ЛАБОРАТОРНА РОБОТА № 7. ЦИФРОВИЙ ПІДПИС

Мета роботи: набути уміння із створення та перевірки підпису повідомлення за допомогою алгоритмів RSA та Ель-Гамаля; навчитись створювати власні ключі криптографічного захисту даних, обмінюватися ними з іншими користувачами, шифрувати та підписувати повідомлення за допомогою системи GNU Privacy Guard.

Матеріально-технічне забезпечення: ПК зі встановленим програмним забезпеченням GNU Privacy Guard, інструкції до лабораторної роботи, текстові повідомлення для шифрування та підписування згідно варіанту.

Теоретичні відомості

ПОНЯТТЯ ЦИФРОВОГО ПІДПИСУ

Із широким розповсюдженням у сучасному світі електронних форм документів, у тому числі і конфіденційних, та засобів їхньої обробки, особливо актуальним є питання автентифікації, ідентифікації та неспростовності електронної документації. Для захисту від підробки, перевірки цілісності даних та достовірності джерела повідомлення використовують цифровий підпис.

Електронний підпис – електронні дані, які додаються підписувачем до інших електронних даних або логічно з ними пов’язуються і використовуються ним як підпис.

(Електронний) цифровий підпис – вид електронного підпису, отриманого за результатом криптографічного перетворення набору електронних даних, який додається до цього набору або логічно з ним поєднується і дає змогу підтвердити його цілісність та ідентифікувати підписувача.

Існує декілька алгоритмів побудови цифрового підпису (ЦП). Найбільш ефективним та найпоширенішим у застосуванні на даний момент є алгоритм ЦП на основі асиметричних криптосистем з використанням хеш-функцій. *Хеш-функція* являє собою функцію, математичну або іншу, що отримує на вхід рядок змінної довжини і перетворює його в рядок фіксованої, зазвичай меншої, довжини. Такі перетворення ще називають *функціями згортки*, а їх результати – *хешем*, *хеш-значенням* або *дайджестом повідомлення*.

Хеш-функція *H*, яка використовується у алгоритмі ЦП, призначена для того, щоб стиснути повідомлення *M* довільної довжини до двійкового хеш-значення *h(M)* фіксованої довжини.

Основні властивості криптографічної хеш-функції:

1) *Детермінованість* – для однакових повідомлень *M* функція має повертати однакові хеш-значення *h*;
2) *Односторонність* – за значенням *h* неможливо відновити *M*;
3) *Наявність лавинного ефекту* – будь-які, навіть незначні, зміни у повідомленні *M* призводять до значних змін у хеш-значенні *h*;
4) *Відсутність колізій (унікальність хеша)* – ймовірність співпадіння хеш-значень двох різних повідомлень повинна бути надзвичайно малою;
5) Висока швидкість роботи.

ЕТАПИ ЦИФРОВОГО ПІДПИСУ

1. *Генерація пари ключів.* За допомогою алгоритму генерації ключів створюється пара ключів – закритий (для створення підпису) та відкритий (для перевірки підпису).
2. *Формування підпису.* Для заданого електронного документа за допомогою деякої хеш-функції обчислюється хеш-значення, після чого воно зашифровується із використанням закритого ключа підписувача. Зашифрований дайджест і є ЦП для даного документу.
3. *Перевірка (верифікація) підпису.* Для отриманого документу одержувач знову обчислює його хеш-значення, після чого за допомогою відкритого ключа підписувача дешифрує ЦП. Якщо хеші рівні – підпис справжній.

Управлянням ключами займаються центри сертифікації ключів (ЦСК), що забезпечують:

- доступ користувача до справжнього відкритого ключа іншого користувача;
- захист ключів від підміни зловмисником;
- організацію відкликання ключа у випадку його компрометації.

Сертифікат відкритого ключа – електронний документ, який засвідчує належність відкритого ключа фізичній або юридичній особі.

АЛГОРИТМ ЦИФРОВОГО ПІДПИСУ RSA

Для створення підпису повідомлення M спочатку необхідно за допомогою деякої хеш-функції обчислити хеш-значення $h(M)$.

Далі за алгоритмом RSA генеруються ключі $(e,\ n)$ і $(d,\ n)$.

ЦП повідомлення $h(M)$ буде мати вигляд: $S = h(M)^d mod\ n.$

Тепер кожний, хто має відкритий ключ підписувача повідомлення, може перевірити дійсність підпису. Для цього необхідно знайти результат хешування прийнятого повідомлення M за допомогою тієї самої хеш-функції $h'(M)$ та порівняти його із $s^e mod\ n = h(M)$. Якщо дайджести рівні – підпис дійсний.

Приклад 7.1:

З використанням алгоритму RSA підписати та перевірити підпис повідомлення M хеш-значення, якого $h(M) = 88$.

Оберемо $p = 17$ і $q = 11$, тоді $n = p \cdot q = 17 \cdot 11 = 187$.

Обчислимо $\varphi(187) = 16 \cdot 10 = 160$.

Виберемо відкритий ключ $e = 7$ та перевіримо виконання умов: $1 < 7 < 160$, НСД(7, 160) =1.

Знайдемо закритий ключ $d = 23$ за розширеним алгоритмом Евкліда з рівняння $7d \equiv 1(mod\ 160)$.

Обчислимо підпис за допомогою закритого ключа підписувача:

$$s = h(M)^d mod\ n = 88^{23} mod\ 187 = 11.$$

Для перевірки підпису повідомлення M одержувачу потрібно знову обчислити його хеш-значення $h(M) = 88$ та порівняти із значенням, отриманим за допомогою відкритого ключа підписувача:

$$s^e mod\ n = 11^7 mod\ 187 = 88.$$

В даному випадку будемо вважати, що підпис справжній.

АЛГОРИТМ ЦИФРОВОГО ПІДПИСУ ЕЛЬ-ГАМАЛЯ

Як правило, спочатку потрібно за допомогою деякої хеш-функції знайти дайджест $h(M)$ для повідомлення M.

Для генерації пари ключів спочатку вибирається просте число p та числа g (первісний корінь за модулем p) й x (закритий ключ). Обидва ці числа повинні бути менше p. Після чого обчислюється $y = g^x\ mod\ p$ (відкрити ключ).

Виберемо сесійний ключ – випадкове число k, таке що $1 < k < p - 1$ та обчислимо $r = g^k mod\, p$. Після чого обчислимо $s = k^{-1}(h(M) - xr) mod\, p - 1$.

Отже, підписом повідомлення M являється пара $(r,\ s)$.

Випадкове значення k повинне зберігатися в секреті і не повинно дублюватися для різних підписів. Для перевірки підпису потрібно використати відкриті параметри $(p,\ g,\ y)$ та переконатися, що $g^{h(M)} \equiv y^r r^s (mod\, p)$.

Приклад 7.2:

З використанням алгоритму Ель-Гамаля підписати та перевірити підпис повідомлення M хеш-значення, якого $h(M) = 14$.

Виберемо $p = 19$ і $g = 10$. Нехай $x = 16$ – закритий ключ. Обчислимо відповідний відкритий ключ $y = g^x mod\, p = 10^{16} mod\, 19 = 4$.

Виберемо $k = 5$ (сесійний ключ), такий що $1 < 5 < 18$.

Визначимо, що $d = 23$ (закритий ключ) за розширеним алгоритмом Евкліда з рівняння $7d \equiv 1(mod\, 160)$.

Обчислимо підпис:

$$r = 10^5 mod\, 19 = 3;$$

$$s = 5^{-1}(14 - 16 \cdot 3) mod\, 18 == -374\, mod\, 18 = 4;$$

$5 \cdot ? \equiv 1\, mod\, 18 \rightarrow 5^{-1} mod\, 18 = 11$ *(за розширеним алгоритмом Евкліда)*.

Приймається $(M, 3,\ 4)$. Обчислимо ліву та праву частину рівняння $g^{h(M)} \equiv y^r r^s (mod\, p)$ за модулем p:

$$g^{h(M)} mod\, p = 10^{14} mod\, 19 = 16;$$

$$y^r r^s\, mod\, p\ = 4^3 \cdot 3^4 mod\, 19 = 16.$$

Можна зробити висновок, що підпис дійсний.

РОБОТА ІЗ СИСТЕМОЮ GNU PRIVACY GUARD ІЗ ВИКОРИСТАННЯМ ОБОЛОНКИ KLEOPATRA

GNU Privacy Guard, **GnuPG** – вільно поширюване програмне забезпечення, що використовує криптографію з відкритим ключем. Перша версія проекту, створена Вернером Кохом (Werner Koch) та профінансована німецьким урядом, вийшла в світ у 1999 році під ліцензією GNU General Public. Функції GnuPG дозволяють шифрувати та підписувати повідомлення за допомогою цифрового підпису, а також керувати списками відкритих ключів респондентів.

Звичним інтерфейсом для GnuPG є командний рядок, проте на сьогоднішній день існують різні зовнішні оболонки, які роблять доступною функціональність цієї програми через графічний інтерфейс користувача, наприклад *Kleopatra* для Windows або *GNU Privacy Assistant* (GPA) для Linux.

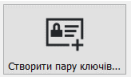
В GnuPG використовуються різні криптографічні алгоритми: симетричні шифри, шифрування з відкритим ключем і змішані (гібридні) алгоритми.

Гібридна (змішана, комбінована) криптосистема – це криптосистема, в якій розподіл ключів здійснюється за допомогою асиметричних криптоалгоритмів, а процес шифрування даних – за допомогою симетричних. Тобто симетричний ключ використовується для шифрування даних, а асиметричний для шифрування самого симетричного ключа. Гібридні криптосистеми поєднують в собі зручність розподілу секретних ключів та високу швидкість шифрування.

Як правило, при гібридному шифруванні створюється *одноразовий секретний сеансовий ключ* – це псевдовипадкове число, яке генерується на основі випадкових рухів миші, натискань клавіш клавіатури тощо. Такий ключ використовується лише один раз для шифрування повідомлення з використанням деякого надійного та швидкого симетричного алгоритму. Сеансовий ключ зашифровується відкритим ключем одержувача та додається до шифротексту. Під час дешифрування процедури виконуються у зворотному порядку.

Створення пари ключів

При першому запуску *Kleopatra* (рис. 7.1) потрібно створити власну зв’язку ключів. Для цього необхідно виконати наступні дії:

1) натиснути кнопку 

або скористатися меню *Файл⇒Створити пару ключів*;

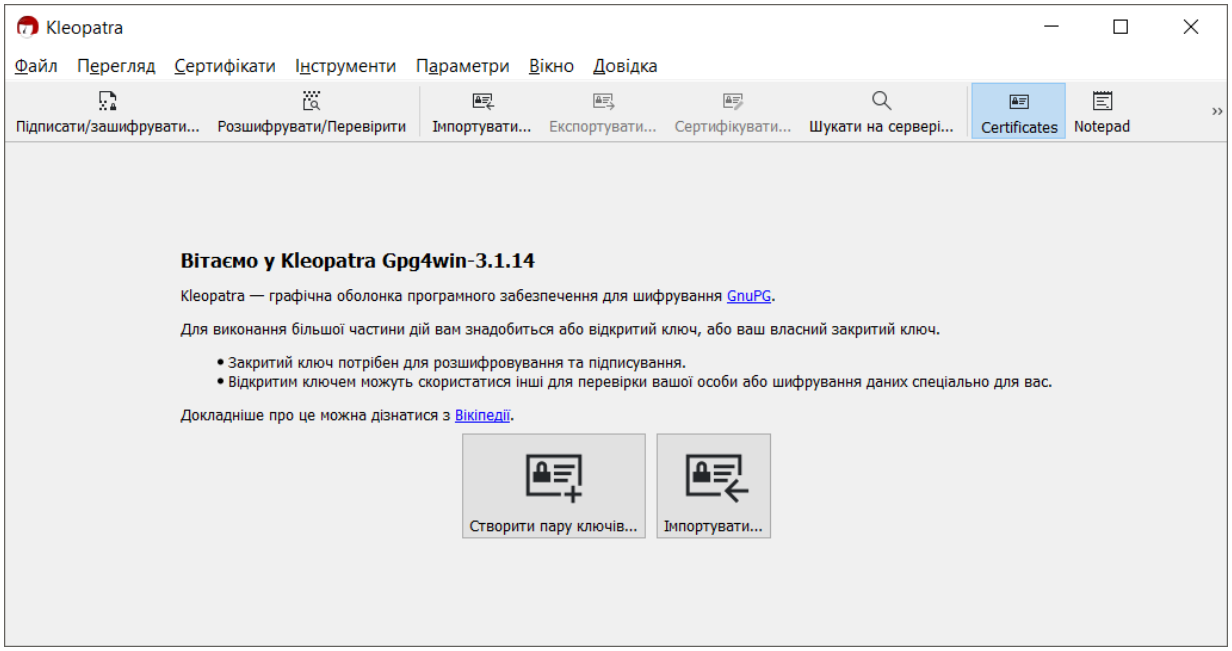


Рис. 7.1. Стартове вікно оболонки Kleopatra

2) у вікні *Майстра створення ключів* (рис. 7.2) потрібно ввести відомості про себе у відповідні поля (ім'я, електронну адресу); кнопка Додаткові параметри... дозволяє вибрати тип ключа його довжину, строк дії тощо.

Основною особливістю GnuPG є система ключів. В GnuPG користувач створює декілька ключів, причому кожен служить для окремої дії (і використовує різні алгоритми). Один із ключів, що створюється першим, є *головним ключем*, решта ключів йому підпорядковані – це *підключі* (субключі).

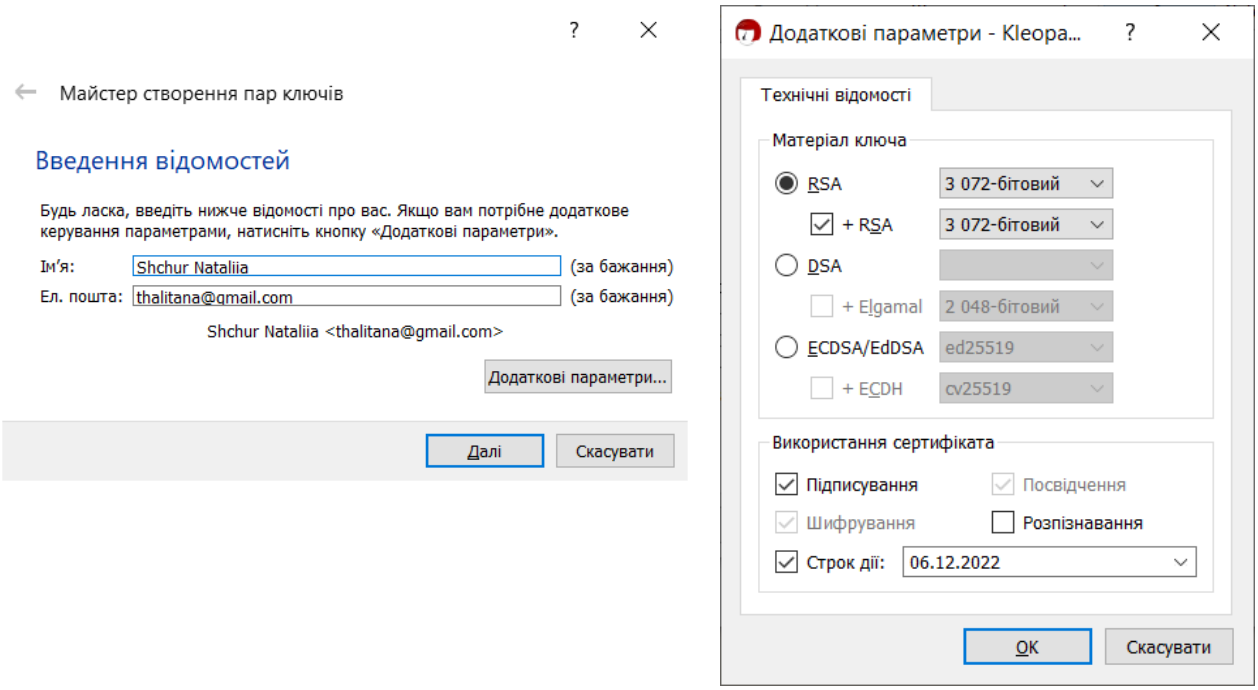


Рис. 7.2. Створення пари ключів за допомогою майстра

3) у наступному вікні необхідно натиснути *Створити* та ввести пароль для захисту нового ключа (рис. 7.3);

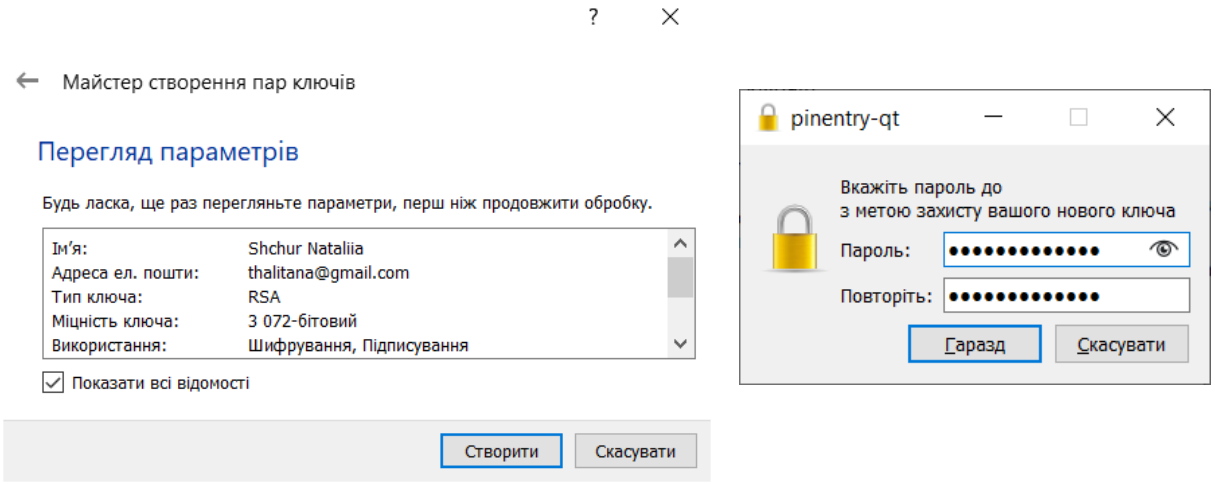


Рис. 7.3. Введення паролю для захисту нового ключа

4) у наступному вікні майстер має повідомити про успішне створення ключів (рис. 7.4), після чого потрібно натиснути кнопку *Завершити*.

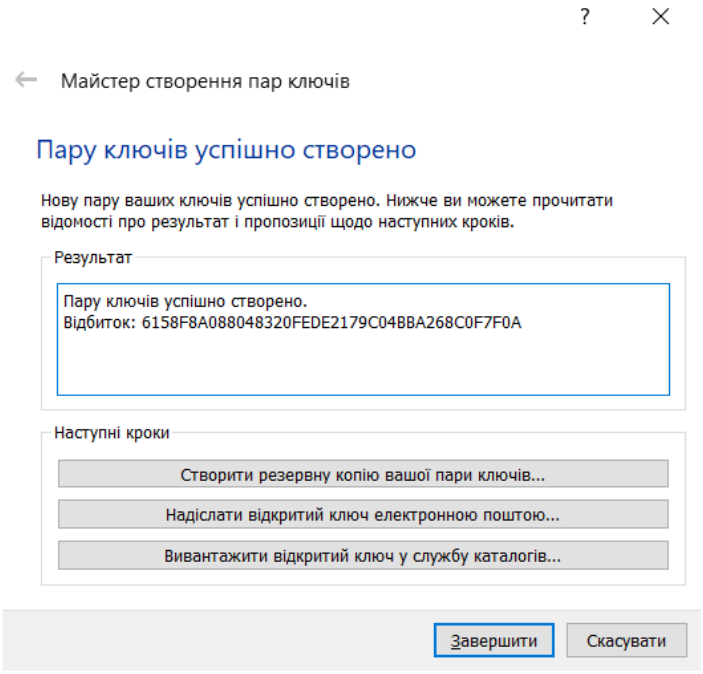


Рис. 7.4. Повідомлення про успішне створення ключів

Усі функції управління ключами здійснюються у вікні ***Kleopatra*** (рис. 7.5), в якому висвітлюються всі ключі, створені користувачем для власного користування, а також усі імпортовані публічні ключі його кореспондентів.

Ключі зберігаються у зашифрованій формі у вигляді двох файлів, які називаються *зв'язками ключів* (keyrings). Ці файли записуються у папках на диску відповідно до поточних налаштувань.

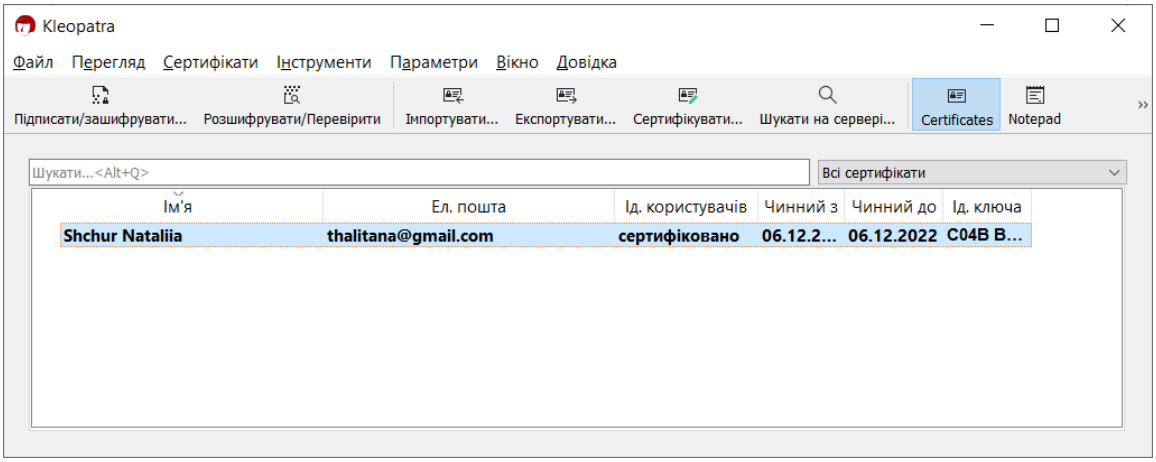
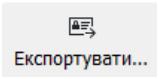


Рис. 7.5. Список наявних ключів у вікні оболонки Kleopatra

Експорт ключів

До початку обміну повідомленнями з іншими користувачами GPG варто обмінятись з ними публічними ключами.

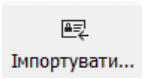
Для експорту ключа потрібно:

1) у вікні ***Kleopatra*** натиснути кнопку 

або у контекстному меню електронного ключа вибрати пункт *Експортувати*, або використати меню *Файл⇒ Експортувати*;
2) обрати папку для збереження ключа, ввести його ім'я та натиснути *Зберегти*.

Імпорт ключів

Імпортувати відкриті ключі інших користувачів можна, виконавши такі дії:

3) у вікні ***Kleopatra*** натиснути кнопку 

або у контекстному меню електронного ключа вибрати пункт *Імпортувати*, або використати меню *Файл⇒ Імпортувати*;
4) обрати ключ на диску, ввести його ім'я та натиснути *Відкрити*;
5) також варто погодитися із перевіркою сертифіката ключа (рис. 7.5).

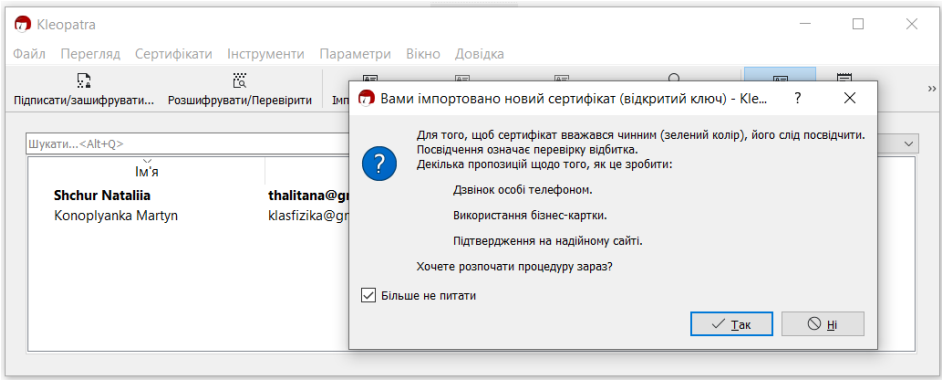
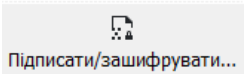


Рис. 7.5. Перевірка сертифіката ключа, що імпортується

Шифрування та (або) підписування файлів

Для шифрування та (або) підписування файлу необхідно натиснути кнопку 

або використати меню *Файл⇒ Підписати/зашифрувати*;

Відкриється діалогове вікно *Підписати/зашифрувати файли* (рис. 7.6), у якому потрібно обрати необхідну дію та обрати відкриті ключі одержувача(-ів) повідомлення, натиснувши по піктограмі .

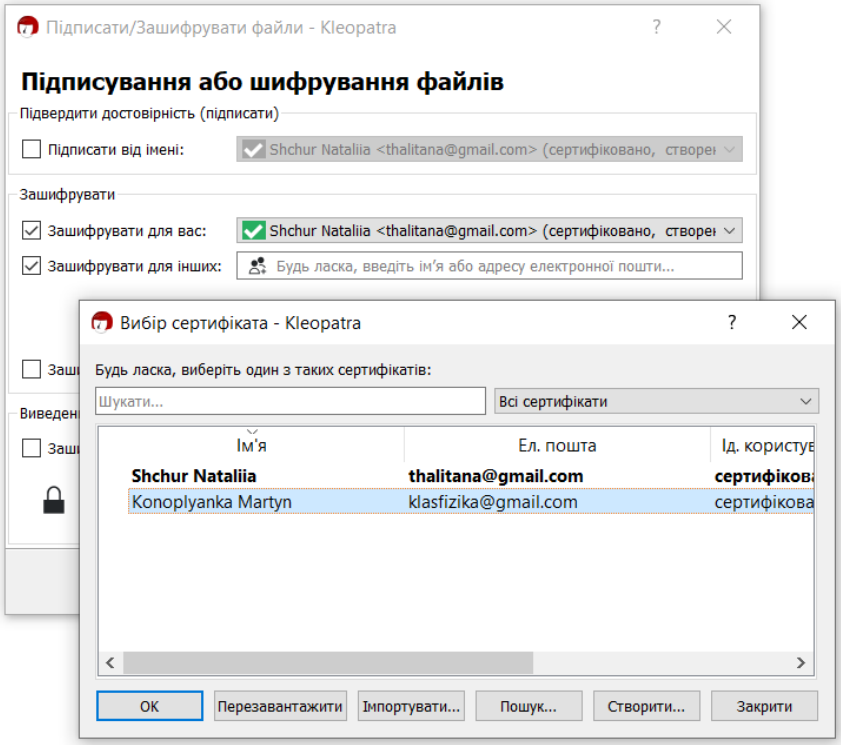
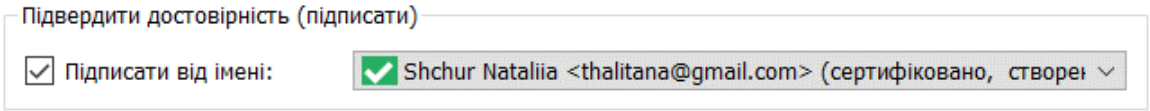


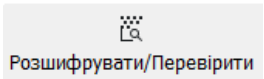
Рис. 7.6. Діалогове вікно Підписати/зашифрувати файли

Підписати файл за допомогою свого відкритого ключа дозволяє опція:



Також існує можливість виконати дві описані вище операції одночасно.

Розшифрування та (або) перевірка підпису файлів

Для розшифрування/перевірки цифрових підписів файлів використовуються кнопка 

та пункт меню *Файл⇒ Розшифрувати/Перевірити.*

Під час розшифрування на екрані з'явиться вікно перевірки пароля. Файл буде розшифрований після введення правильного пароля за умови, що його було зашифровано з використанням відкритого ключа отримувача (рис. 7.7). Очевидно також, що розшифрування файлу можливе тільки за умов наявності у середовищі

вікна ***Kleopatra*** закритого ключа отримувача. Розшифрованому файлу автоматично присвоюється назва файлу-оригіналу (файлу, який було зашифровано).

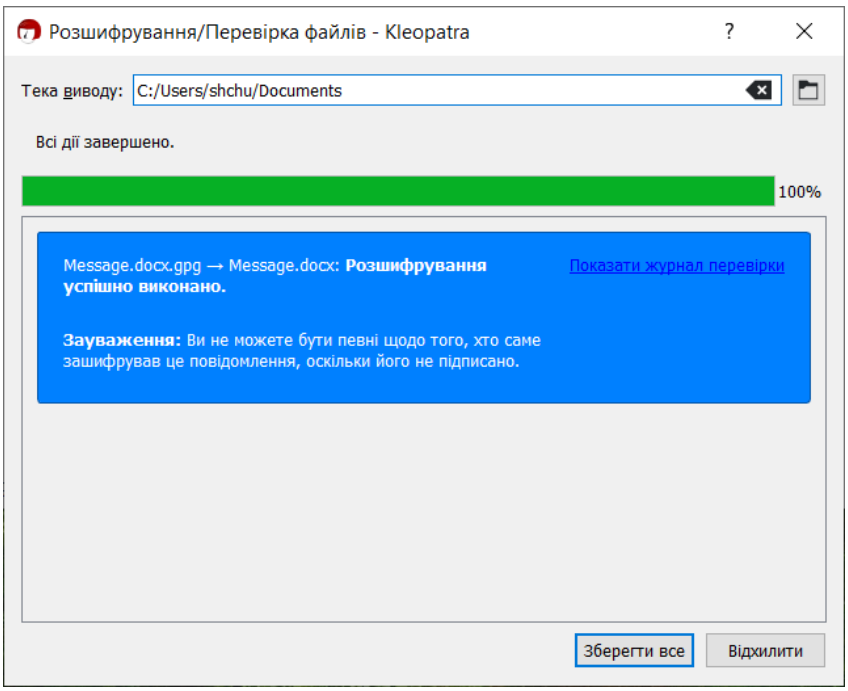


Рис. 7.7. Вікно розшифрування файлу

Якщо файл має підпис, на екрані з'являється вікно з повідомленням, яке містить назву файлу, відомості про особу, яка підписала файл, дату і час накладання підпису та позначку, чи залишається підпис дійсним (рис. 7.8.).

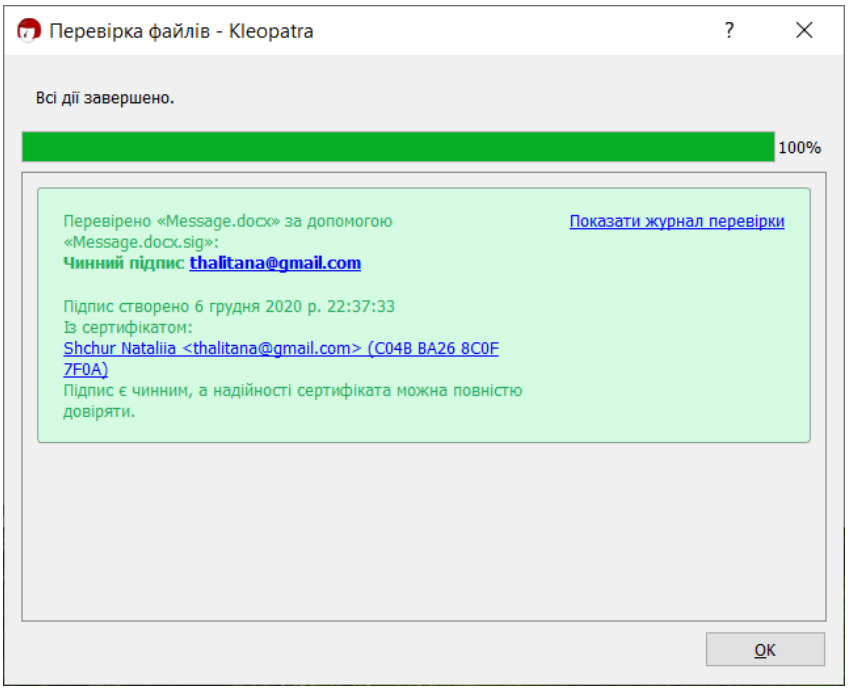


Рис. 7.8. Вікно перевірки підпису

Доступ до функцій GPG

Для забезпечення зручного виконання операцій шифрування, підписування, дешифрування, перевірки підпису тощо, у контекстному меню файлу (рис. 7.8) можна обрати *Sign and encrypt* або *More GpgEX option*.

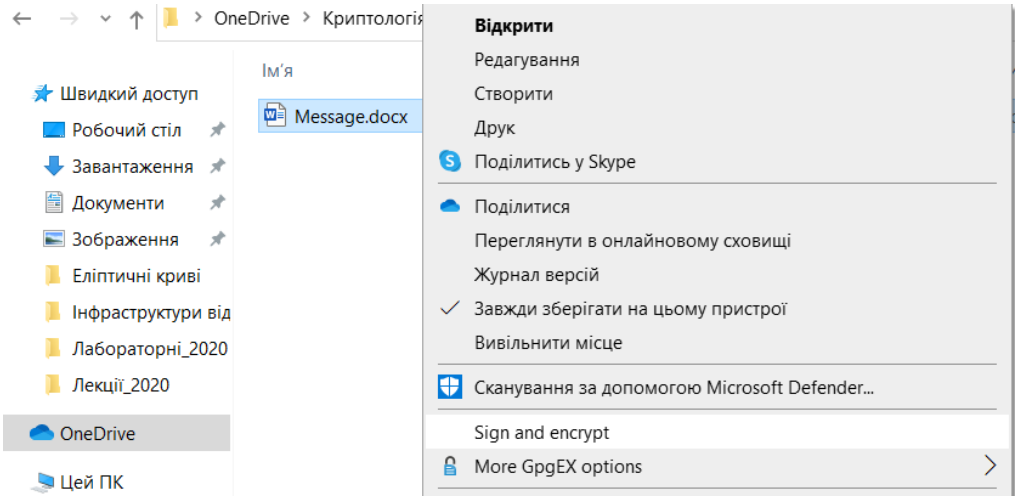


Рис. 7.9. Вибір у контекстному меню документа команд GPG

Завдання до лабораторної роботи

Завдання 1

Виконати створення та перевірку ЦП повідомлення згідно варіанту (визначається номером студента у журналі: непарний – 1 варіант, парний – 2 варіант). Усі кроки алгоритму описати у звіті.

1. З використанням алгоритму RSA створіть та перевірте підпис повідомлення, якщо його хеш $h(M) = 7$, а параметри p=13 та q=17. Самостійно оберіть відкритий ключ e та обчисліть закритий ключ d.

   За алгоритмом Ель-Гамаля виконайте формування та перевірку підпису повідомлення, якщо його хеш $h(M) = 8$, а параметри p=23 та g=5. Оберіть закритий ключ x, сесійний ключ k та обчисліть відкритий ключ y.
2. З використанням алгоритму RSA створіть та перевірте підпис повідомлення, якщо його хеш $h(M) = 6$, а параметри p=11 та q=13. Самостійно оберіть відкритий ключ e та обчисліть закритий ключ d.

   За алгоритмом Ель-Гамаля виконайте формування та перевірку підпису повідомлення, якщо його хеш $h(M) = 7$, а параметри p=19 та g=3. Оберіть закритий ключ x, сесійний ключ k та обчисліть відкритий ключ y.

Завдання 2

Виконати завдання у системі GPG та додати до звіту скріншоти вікна GPG на кожному кроці: створення ключів, експортування/імпортування ключів, шифрування/підписування, дешифрування/перевірки підпису, а також скріншот дешифрованого текстового повідомлення від викладача.

2.1. Створити ключі у діалоговому вікні ***Kleopatra*** на основі алгоритму RSA, довжиною 3072 біт. Заповнити поля *Ім'я* та *Елек. пошта* (латинськими літерами).

2.2. Експортувати свій публічний ключ у свою робочу папку. Відповідний файл повинен мати назву за шаблоном, наприклад *Shchur Nataliia_0x8C0F7F0A_public.asc*.

2.3. Відправити свій публічний ключ на пошту викладача.

2.4. Імпортувати публічний ключ викладача до середовища ***Kleopatra***.

2.5. За допомогою текстового редактора створити файл, вказати у ньому своє прізвище, ім'я, по батькові. Присвоїти файлу назву *Enc_N.docx*, де *N* – номер студента за списком групи, впорядкованим за алфавітом (наприклад, *Enc _12.docx*).

2.6. Із використанням відкритого ключа викладача зашифрувати *Enc_N.docx* за допомогою GPG. Схема зашифрування повідомлення із використанням GnuPG представлена на рис. 7.10.

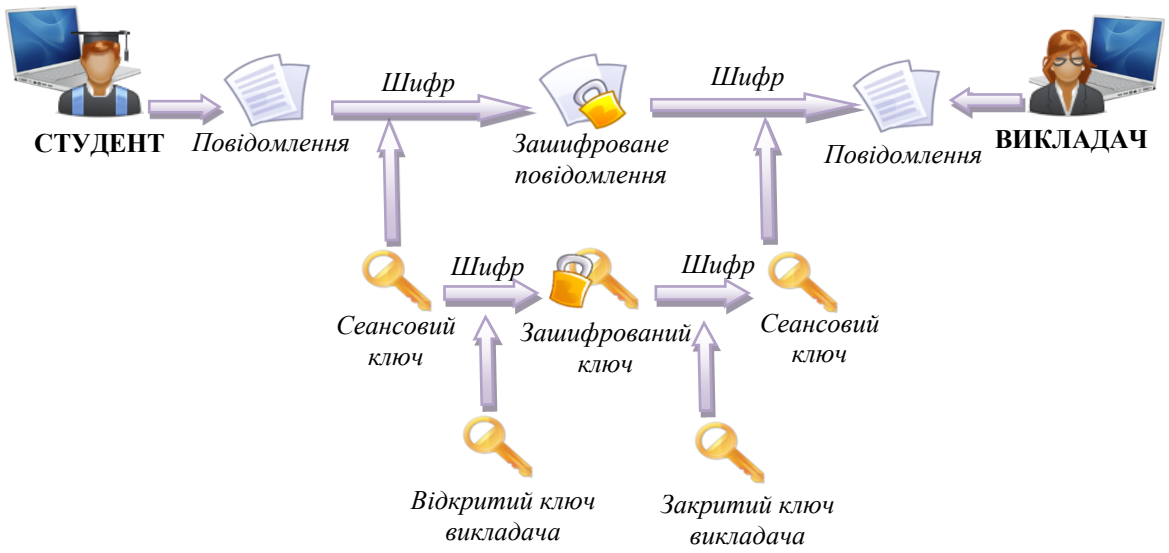


Рис. 7.10. Схема зашифрування повідомлення із використанням GnuPG

2.7. За допомогою текстового редактора створити файл, вказати у ньому свій варіант, курс, групу. Присвоїти файлу назву *Enc_Sign_N.docx*, де *N* – номер

студента за списком групи, впорядкованим за алфавітом (наприклад, *Enc_Sign _12.docx*).

2.8. Із використанням свого ключа підписати *Enc_Sign_N.docx* та зашифрувати за допомогою ключа викладача. Схема алгоритму створення та перевірки підпису з використанням GnuPG представлена на рис. 7.11.

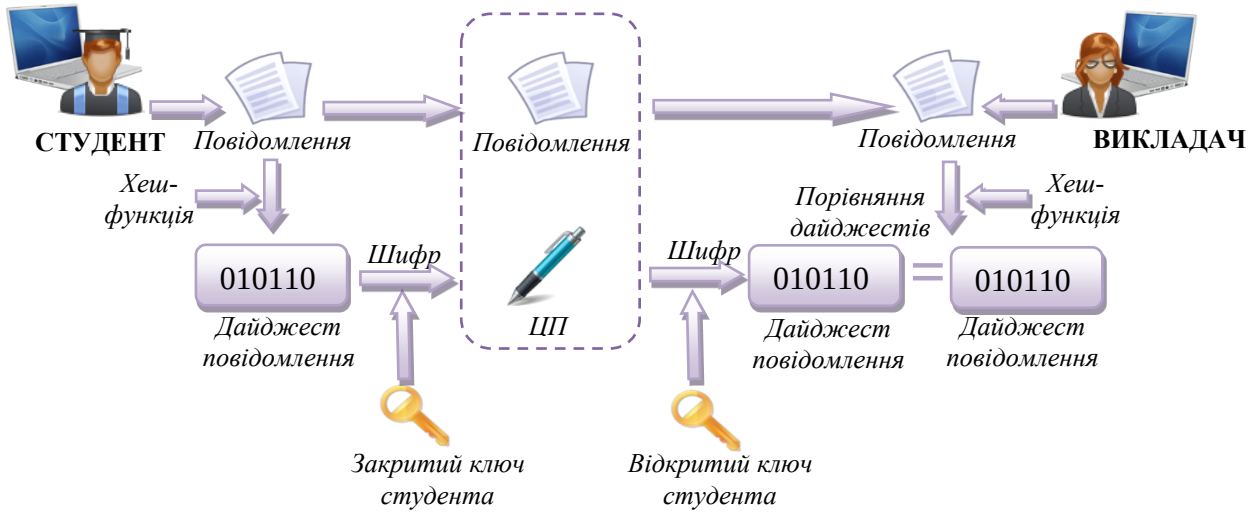


Рис. 7.11. Схема створення та перевірки підпису з використанням GnuPG

2.9. Відправити два файли викладачеві: зашифрований *Enc_N.docx* та підписаний/зашифрований *Enc_Sign_N.docx*.

2.10. Отримати від викладача зашифроване повідомлення, підписане його цифровим підписом.

2.11. Розшифрувати повідомлення викладача та перевірити дійсність його підпису у системі GPG.

Контрольні питання:

1. Для чого потрібен цифровий підпис?
2. Дайте визначення поняттям «хешування», «хеш-функція».
3. Опишіть схему створення і перевірки ЦП.
4. Який порядок використання відкритого та закритого ключів при створенні і перевірці ЦП?
5. Які схеми цифрового підпису існують?
6. Як здійснюється підпис RSA? Яка відмінність підпису RSA від шифру RSA?
7. Як здійснюється підпис Ель-Гамаля?
8. Як здійснюється перевірка на дійсність підпису Ель-Гамаля?