
АЦП	Алгоритм цифрового підпису	КЦП	Ключ для перевірки ЦП

Таблиця 2. Назви банківських документів

ОПЛ	Оплата	НАК	Накладна

Припустимо, що ТОВ «Кооператор» бажає заплатити ТОВ «Калинівський ринок» 10 тис.грн. за товар по рахунку №23 від 10.09.2010 р. Банківські реквізити сторін такі:

Платник – ТОВ «Кооператор»:
Рахунок – 2600123456789;
Банк – Чернівецьке відділення КБ «Приватбанк»;
МФО – 356032.

Отримувач – ТОВ «Калинівський ринок»:
Рахунок – 2600987654321;
Банк – ЧФ АКБ «Укрексімбанк»;
МФО – 356026.

Сформуємо повідомлення з таких даних:
ЗАГ+0002' – заголовок повідомлення та його номер;
ПОЧ' – початок повідомлення;
ДЧП+20100910:1048:24' – дата, час та період дії повідомлення: дата – рік (2010), місяць (09), день (10): час (10 годин 48 хвилин): період дії повідомлення (24 години);
ДОК+ОПЛ' – документ – платіжне доручення;
ПЛА+ТОВ «Кооператор»' – назва платника;
БКВ+Чернівецьке відділення КБ «Приватбанк»' – банк платника;
МФВ+356032' – МФО банку платника;
РХВ+2600123456789' – банківський рахунок платника;
ОТР+ТОВ «Калинівський ринок»' – назва отримувача;
БКО+ЧФ АКБ «Укрексімбанк»' – банк отримувача;
МФО+356026' – МФО банку отримувача;

РХО+2600987654321' – банківський рахунок отримувача;

ОПП+Оплата за товар по рахунку №23 від 10.09.2010 р.' – призначення платежу;

ВАЛ+грн.' – валюта оплати;

СУМ+10000' – сума оплати;

КІП' – Кінець повідомлення;

КІН+0016' – кінець пакета+загальна кількість сегментів у ньому (для перевірки.

Отже, пакет для такої задачі буде мати вигляд:

ЗАГ+0002'ПОЧ'ДЧП+20100910:1048:24'ДОК+ОПЛ'ПЛА+ТОВ «Кооператор»'БКВ+Чернівецьке відділення КБ «Приватбанк»'МФВ+356032' РХВ+2600123456789'ОТР+ТОВ «Калинівський ринок»' БКО+ЧФ АКБ «Укрексімбанк»'МФО+356026'РХО+2600987654321'ОПП+Оплата за товар по рахунку №23 від 10.09.2010 р.'ВАЛ+грн.'СУМ+10000'КІП'КІН+0016'

Цей пакет тепер можна захистити від несанкціонованого доступу, наприклад, зашифрувавши його, та відправити отримувачу.

Отримавши такий пакет, програмне забезпечення банку розшифровує його, проводить аналіз сегментів, та формує відповідний документ (чи кілька документів, якщо їх міститься кілька у пакеті – таку ситуацію ми розглядати не будемо).

Ключ шифрування, як правило передають двома способами: разом з повідомленням в окремих сегментах; та окремим повідомленням, яке називається тоді повідомленням захисту. Ми будемо користуватися останнім способом.

Сеансовий ключ шифрування захищається публічним ключем отримувача асиметричної системи шифрування.

Сегменти повідомлення захисту мають приблизно такий вигляд:

ЗАГ+0001' – заголовок повідомлення та його номер;

ПОЧ' – початок повідомлення;

ДЧП+20100910:1030:24' – дата, час та період дії повідомлення: дата – рік (2010), місяць (09), день (10): час (10 годин 30 хвилин): період дії повідомлення (24 години);

ОПП+0002' – яке повідомлення зашифровано цим ключем;

ШИФ+DES' – алгоритм шифрування, який використано для захисту повідомлення №0002;

КЛШ+13FA78BB0FA96C4D' – ключ шифрування;

ВІН+32AFBC9832F2D5CA2' вектор ініціалізації;

КІП' – кінець повідомлення;

КІН+0008' – кінець повідомлення+кількість сегментів у ньому.

Отже, повідомлення захисту повинно мати приблизно такий вигляд:

ЗАГ+0001'ПОЧ'ДЧП+20100910:1030:24'ОПП+0002'ШИФ+DES' КЛШ+13FA78BB0FA96C4D' ВІН+32AFBC9832F2D5CA2'КІП'КІН+0008'

Таке повідомлення шифрується на публічному ключі отримувача.

Практична частина

Для виконання цієї лабораторної роботи необхідно встановити на Вашому комп'ютері криптографічний інтерфейс від Microsoft – CryptoAPI, який постачається з Microsoft Visual Studio або аналогічний. Детальніше CryptoAPI описано у Лабораторній роботі №10 практикуму з курсу «Криптографія та побудова систем безпеки» або у книзі.

1. Ознайомтеся з можливостями CryptoAPI.
2. Бригадою з двох студентів напишіть програму, яка реалізує обмін захищеними повідомленнями між клієнтською та серверною частинами. Один студент з бригади розробляє клієнтську частину, а другий – серверну.
3. Клієнтська частина повинна задовольняти наступним вимогам:
    a. Надавати можливість користувачеві вибрати тип документа, що формується.
    b. Надавати можливість уведення даних, необхідних для заповнення полів обраного документа: назви отримувача та відправника, їх банківських реквізитів, товари та їх параметрів, суми тощо. Сукупність даних формується в залежності від обраного документа.
    c. Формувати пакет даних для передавання серверній частині згідно з описом, поданим у теоретичній частині.
    d. Забезпечувати захист інформації за допомогою симетричної системи шифрування. Систему шифрування оберіть згідно Вашого варіанту.
    e. Забезпечувати механізм передавання ключа шифрування у спосіб, який описано у теоретичній частині.
    f. Забезпечувати передавання інформації серверній частині за допомогою технології сокетів.
4. Серверна частина повинна задовольняти таким вимогам:
    a. Генерувати пару ключів асиметричної системи шифрування. Система шифрування обирається згідно Вашого варіанту.
    b. Забезпечити передавання публічного ключа шифрування клієнтській частині.
    c. Забезпечувати розшифрування отриманої від клієнтської частини інформації.
    d. Забезпечувати перевірку правильності кількості сегментів у пакеті та попереджати користувача про втрати інформації в разі невідповідності підрахованої кількості сегментів заявленій.

е. Забезпечувати формування необхідних документів з використанням отриманої від клієнтської частини інформації та візуалізацію отриманої інформації.

5. Продемонструйте працездатність системи передавання даних згідно свого варіанту.

Таблиця 3. Варіанти лабораторної роботи

№ варіанта	Симетричний алгоритм	Обмін ключами
1	DES	Діффі-Хеллман
2	3DES (2 ключі)	RSA
3	3DES (3 ключі)	Діффі-Хеллман
4	RC2	RSA
5	RC4	RSA

Оформіть звіт з лабораторної роботи.

Звіт повинен містити:

1) код клієнтської або серверної частин програми (в залежності від ролі студента у бригаді);
2) порівняння документів клієнтської та серверної частини;
3) обговорення стійкості застосованих засобів захисту;
4) блок-схему;
5) діаграму класів або модулів, або data-flow diagram (на вибір) та діаграму прецедентів розробленого ПЗ (виконуються за допомогою UML – ПЗ);
6) відповіді на контрольні запитання;
7) висновки з лабораторної роботи.

Лабораторна робота №4 Розробка системи «Банкоматик»

Мета: Вивчити роботу спрощеної системи захисту інформації сучасних банкоматів.

Завдання:

1. Будь-якою мовою програмування розробити спрощену систему емуляції роботи банкоматів «Банкоматик» з використанням Microsoft CryptoAPI.
2. Перевірити її роботу.

Теоретичні відомості

1. Засоби ідентифікації пластикових карток

Пластикова картка являє собою пластинку стандартизованих розмірів (85.6×53.9×0.76мм), виготовлену зі спеціальної, стійкої до механічних та термічних дій пластмаси.

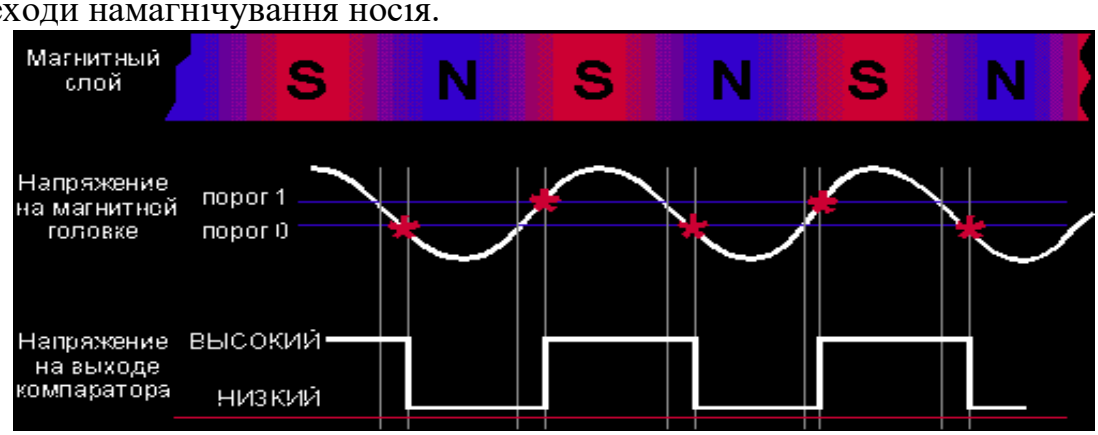
Одна з основних функцій пластикової картки – забезпечення ідентифікації її власника як суб'єкта платіжної системи. Для цього на пластикову картку нанесено логотипи банку-емітента та платіжної системи, яка обслуговує картку, ім'я власника картки, номер його рахунку, термін дії картки і т.ін. Крім цього, на картці може бути фотографія власника картки та його підпис. Алфавітно-цифрові дані - ім'я, номер рахунку та ін., можуть бути ембосовані. Це дає можливість швидко перенести дані на чек за допомогою спеціального пристрою, який «прокатує» картку (аналогічно до копіювання при друкуванні на друкарській машинці)

Графічні дані забезпечують можливість візуальної ідентифікації картки. Для використання картки у платіжній системі необхідно зберігати дані на картці у форматі, який дозволяє виконувати процедуру автоматичної аутентифікації. Ця задача може бути вирішена з використанням різних фізичних механізмів.

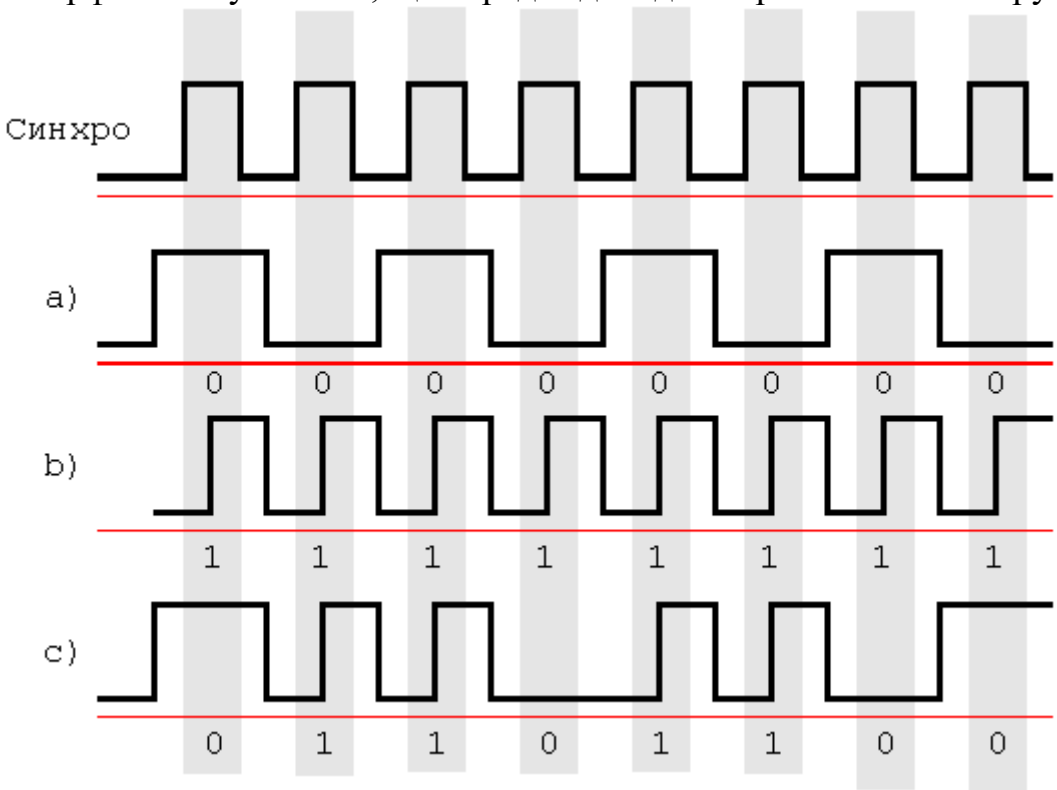
Картки зі штрих-кодом використовують код, аналогічний тому, що використовують для маркування товарів. Як правило кодова стрічка закривається непрозорим матеріалом, і зчитування відбувається в інфрачервоних променях. Картки зі штрих-кодом досить дешеві та прості у виготовленні. Остання особливість зумовлює їх слабкий захист від підробки та робить їх мало придатними для використання у платіжних системах.

Картки з магнітною стрічкою сьогодні найбільш розповсюджені – у вжитку знаходяться понад 2 млрд. подібних карток. Магнітна стрічка розміщується на зворотній стороні картки та, згідно зі стандартом ISO 7811, складається з трьох доріжок. Перші дві призначені для зберігання ідентифікаційних даних, а на третю можна записувати інформацію (наприклад, поточне значення суми грошей на рахунку). Однак внаслідок малої надійності запису на магнітну стрічку, режим запису, як правило, не практикується, а так картки використовуються лише у режимі читання.

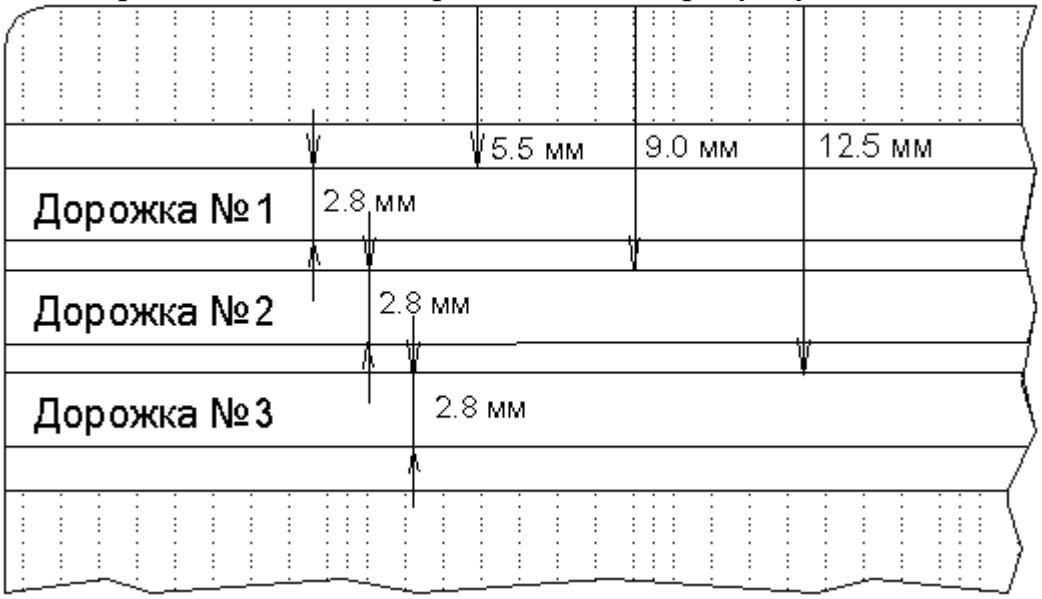
Як же влаштована картка? Принцип магнітного запису на карту нічим не відрізняється від звичайного звукозапису. Знищення інформації можна виконувати постійним магнітом з концентратором магнітного потоку. Запис виконують без підмагнічування, оскільки тоді досягаються більш різкі переходи намагнічування носія.



Цифрові дані, зчитані з картки, подаються на стандартний послідовний інтерфейс типу RS-232, що передає дані для обробки комп'ютеру.



Якщо в момент синхровідліку рівень сигналу не змінюється, то він вважається рівним нулю, а якщо має перепад – тоді одиниці. Типове розміщення доріжок на магнітній стрічці банківської картки подано на рисунку.



Якщо картку розмістити магнітною стрічкою до себе так, щоби стрічка була знизу картки, то дані пишуться зліва направо.

Захищеність карток з магнітною стрічкою значно вища, ніж у карток з штрих-кодом. Однак і такий тип карток відносно вразливий для шахрайства. Тим не менше, розвинена інфраструктура існуючих платіжних систем є причиною інтенсивного використання карток з магнітною стрічкою. Відмітимо, що для підвищення захисту карток використовують додаткові графічні засоби: голограми, нестандартні шрифти для ембосування.

На лицевій стороні картки з магнітною стрічкою зазвичай вказується логотип банку-емітента, платіжної системи, номер картки (перші 6 цифр – код системи і банку, наступні 9 – банківський номер картки, остання цифра – контрольна. Останні 4 цифри нанесено на голограму. Далі вказують термін дії картки, ім’я власника картки. На звороті розміщується магнітна стрічка та місце для підпису.

2.Використання криптографії у картках з магнітною стрічкою

Найпоширеніше використання криптографії – це забезпечення PIN-коду для ідентифікації користувача в місцях, де не можна забезпечити контроль доступу, наприклад в АТМ (банкоматах), або в інших ситуаціях, де неможливо подати звичайний паперовий підпис.

Друге за розповсюдженістю використання криптографії – це механізми контролю за оригінальністю магнітної стрічки. Це спрямовано проти загрози використання підробних карток, коли на стрічку записується значення, яке не відповідає тій інформації, яку видно на лицеві стороні картки. Коли картка перевіряється у режимі on-line, це значення може підтверджувати істинність

картки. Для цього існує кілька різних стандартів. Найбільш популярні – це Visa Card Verification Value (CVV) або її аналог для MasterCard, CVC.

2.1. Обробка PIN

Принцип захисту за допомогою PIN ґрунтується на тому факті, що ніхто, окрім власника картки, не знає цього коду. Тому вимоги до PIN такі:

- він не повинен зберігатися у відкритому вигляді;
- PIN не можна отримати на основі інформації на магнітній стрічці або бази даних.

Зазвичай, PIN - це 4-значне число, але зараз зустрічаються і 5-значні PIN-коди.

Для підтримки PIN виконуються такі обчислення:

- Генерується 4-значне число - це PIN;
- PIN комбінується з іншою інформацією, наприклад, з номером рахунку, щоб створити блок даних для процесу шифрування;
- Цей блок тричі шифрується на робочих ключах PIN;
- З отриманого результату обираються деякі цифри. Вони і є Pin Verification Value (Число Перевірки PIN) або Pin Offset (Зміщення PIN);
- Зміщення PIN зберігається;
- Друкується захищений конверт з PIN;
- Пам'ять очищується нулями, щоби приховати усі сліди існування PIN.

На цьому етапі єдине місце, де знаходиться відкрите значення PIN – це конверт, а сам PIN не можна отримати зі зміщення PIN.

Коли картка використовується, власник вводить PIN-код, а зміщення обчислюється та порівнюється з тим, що зберігається у базі даних комп'ютера. Отже і у цьому разі PIN-код не передається мережами у відкритому вигляді.

Ще раз підкреслюємо, що зміщення складається з цифр, які вибрано з шифрованих даних. Зазвичай це 4-6 цифр, знаючи які неможливо відновити власне PIN.

2.2. Обробка CVV

Загрози безпеці інформації з боку шахраїв призвело до необхідності введення додаткової аутентифікації карток відносно платіжної системи, так званого числа перевірки картки (Card Verification Value). CVV – це складна для обчислення послідовність цифр, яка створюється зашифруванням певної інформації. CVV записано на магнітну стрічку картки, так що збирання візуальної інформації про власника картки та власне про картку нічого зловмисникові не дає.

Для утворення CVV комбінуються статичні дані, наприклад, номер рахунку, тричі шифрується на ключах Card Verification. З утвореного результату обираються цифри для створення CVV та записуються на магнітну стрічку.

Отже, CVV надає додатковий рівень захисту картки від підробки. Треба мати на увазі, однак, що цей спосіб не захищає від такої атаки, як збирання даних про картки за допомогою фальшивих банкоматів.

Існує ще один варіант CVV - CVV2, який використовується для авторизації телефоном. Він розраховується приблизно за таким самим алгоритмом, як і CVV, а результат друкується на звороті картки. Ці цифри можуть запитувати при виконанні трансакцій по телефону для перевірки легітимності операції.

2.3. Приклад шифрування при зніманні готівки у банкоматі

Звичайна ATM-транзакція виглядає таким чином:
- Клієнт надає картку банкомату;
- Клієнт вводить свій PIN –код;
- Клієнт робить запит на готівкову суму;
- Транзакцію підтверджено, готівку видано.

Цей процес включає досить багато шифрувань та перевірок.
1. Клієнт надає картку банкомату:
Магнітна стрічка зчитується та дані зберігаються у захищеному пристрої банкомату.
2. Клієнт вводить свій PIN:
Він вводиться у захищений апаратний модуль.
3. Клієнт робить запит на готівкову суму:
Банкомат створює повідомлення, що шифрується на ключі терміналу;
Зашифроване повідомлення відправляється банку.
Це повідомлення розшифровується у банку, обчислюється CVV та порівнюється зі значенням на магнітній стрічці.
Розшифровується зміщення PIN та порівнюється з тим, що записано у базі даних банку.
4. Транзакцію підтверджено, готівку видано.

Зауваження: усі функції шифрування зазвичай виконуються у захищеному апаратному модулі. Ніякі відкриті значення не передаються іншим програмам або назовні захищеного модуля.

Практична частина.

1. Бригадою з двох студентів напишіть програму «Банкоматик», яка спрощено емулює роботу банкомату з використанням криптографічного інтерфейсу CryptoAPI (або аналогічного). Один учасник бригади пише клієнтську частину програми, а другий – серверну.
2. В якості карток використайте флеш-диск.
3. Вимоги до серверної частини:
   - Створює системну базу даних, яка складається з:
     a. Card Verification Value (Число перевірки картки);
     b. Pin Verification Value (Число перевірки ПІН);
     c. Номер банківського рахунку користувача;
     d. Сума на банківському рахунку користувача.

- Виконує реєстрацію нової картки та записує інформацію на картку.
- Виконує перевірку картки за CVV після отримання інформації від клієнтської частини;
- Виконує перевірку користувача за PVV після отримання інформації від клієнтської частини;
- Виконує операції по зарахуванню грошей на рахунок користувача;
- Виконує операції по списанню коштів з рахунку користувача;
- Надає дозвіл клієнтській частині на видачу готівки власнику карти.

Обчислення PVV.

Використайте генератор випадкових чисел з набору CryptoAPI для генерування 4-значного десяткового числа, яке буде служити ПІН-кодом. Це число повідомляється користувачу (запишіть його у текстовий файл). PVV обчислюється потрійним хешуванням цього числа одним з алгоритмів хешування з набору CryptoAPI. З отриманого хеш-образу обирається половина знаків (на Ваш розсуд: старша, молодша половина, або парні, непарні символи). Отримане число зберігається у системній базі даних в якості PVV і використовується при подальшій аутентифікації власника карти.

Обчислення CVV.

Для ідентифікації самої картки використовують т.зв. CVV (Card Verification Value). Використайте генератор CryptoAPI для генерування випадкового числа з 16 десяткових знаків (номера картки) та 14-значне число, яке буде зображати номер банківського рахунку корситувача. Сповістіть їх власнику картки (запишіть у файл на диску). CVV обчислюється потрійним хешуванням числа, яке утворюється конкатенацією номера картки та номера банківського рахунку. Отриманий хеш-образ вкорочується вдвічі аналогічно до ПІН-коду і записується на картку та у системну базу даних.

При реєстрації нової картки у системну базу даних, яку обслуговує серверна частина, також записуються номер банківського рахунку користувача у відкритому вигляді. Системна база даних зберігається на захищеному комп'ютері у банку.

4. Вимоги до клієнтської частини програми:
   - Приймати дані від картки (USB-флешки) та передавати їх серверній частині;
   - Приймати ПІН-код від користувача, формувати PVV та передавати його серверній частині;
   - Отримувати від серверної частини дозвіл на операції з банківським рахунком користувача (отримання готівки).
   - Файл з реєстраційними даними на флеш-диску містить такі дані:
     a. Card Verification Value (Число перевірки картки);
     b. Сума на рахунку користувача.

Робота з карткою.

Після того, як користувач вставить картку в кард-рідер, клієнтська частина програми повинна знайти файл з реєстраційними даними, прочитати їх та запросити користувача увести ПІН-код.

Після отримання даних від користувача, клієнтська частина виконує такі операції:

i. Відправляє CVV серверній частині та отримує результат перевірки істинності картки;
ii. Формує PVV, відправляє його серверній частині і отримує результат аутентифікації власника картки;
iii. В разі успішної перевірки картки та її власника клієнтська частина запитує користувача про суму, яку він хоче зняти з рахунку, перевірити, чи не перевищує вона залишку на банківському рахунку користувача, і видає готівкою ці кошти.
iv. Записує залишок по рахунку на картку та передає серверній частині для запису в системну базу даних.

5. Відладьте програму, користуючись інтерфейсом "зворотної петлі" (loopback), IP-адреса 127.0.0.1.
6. Виконайте програму з фізично розподіленими по двох комп'ютерах клієнтом і сервером.

Для підсилення захисту інформації під час роботи з банкоматом Ви можете використати будь-який метод суцільного шифрування інформації, яка передається між клієнтською та серверною частинами.

Підготуйте звіт з лабораторної роботи, який повинен містити:

1) код клієнтської або серверної частин програми (в залежності від ролі студента у бригаді);
2) склад інформації у файлі на «картці» та в системній базі даних;
3) протокол роботи «Банкоматика»;
4) відповіді на контрольні запитання;
5) висновки з лабораторної роботи.

Лабораторна робота №5 Вивчення захисту повідомлень в протоколі SET

Мета: Вивчити методи захисту повідомлень в протоколі SET.

Завдання:

1. Будь-якою мовою програмування розробити просту систему захисту повідомлень з використанням процедур захисту, які використовуються в протоколі SET.
2. Для збереження ключової інформації використайте флеш-диск.
3. Перевірити роботу системи.

Теоретичні відомості

Протокол SET був розроблений у 1997 році завдяки спільним зусиллям Visa, MasterCsrd та Netscape. Він призначений для забезпечення безпеки електронних платежів за допомогою банківських карток.

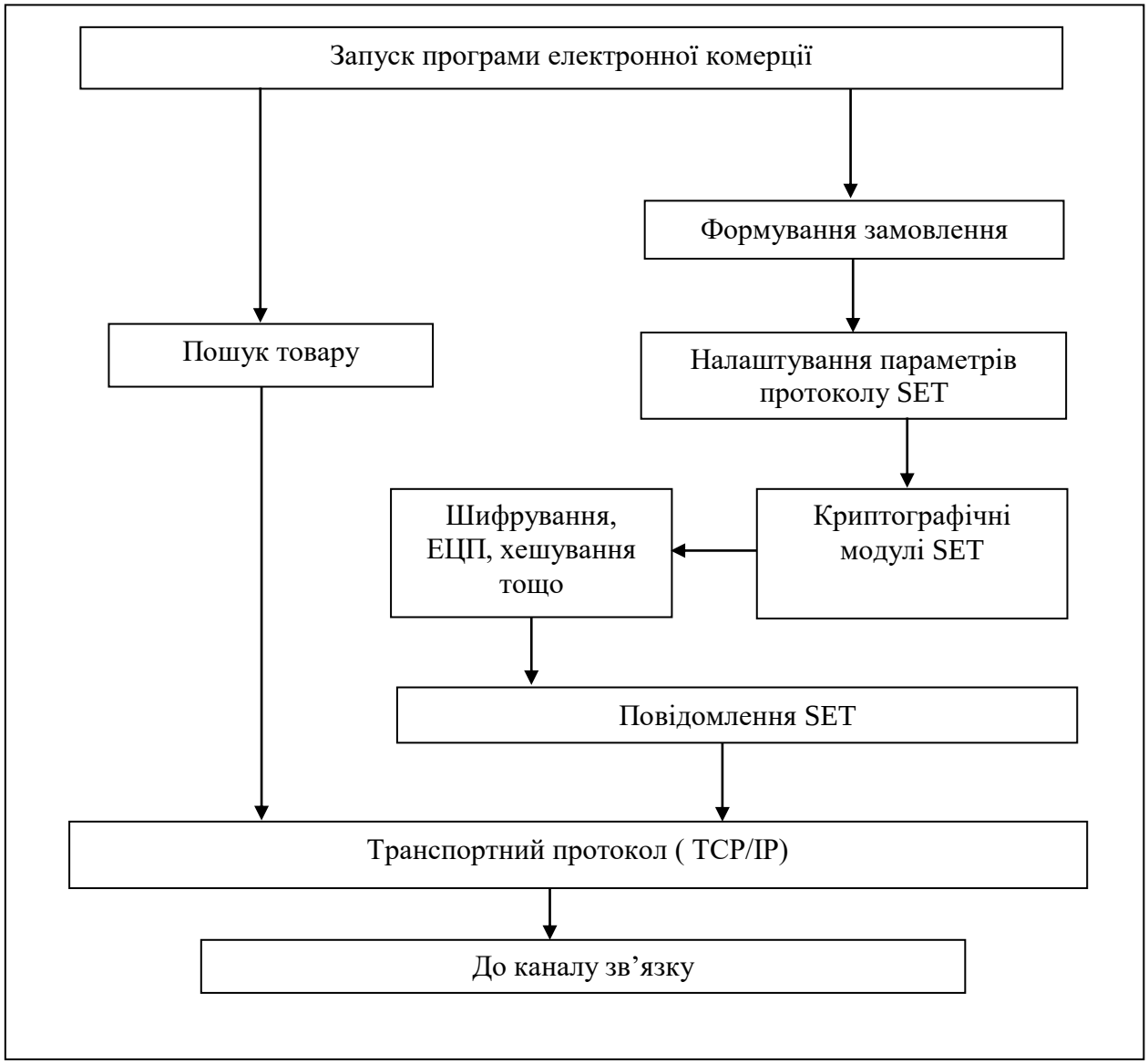
На базовому рівні SET забезпечує такі функції:

a. Аутентифікація. Усі учасники транзакції ідентифікуються за допомогою електронних підписів. Це стосується покупця, продавця та банків, емітента та еквайєра.
b. Конфіденційність. Усі операції виконуються у зашифрованому вигляді.
c. Цілісність. Забезпечується електронним цифровим підписом учасників транзакції.
d. Подвійний підпис. SET реалізує принцип подвійного підпису, коли до базового повідомлення приєднується додаткове повідомлення, призначене для одного з партнерів.
e. Незалежність від транспортного протоколу. Служби захисту протоколу SET встановлюються на вищих рівнях моделі OSI, тобто не залежать від транспортних протоколів, які працюють в Інтернет.

Протокол працює з чотирма суб'єктами: власником картки, банком-емітентом, продавцем, банком-еквайєром.

Окрім цих сутностей в системі присутні центри сертифікації, завдання яких полягає у підтвердженні істинності параметрів аутентифікації сторін. З цими центрами взаємодіють усі учасники транзакції.

Сеанс роботи з участю протоколу SET виглядає наступним чином.



Як видно з рисунку, протокол SET підключається лише на стадії формування замовлень. Пошук та вибір товарів відбувається без його участі.

Таблиця 1. Криптографічні алгоритми SET.

№	Назва	Функція	Призначення
1.	DES	Симетричне шифрування	Захист конфіденційності
2.	RSA	Асиметричне шифрування	Забезпечення аутентифікації, ЕЦП
3.	SHA	Хешування	Забезпечення цілісності
4.	HMAC-SHA	Хешування	Автентичність повідомлень

Обробка повідомлень в протоколі SET виконується у такий спосіб:

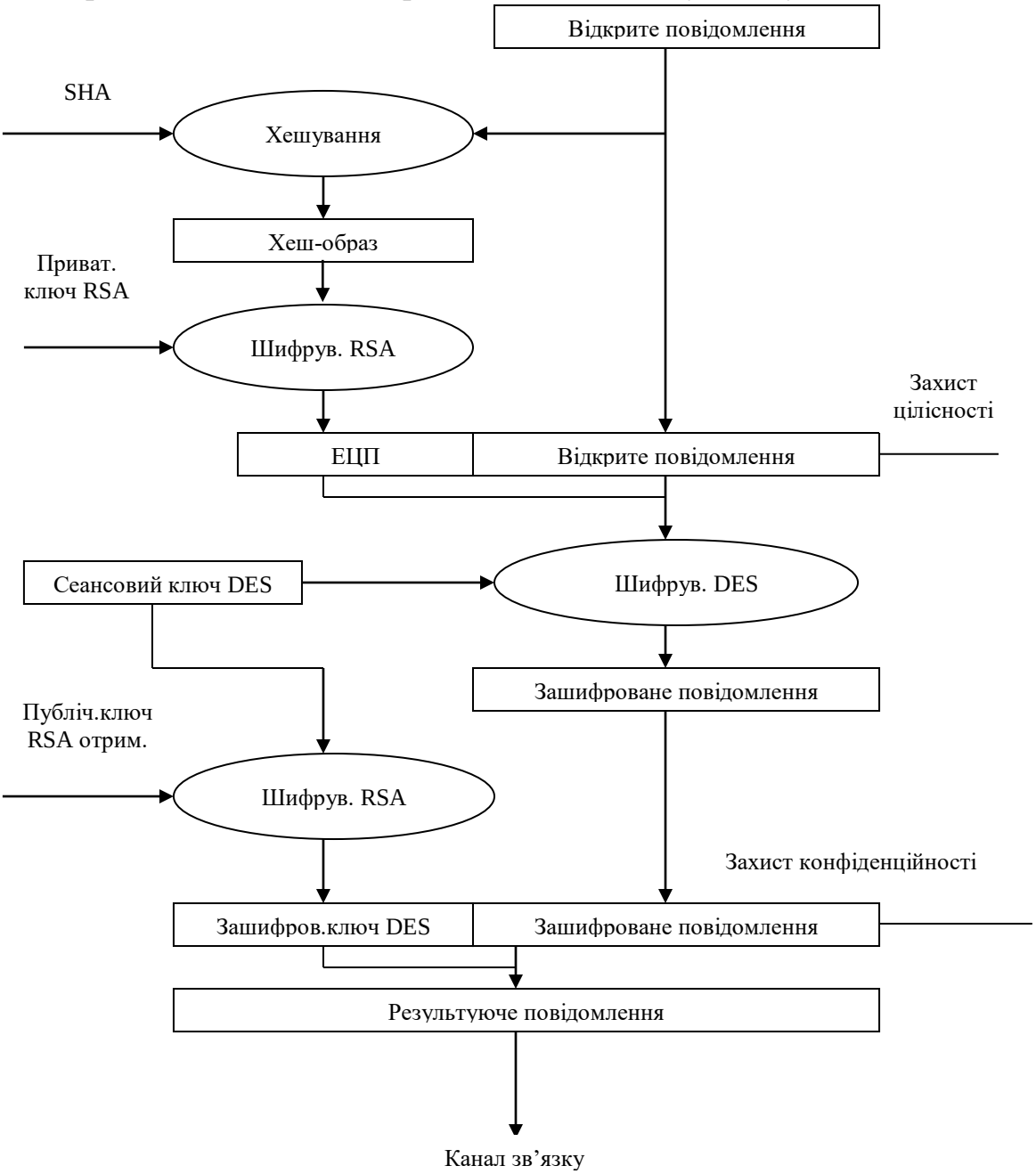


Рис.1. Схема захисту повідомлень в протоколі SET.

Для захисту ключа DES часто використовують технологію OAEP (Optimal Asymmetric Encryption Padding) – тобто технологію оптимального доповнення випадковою величиною. Ця технологія доповнює ключ (64 біти) до 128 біт спеціально згенерованою випадковою величиною, після чого симетричний ключ шифрування шифрується асиметричним алгоритмом і передається каналом зв'язку разом із зашифрованим повідомленням. Це значно підсилює захист ключа від атаки повного перебору.

Технологія детальніше виглядає таким чином.

Нехай $G(X)$ та $H(X)$ – хеш-функції, I – випадкове доповнення. Нехай блок тексту, який вимагає доповнення, має довжину k (в нашому випадку – 64 біти). Доповнюємо повідомлення нулями до розмірності хеш-образу (наприклад, до

128 біт), тобто за блоком тексту буде стояти *n-k* нулів (у нашому випадку – 8 нулів).

Формуємо таке повідомлення:

$$M_i = \{m_i \| 0^{n-k} \oplus G(I)\} \| \{I \oplus H(m_i \| 0^{n-k} \oplus G(I))\}$$

Шифруємо це повідомлення асиметричним алгоритмом, наприклад, RSA на публічному ключі отримувача:

$$C_i = M_i^e \bmod N$$

і відправляємо разом з шифрованим повідомленням в канал зв'язку.

Розшифровка та виділення повідомлення m_i.

1. Розшифровуємо повідомлення:

$$M_i = C_i^d \bmod N$$

2. Виділяємо ліву та праву частини:

Ліва частина = $\{m_i \| 0^{n-k} \oplus G(I)\}$;

Права частина = $\{I \oplus H(m_i \| 0^{n-k} \oplus G(I))\}$

1. Маючи ліву частину, знаходимо $H(m_i \| 0^{n-k} \oplus G(I))$
2. Знаючи $H(m_i \| 0^{n-k} \oplus G(I))$, знаходимо I, додаючи H() до правої частини за модулем 2.
3. Знаючи I, знаходимо $G(I)$ та m_i.

Практична частина

Метою цієї лабораторної роботи є реалізація схеми захисту інформації, поданої на рис. 1.

Для його реалізації використайте бібліотеки Microsoft CryptoAPI або іншої криптографічної бібліотеки з аналогічним функціоналом.

Будь-якою мовою програмування створіть програмне забезпечення, яке повинно мати наступний функціонал:

1. Передавати та приймати захищені повідомлення мережею.
2. Генерувати пари ключів RSA для приймальної та передавальної сторін та розповсюджувати ці ключі **без сертифікатів** (наприклад, записом на флеш-диск).
3. Використовувати функції Microsoft CryptoAPI (або аналогічної системи) для реалізації процедур захисту, поданих на рис.1 та в табл.1.
4. Розшифровувати отримані повідомлення, перевіряти цифровий підпис і в разі порушень цілісності – повідомляти користувача.
5. **За бажанням** реалізувати технологію OAEP для ключів DES. За правильну реалізацію технології нараховуються додаткові бали.

Підготуйте звіт з лабораторної роботи, який повинен містити:

1. Код програми.
2. Склад інформації на флешці (ключова інформація).
3. Протокол виконання роботи.
4. Відповіді на контрольні запитання.
5. Висновки з лабораторної роботи.

Список використаної літератури

1. Закон України “Про захист інформації в інформаційно-телекомунікаційних системах” (1994);
2. Закон України “Про захист персональних даних” (2010)
3. СТРАТЕГІЯ національної безпеки України (затверджена Указом Президента України від 26 травня 2015 року № 287/2015)
4. Закон України “Про національну безпеку (2018)
5. Стратегія кібербезпеки України” (Введено в дію Указом Президента України від 15 березня 2016 року №96/2016)
6. Положення про технічний захист інформації в Україні, затверджене Указом Президента України від 27.09.99 № 1229;
7. ДСТУ 3396 0-96 Захист інформації. Технічний захист інформації. Основні положення;
8. ДСТУ 3396 1-96 Захист інформації. Технічний захист інформації. Порядок проведення робіт;
9. НД ТЗІ 2.1-001-2001 Створення комплексів технічного захисту інформації. Атестація комплексів. Основні положення.
10. НД ТЗІ 1.1-003-99: Термінологія в області захисту інформації в комп’ютерних системах від несанкціонованого доступу. НД ТЗІ 2.5-004-99: Критерії оцінки захищеності інформації у комп’ютерних системах від несанкціонованого доступу. Затверджено наказом ДСТСЗІ СБУ № 22 від 28.04.1999. ДСТСЗІ СБУ, К: 1999. – 34с.
11. НД ТЗІ 1.1-002-99. Загальні положення щодо захисту інформації в комп'ютерних системах від несанкціонованого доступу.
12. НД ТЗІ 1.1-003-99. Термінологія в області захисту інформації в комп'ютерних системах від несанкціонованого доступу.
13. НД ТЗІ 1.1-005-07 Захист інформації на об’єктах інформаційної діяльності. Створення комплексу технічного захисту інформації. Основні положення.

14. НД ТЗІ 1.4-001-00. Типове положення про службу захисту інформації в автоматизованій системі.
15. Ахрамович В.М. Інформаційна безпека: навч. посіб. К.:ДП «Інформ.-аналіт. Агенство», 2009.-276с.
16. Головань СМ., Васюков І.В., Давиденко А.М., Хорошко В.О., Щербак Л.М. Основи організації електронного документообігу: У 2 т./ – К.: ДУІКТ, 2008. – Т. 1. – 230 с., Т. 2. – 233 с.
17. Голубенко О.Л., Хорошко В.О., Петров О.С., Головань С.М. Конфіденційне діловодство. Практикум: Навч. Посіб. – Луганськ: СНУ ім. В.Даля, 2010. – 180 с.
18. Домарєв В.В., Скворцов С.О. Організація захисту інформації на об’єктах державної та підприємницької діяльності. Навчальний посібник. – К.: Вид-во Європ. Ун-ту, 2006. – 102 с.
19. Богуш В.М., Юдін О.К. Інформаційна безпека держави. Навчальний посібник. –К.: "МК-Прес", 2005. – 432 с.
20. Бондаренко М. Ф. Визначення та обґрунтування суті політики інформаційної безпеки / М. Ф. Бондаренко, О. В. Потій, Ю. І. Горбенко та ін.// Радиотехника. – 2003. – № 134. – С. 9-25.
21. Домарєв В. В. Обґрунтування основних функцій системи управління інформаційною безпекою / В. В. Домарєв, Д. В. Домарєв, С. Б Гордієнко. // Вісник Державного університету інформаційно-комунікаційних технологій. – 2012. – Т. 10, № 2. – С. 102-104.
22. Домарєв В.В., Швець В.А., Шестакова В.В. Організаційне забезпечення захисту інформації з обмеженим доступом. Навчальний посібник. – К.: НАУ, 2006. – 108 с.
23. Зубок М. І. Безпека банківської діяльності: навч. посібник / Зубок . І. — К. : КНЕУ, 2002. — 190 с.
24. Кобозева А.А., Мачалін І.О., Хорошко В.О. Аналіз захищеності інформаційних систем. Підручник.-К. ДУІКТ, 2010. - 316 с.

25. Лужецький В.А. Захист персональних даних. Навчальний посібник./ Лужецький В.А., Войтович О.П., Дудатьєв А.В – Вінниця: ВНТУ, 2009. – 487 с.
26. Лужецький В.А., Войтович О.П., Дудатьєв А.В. Інформаційна безпека. Навчальний посібник. – Вінниця: УНІВАР-СУМ-Вінниця, 2009. – 240 с.
27. Самохвалов Ю.Я., Темніков В.О., Хорошко В.О. Організаційно-технічне забезпечення захисту інформації / За ред. проф. В.О.Хорошка – К.: Видавництво НАУ, 2002. – 208с.
28. Хорошко В.О, Чередниченко В.С., Шелест М.Є. Основи інформаційної безпеки : К.: ДУІКТ, 2008. – 186 с.
29. Юдін О. К. Захист інформації в мережах передачі даних: підруч. / Г. Ф. Конахович, О. Г. Корченко, О. К. Юдін. — К.: Вид-во ТОВ НВП «ШТЕРСЕРВІС», 2009. — 714 с.
30. Юдін О. К. Інформаційна безпека. Нормативно-правове забезпечення: підруч. / О. К. Юдін. — К. : НАУ, 2011. — 640 с.
31. Юдін О.К., Корченко О.Г., Конахович Г.Ф. Захист інформації в мережах передачі даних. – К.: Вид-во ТОВ —НВПІНТЕРСЕРВІС, 2009. – 716 с.

Інформаційні ресурси

32. GnuPG для Linux.– URL: https://gnupg.org/download/index.html
33. Kleopatra для Linux .– URL: https://kde.org/applications/en/utilities/org.kde.kleopatra
34. Kleopatra для Windows.– URL: https://www.gpg4win.org/download.html
35. Довідкова система для „VeraCrypt”.– URL: https://www.veracrypt.fr/en/Home.html