УДК 004.5

Сніжко Тетяна, ст. магістратури факультету кібернетики; науковий керівник – ст. викл. Суховецький І. О. (Міжнародний економіко-гуманітарний університет імені академіка Степана Дем'янчука, м. Рівне)

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ КРИПТОГРАФІЧНОГО ЗАХИСТУ ІНФОРМАЦІЇ

Анотація. *У статті досліджено процес розробки програмного забезпечення для криптографічного захисту інформації. Проаналізовано такі рівні його захисту: фізичний захист (центрального процесора, дисків і терміналів), захист файлів, захист процесів і працюючої системи. Обгрунтовано, чому кожен користувач має право засекречувати і захищати авторське середовище і власні файли.*
Ключові слова: *криптографія, захист інформації, симетричні та асиметричні алгоритми.*

Аннотация. *В статье исследован процесс разработки программного обеспечения для криптографической защиты информации. Проанализированы следующие уровни его защиты: физическая защита (центрального процессора, дисков и терминалов), защиту файлов, защиту процессов и работающей системы. Обосновано, почему каждый пользователь имеет право засекречивать и защищать авторские среду и собственные файлы.*
Ключевые слова:. *криптография, защита информации, симметричные и асимметричные алгоритмы.*

Annotation. *In the article the process of developing software for cryptographic information protection is investigated. It is analyzed the following levels of protection: physical protection (CPU, disk and terminals), file protection, protect working processes and systems. It is substantiated why every user has the right to classify and protect copyright environment and own files.*
Keywords: *cryptography, information security, symmetric and asymmetric encryption algorithms.*

У відповідності зі стандартом ISO 7498-2, під цілісністю інформації необхідно розуміти умови, при яких дані (повідомлення, команди, програмне забезпечення) зберігаються для використання за призначенням. «Автентифікація» в перекладі з латинської означає «встановлення справжності». Тому під автентифікацією розуміється встановлення справжності інформації (повідомлень, команд, даних, програмного забезпечення), джерела та приймача даних винятково на основі внутрішньої структури самої інформації.

Кінцевою метою автентифікації інформації є забезпечення захисту учасників інформаційного обміну від навмисного або випадкового нав'язування неправдивої інформації. Автентифікація як процедура встановлення справжності інформації може бути реалізована для однієї з моделей взаємодії учасників (абонентів, прикладних процесів) інформаційного обміну – взаємної довіри та захисту, а також взаємної недовіри та взаємного захисту. На рис. 1 розглянуті моделі, відображені схематично [1].

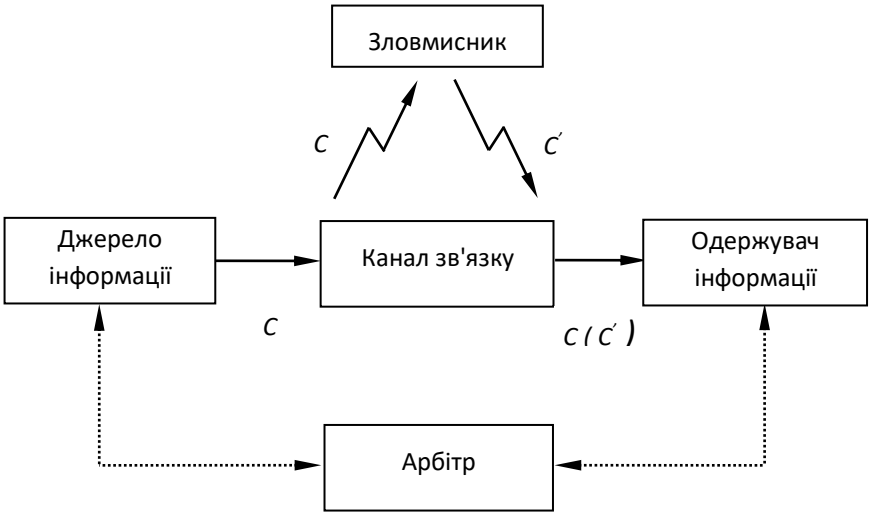


Рис. 1. Модель інформаційного обміну та загроз

В розглянутій моделі представлені чотири діючі сторони: джерело та одержувач інформації, зловмисник (крипто-аналітик) та арбітр. Санкціонованими, тобто тими, що заслуговують довіри, є джерело та одержувач інформації. Зловмисник є основним джерелом загроз порушення автентичності, тобто джерелом створення неправдивих повідомлень (криптограм) *C′*. В цій моделі вважається, що зловмиснику відомі засоби, алгоритми та засоби автентифікації, що реалізувалися в інформаційній системі, та він здатний з високою ймовірністю перехоплювати криптограми *C*, проводити їхній крипто-аналіз, підміну, модифікацію істинних повідомлень, а також створювати неправдиві.

Четвертою стороною є арбітр, що має прийняти рішення відносно джерела інформації, її цілісності та справжності при виникненні спору. В залежності від ступеня довіри його можна вважати як «своїм», так і «стороннім» (несанкціонованим).

На рис. 1 відображена модель, що може бути використана не тільки для процесу передачі повідомлень (криптограм), але і для процесів зберігання об'єктів (програмного забезпечення, файлів та даних) у ЕОМ. В цьому випадку користувач (одержувач) може під час процесу автентифікації перевірити, чи не був об'єкт підмінений або змінений протягом часу, коли він був поза контролем, тобто перевірити його справжність та цілісність перед використанням [2].

У моделі взаємної довіри та захисту припускається, що джерело та одержувач інформації довіряють один одному, бо вони володіють ідентичною конфіденційною інформацією автентифікації. Захист з певною імовірністю забезпечується тільки відносно загроз зловмисника. Арбітр же не може встановити однозначно джерело криптограм, бо одержувач може створити та автентифікувати повідомлення, а після цього стверджувати, що це повідомлення (криптограма) отримане від відповідного джерела, а також модифікувати або підмінити об'єкт. Реалізаційною основою такої системи, наприклад, можуть бути несиметричні системи автентифікації, в яких ключ прямих перетворень, що виконуються джерелом, та ключ перевірки справжності та цілісності повідомлення одержувача не співпадають та жоден з них не може бути обчислений при знанні іншого за прийнятний час.

Сучасні технології потребують досконалих засобів для їх реалізації. Для вибору необхідних засобів проектування і створення системи необхідно визначити основні технічні характеристики майбутньої системи. В залежності від цих характеристик можливо обґрунтувати вимоги до програмних засобів проектування та реалізації проекту системи. Тож розроблена система має володіти такими основними властивостями [3]:

– кінцева програма системи має працювати в графічному середовищі операційних систем Windows 98\ME\NT\2000\XP\Vista\7\8, оптимально використовувати ресурси комп'ютерної системи, такі як оперативна пам'ять, дисковий простір та ресурси центрального процесора;

– зручний віконний графічний інтерфейс з усіма можливостями та перевагами, які дає використання системи GUI (Graphic User Interface – графічний інтерфейс користувача) ОС Windows;

– забезпечення якісної та непомітної для користувача системи обробки та виправлення помилок, що можуть виникнути при роботі системи і викликані такими причинами, як збої в роботі операційної системи, невірні дії користувача або внутрішні помилки в програмі при роботі з файлами та дисковими носіями.

В якості засобу розробки системи було обране інтегроване середовище програмування Microsoft Visual Studio .NET, а саме Visual C++ 7.0 виробництва корпорації Microsoft. Цей програмний пакет являє собою середовище для візуальної розробки інтерфейсної частини програмної

системи, редактор програмного коду, засоби для роботи з програмними компонентами, утиліти для роботи з базами даних та інші додаткові можливості для ефективної розробки програмних засобів [4].

Звичайно, як і кожна інша, дана програмна система має свої недоліки, основними з яких є досить високі вимоги до апаратної та програмної складових комп'ютерної системи, порівняно невелика швидкість компіляції вихідного тексту, а також великі розміри кінцевих виконуваних файлів. Більшість цих недоліків можна виключити при застосуванні додаткових програмних засобів, а за своїми якісними характеристиками система Visual Studio .NET займає одне з провідних місць на ринку систем розробки програмного забезпечення [5].

Розроблена програма призначена для криптографічного шифрування та дешифрування файлів на основі шести алгоритмів: RC4, Blowfish, ГОСТ, AES (Rijndael), Triple DES та VMPC.

Головне вікно програми наведено на рис. 2.

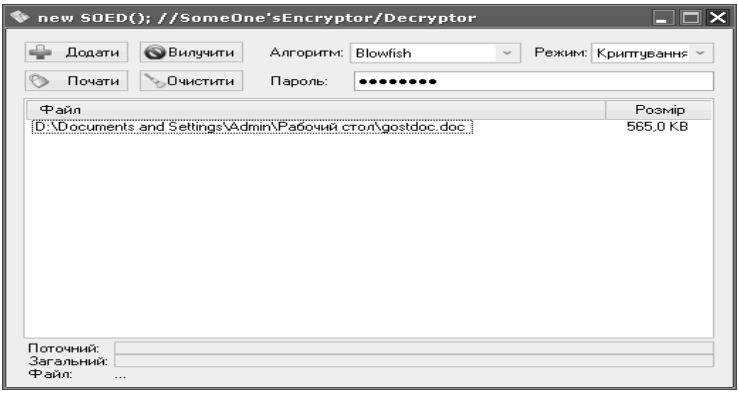


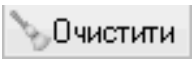
Рис. 2. Головне вікно програми

Зручний інтерфейс забезпечує легку роботу користувача, всі функції здійснюються через головне меню та дублюються кнопками.

– Додати файл.

– Вилучити файл зі списку.

– Почати операцію.



– Очистити список файлів.

Також програма містить поле для вводу ключа:

Поле, в якому відображаються файли, над якими ми проводимо всі операції, наведено на рис. 3.

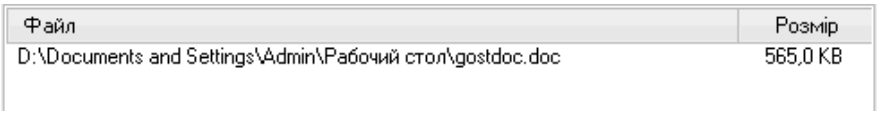


Рис. 3. Поле відображення файлів, над якими проводяться операції

Рядок виконуваної операції з поточним файлом

Для того, щоб провести дії над файлом, його спочатку потрібно відкрити за допомогою кнопки «Додати». В цьому випадку відкриється таке вікно:

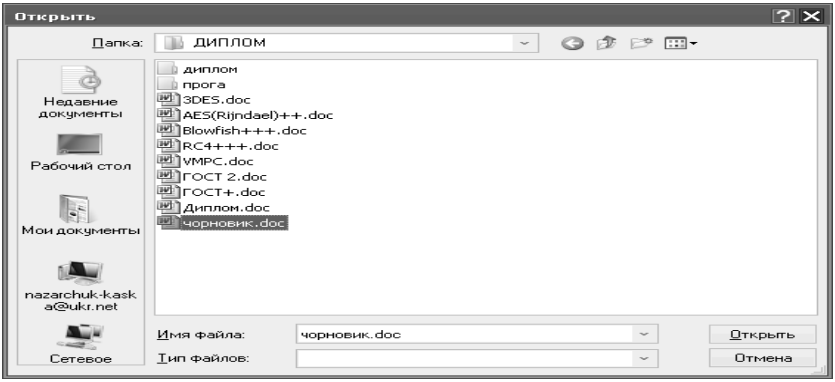
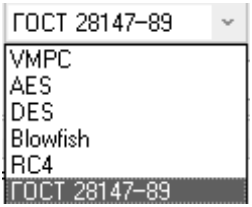


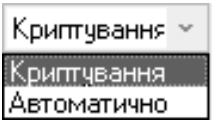
Рис. 4. Вікно вибору файлу для криптування

Вибираємо файл, вміст якого ми маємо зашифрувати (файли можуть містити різні розширення, як текстові, так графічні та відеофайли). В нашому випадку був вибраний текстовий файл «чорновик». Вибравши файл, вводимо ключ, який може містити як літери, так і цифри.

Далі вибираєм алгоритм, за яким хочемо зашифрувати файл:



та вибираєм режим «Криптування»:



натискаємо кнопку «Почати» і вибираємо місце, де буде збережений кодований файл. Створюється ще один файл, який називатиметься «чорновик.doc.<розширення алгоритму>».

Для дешифрації шифрованого файлу потрібно спочатку вибрати шифрований файл, ввести ключ, за яким раніше проводилася шифрація, вибрати режим «Автоматично» та натиснути «Почати».

Програма реалізує всі зазначені при постановці задачі функції і разом з тим, завдяки відкритій технології, є можливість доповнення її функціональних можливостей та вдосконалення програми в цілому. Одним з основних напрямків розвитку програми можна вважати її переведення на платформу Unix-подібних систем Linux, особливо беручи до уваги те, що в сучасних економічних умовах ці операційні системи набули великої популярності через свою дешевизну та надійність. Ще один перспективний напрямок вдосконалення системи – вдосконалення її аналітичних функцій.

1. Чмора А. Л. Современная прикладная криптографія / А. Л. Чмора. – М. : Гелиос АРВ, 2001. **2.** Салома А. Криптография с открытым ключом / А. Салома. – М. : «Мир», 1995. – 318 с. 3**.** Романец Ю. В. Защита информации в компьютерных системах и сетях / Ю. В. Романец, П. А. Тимофеев. – М. : Радио и связь, 2001. **4.** http://cpp.dore.ru/ – бібліотеки, довідники, приклади по різним мовам програмування. **5.** Бобровский С. В. Delphi 7 : учебный курс / С. В. Боровський. – СПб. : Питер, 2003. – 736 с.