МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ СИСТЕМ ЗАХИСТУ ІНФОРМАЦІЇ

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри
_____ С.В. Казмірчук

«_____» _____2020 р.

На правах рукопису
УДК 004.056.5:004.773

КВАЛІФІКАЦІЙНА РОБОТА
ЗДОБУВАЧА ВИЩОЇ ОСВІТИ
ОСВІТНЬОГО СТУПЕНЯ «МАГІСТР»

Тема: Програмний модуль захисту інформації на основі OpenPGP

Виконавець: Д. Г. Денисенко

Науковий керівник: к.т.н., доцент С. Є. Карловський

Нормоконтролер: к.т.н., доцент С. Є. Карловський

Київ 2020

ВСТУП

Актуальність. З кожним днем кількість інформації, яку споживає пересічна людина зростає у геометричній пропорції. Інтернет, телебачення, газети, вдома та на роботі, в метро та на будівлях, повсюди людей оточує інформація. Ведуться інформаційні війни, велика кількість фейків, неперевіреної інформації дуже багать. Через це вартість достовірної, захищеної інформації досягає великих значень. Не просто так кажуть: «Хто володіє інформацією – той володіє світом». Саме тому питання захисту інформації стає все більш важливим.

В Україні питання достовірної та захищеної інформації дуже болюче через усі події, які відбуваються в нашій країні. Проте Україна має багато можливостей для забезпечення захисту інформації. Створена велика законодавча база, створюється міністерства, відділи та багато інших структур, праця яких спрямована на захист інформації. Саме тому донести до людей думку про використання засобів для захисту інформації може бути вирішена.

Звичайні громадяни не замислюються про те, як треба споживати та використати інформацію. Багато хто не знає, що їх повідомлення в усіх додатках може бути використано з метою завдання шкоди. Інформація, що знаходиться на персональних комп’ютерах не захищена достойним чином. Необхідно використовувати додаткові засоби, якщо ви не хочете втратити свою особисту інформацію.

Метою даної роботи – є створення програмного модулю на основі методу *PGP*, для забезпечення захисту інформації під час її передання за допомогою різних додатків.

Для досягнення даної мети буде проведено такі етапи:

- Аналіз законодавчої та нормативно-правової бази Україні, стосовно захисту інформації, відповідальності за її втрату, викрадення тощо.
- Аналіз методу *PGP*, створення схеми його використання.

- Аналіз вразливостей, загроз та атак на метод *PGP*.
- Створення програмного модулю за допомогою мови програмування *Java* на основі методу *PGP*, з його модифікацією.

Об'єктом дослідження даної роботи є –створення програмного модулю на основі *PGP* для захисту інформації.

Предмет дослідження – метод *PGP*.

Методологічною базою дослідження є положення різних концепцій управління безпекою та праці провідних вітчизняних і зарубіжних учених у сфері забезпечення безпеки діяльності підприємства. Так, в роботі були використані загальнонаукові методи дослідження, зокрема:

- Аналізу;
- Логічного узагальнення;
- Порівняння;
- Описового.

Наукова новизна. Вдосконалено метод захисту інформації PGP за рахунок використання сучасних методів симетричного та асиметричного шифрування, метод хешування. Ще одним вдосконаленням є використання двох ключових пар для шифрування та ЕЦП, що дозволило забезпечити захист інформації, та актуалізувати даний метод для сучасного стану розвитку технологій.

На відміну від класичної реалізації де використовувалась велика кількість алгоритмів, в даній праці були використані наступні методи:

- *RSA* із ключем довжиною 4096 біт.
- *AES* із ключем довжиною 256 біт.
- *SHA*-3 із ключем довжиною 512 біт.

За допомогою цього була одержана набагато вищий рівень захищеності особистої інформації, адже це найсучасніші алгоритми, злам яких вимагає дуже великих грошових та часових затрат.

Удосконалено шифрування за допомогою *RSA* через використання двох ключових пар. Одна для шифрування сеансового ключа, інша для шифрування

цифрового підпису. За допомого цього при спілкуванні двох користувачів, за один сеанс відправлення – отримання – відправлення – отримання жоден із ключів не використовується більш ніж один раз, що забезпечує високий рівень цілісності та достовірності даних.

Даний метод можна використовувати як в повсякденному житті, так і на корпоративному рівні, проте є моменти які треба покращувати.

РОЗДІЛ 1. АНАЛІЗ ЗАКОНОДАВЧОЇ БАЗИ УКРАЇНИ ЩОДО ЗАХИСТУ ІНФОРМАЦІЇ ТА АНАЛІЗ МЕТОДУ PGP

1.1. Нормативно-правове регулювання захисту інформації в Україні

Правова форма захисту інформації – це захист інформації, який «*…базується на використанні статей Конституції і законів держави, положень цивільного і кримінального кодексів та інших нормативно-правових документів в галузі інформатики, інформаційних відносин та захисту інформації. Вона регламентує права і обов'язки суб'єктів інформаційних відносин, правовий статус органів, технічних засобів і способів захисту інформації і є базою для створення морально-етичних норм в області захисту інформації*»[4].

Правовий захист інформації загальновизнаний на міжнародному (міжнародні договори, угоди, конвенції, декларації тощо) та державному рівні. На державному рівні правовий захист регулюється державними та відомчими нормативно-правовими актами.

Розроблена на їх основі система законодавчих актів та нормативних, організаційних та адміністративних документів повинна забезпечити організацію ефективного контролю за їх виконанням правоохоронними органами та реалізацію засобів захисту та відповідальності суб'єктів інформаційних відносин. Цю систему можна віднести до морально-етичних норм поведінки, які склалися традиційно або розвиваються з поширенням обчислювальних засобів у суспільстві.

Законодавчі документи щодо захисту інформації:

- Закон України «Про інформацію»;
- Закон України «Про державну таємницю»;
- Закон України «Про захист персональних даних»;

- Закон України «Про електронні документи та електронний документообіг»;
- Закон України «Про електронний цифровий підпис»;
- Закон України «Про Національну програму інформатизації»;
- Закон України «Про захист інформації в інформаційно-телекомунікаційних системах»;
- Положення про порядок здійснення криптографічного захисту інформації в Україні;
- Стаття 31 Конституції України;

Державну політику у сфері захисту інформації відповідно до закону реалізує Державна служба спеціального зв'язку та захисту інформації України (далі – Держспецзв'язку України).

Проаналізуємо наведені вище закони та законодавчі документи більш детально.

Закон України «Про інформацію»

Закон України "Про інформацію"[11] встановлює право громадян України на інформацію та закладає правові основи інформаційної діяльності держави.

Згідно із Законом він «*...встановлює загальні правові підстави для отримання, використання, поширення та зберігання інформації, закріплює право особи на інформацію в усіх сферах суспільного і державного життя України, а також систему інформації, її джерела, визначає статус учасників інформаційних відносин, регулює доступ до інформації та забезпечує її охорону, захищає особу та суспільство від неправдивої інформації. Дія цього Закону поширюється на інформаційні відносини, що виникають у всіх сферах життя і діяльності суспільства та держави при отриманні, використанні, поширенні та зберіганні інформації. Законодавство України про інформацію складається з Конституції України, зазначеного Закону, законодавчих актів з окремих галузей, видів, форм і засобів інформації, ратифікованих Україною*

міжнародних договорів та угод, а також принципів і норм міжнародного права…»[11].

Під поняттям «інформація» цей Закон встановлює відомості про події та явища, що відбуваються у суспільстві, державі та навколишньому природному середовищі, що мають бути задокументовані або публічно оголошені.

Згідно із Законом «*…основними принципами інформаційних відносин є:*

- *гарантованість права на інформацію;*
- *відкритість, доступність інформації та свобода обміну нею;*
- *об'єктивність, вірогідність інформації;*
- *повнота інформації;*
- *точність інформації;*
- *законність отримання, використання, поширення та зберігання інформації…*»[11].

Закон України "Про державну таємницю"

Закон України "Про державну таємницю" регулює соціальні відносини, пов'язані з віднесенням інформації до державної таємниці, засекречуванням, розсекречуванням її матеріальних носіїв і захистом державної таємниці з метою захисту національної безпеки України.

Згідно до Закону «*…Під державною таємницею розглядається вид таємної інформації, що містить відомості у сфері оборони, економіки, науки і техніки, зовнішніх відносин, державної безпеки та охорони правопорядку, розголошення яких може завдати шкоди національній безпеці України та які визнані у порядку, встановленому цим Законом, державною таємницею і підлягають охороні державою. Відносини у сфері охорони державної таємниці регулюються Конституцією України, Законом України "Про інформацію", цим Законом, міжнародними договорами, згода на обов'язковість яких надана Верховною Радою України та відповідними нормативно-правовими актами. Державну політику щодо державної таємниці як складову засад внутрішньої та зовнішньої політики визначає Верховна Рада України. Спеціально*

уповноваженим органом державної влади у сфері забезпечення охорони державної таємниці є Служба безпеки України…»[6].

Закон України «Про електронні документи та електронний документообіг»

Як вказано у Законі України «Про електронні документи та електронний документообіг» «*…встановлює основні організаційно-правові основи електронного документообігу та використання електронних документів. Дія цього Закону поширюється на відносини, що виникають у процесі створення, відправлення, передавання, одержання, зберігання, оброблення, використання та знищення електронних документів. Кабінет Міністрів України та інші органи виконавчої влади в межах повноважень, визначених законом, реалізують державну політику електронного документообігу.*

Державне регулювання у цій сфері спрямовано на:

- *реалізацію єдиної державної політики електронного документообігу;*
- *забезпечення прав і законних інтересів суб'єктів електронного документообігу;*
- *нормативно-правове забезпечення технології оброблення, створення, передавання, одержання, зберігання, використання та знищення електронних документів…*»[8].

Закон України «Про електронний цифровий підпис»

Згідно до Закону України «Про електронний цифровий підпис» він «*…визначає правовий статус електронного цифрового підпису та регулює відносини, що виникають при використанні електронного цифрового підпису.*

Електронний цифровий підпис призначений для забезпечення діяльності фізичних та юридичних осіб, яка здійснюється з використанням електронних документів. ЕЦП використовується фізичними та юридичними особами - суб'єктами електронного документообігу для ідентифікації підписувача та підтвердження цілісності даних в електронній формі…»[7].

Закон України «Про Національну програму інформатизації»

Даним Законом розглядаються загальні принципи формування, впровадження та коригування Національної програми інформатизації, яка, у свою чергу, визначає стратегію вирішення проблеми забезпечення інформаційних потреб та інформаційного забезпечення соціально-економічних, екологічних, науково-технічних, оборонних, національно-культурних та інших видів діяльності у сферах державного значення.

Згідно до Закону «*...Головною метою Національної програми інформатизації є створення необхідних умов для забезпечення громадян та суспільства своєчасною, достовірною та повною інформацією шляхом широкого використання інформаційних технологій, забезпечення інформаційної безпеки держави. Програма спрямована на вирішення таких основних завдань:*

- *формування правових, організаційних, науково-технічних, економічних, фінансових, методичних та гуманітарних передумов розвитку інформатизації;*
- *застосування та розвиток сучасних інформаційних технологій у відповідних сферах суспільного життя України;*
- *формування системи національних інформаційних ресурсів;*
- *створення загальнодержавної мережі інформаційного забезпечення науки, освіти, культури, охорони здоров'я тощо;*
- *створення загальнодержавних систем інформаційно-аналітичної підтримки діяльності органів державної влади та органів місцевого самоврядування;*
- *підвищення ефективності вітчизняного виробництва на основі широкого використання інформаційних технологій;*
- *інтеграція України у світовий інформаційний простір…*»[12].

Закон України "Про захист персональних даних"

Вказано, що цей Закон «*...має важливе суспільно-політичне значення для держави, є свідченням позитивних демократичних зрушень у суспільстві, долученням до кращих правових здобутків людства. Він надає можливість законодавчо врегулювати конституційні положення щодо права кожного на*

захист конфіденційної інформації про особу. Дія цього Закону поширюється на відносини пов'язані із обробленням відомостей про певну фізичну особу в органах державної влади та органах місцевого самоврядування, організаціях, установах і підприємствах усіх форм власності, а також фізичними особами, які виконують професійні обов'язки приватно практикуючих адвоката, нотаріуса, лікаря тощо…»[10].

Закон "Про захист інформації в інформаційно-телекомунікаційних системах"

Цей закон «*...регулює суспільні відносини у сфері захисту інформації в інформаційних, телекомунікаційних та інформаційно-телекомунікаційних системах з метою забезпечення дотримання права власності фізичних і юридичних осіб на інформацію та їх права доступу до неї, а також права власника інформації на її захист. Захист інформації в системі забезпечується запровадженням комплексної системи захисту інформації; дотриманням суб'єктами відносин, пов'язаних з обробкою інформації в системі, законодавства України та нормативних документів у сфері захисту інформації в системі; використанням засобів електронно-обчислювальної техніки, програмного забезпечення, телекомунікаційного обладнання, а також засобів захисту інформації у системі, які відповідають вимогам законодавства України щодо захисту інформації (наявність сертифіката, експертного висновку тощо)…*»[9].

Положення про порядок здійснення криптографічного захисту інформації в Україні

Дане Положення «*...визначає порядок здійснення криптографічного захисту інформації з обмеженим доступом, розголошення якої завдає (може завдати) шкоди державі, суспільству або особі…*»[25].

Стаття 31 Конституція України

Згідно зі Статтею 31 Конституції України «*Кожному гарантується таємниця листування, телефонних розмов, телеграфної та іншої кореспонденції. Винятки можуть бути встановлені лише судом у випадках,*

передбачених законом, з метою запобігти злочинові чи з'ясувати істину під час розслідування кримінальної справи, якщо іншими способами одержати інформацію неможливо…»[15].

У даній статті проголошено, що ніхто не може зазнавати безпідставного посягання на таємницю його кореспонденції. В ній конкретизовано основні форми кореспонденції: телеграфна, листування, телефонні розмови та інша кореспонденція. Під визначення "інша кореспонденція" може бути віднесена, наприклад, електронна пошта. Таємниця кореспонденції належить до особистих таємниць людини.

1.2. Відповідальність за порушення законодавства у сфері захисту інформації

Результатом незаконного втручання у листування електронною поштою можуть бути: порушення інтелектуальної власності, розкриття інформації, що становить державну та комерційну таємницю, розголошення відомостей про приватне життя громадян, матеріальна шкода у вигляді втрат і неотриманих доходів, втрата репутації фірми і т.д. Небезпека даного злочину багатократно зростає, коли злочинець отримує доступ до автоматизованих систем, котрі обслуговують сферу національної оборони, атомної енергетики, транспорту.

1.2.1. Адміністративна відповідальність

Накладення штрафу викликають наступні порушення[14]:

- невиконання законних вимог посадових осіб органів Державної служби спеціального зв'язку та захисту інформації України щодо усунення порушень законодавства про криптографічний та технічний захист інформації, яка є власністю держави, або інформації з обмеженим доступом, вимога щодо захисту якої встановлена законом, та законодавства у сфері надання послуг

електронного цифрового підпису, а також створення інших перешкод для виконання покладених на них;

- неповідомлення або несвоєчасне повідомлення Уповноваженого Верховної Ради України з прав людини про обробку персональних даних або про зміну відомостей, які підлягають повідомленню згідно із законом, повідомлення неповних чи недостовірних відомостей;
- недодержання встановленого законодавством про захист персональних даних порядку захисту персональних даних, що призвело до незаконного доступу до них або порушення прав суб'єкта.

1.2.2. Кримінальна відповідальність

Наступні дії несуть за собою кримінальні покарання[17]:

- незаконне збирання, зберігання, використання, знищення, поширення конфіденційної інформації про особу або незаконна зміна;
- несанкціоноване втручання в роботу електронно-обчислювальних машин (комп'ютерів), автоматизованих систем, комп'ютерних мереж чи мереж електрозв'язку, що призвело до витоку, втрати, підробки, блокування інформації, спотворення процесу обробки інформації або до порушення встановленого порядку її маршрутизації;
- несанкціоновані збут або розповсюдження інформації з обмеженим доступом, яка зберігається в електронно-обчислювальних машинах (комп'ютерах), автоматизованих системах, комп'ютерних мережах або на носіях такої інформації, створеної та захищеної відповідно до чинного законодавства;

- несанкціоновані зміна, знищення або блокування інформації, яка оброблюється в електронно-обчислювальних машинах (комп’ютерах), автоматизованих системах чи комп’ютерних мережах або зберігається на носіях такої інформації;
- несанкціоновані перехоплення або копіювання інформації, яка оброблюється в електронно-обчислювальних машинах (комп’ютерах), автоматизованих системах, комп’ютерних мережах або зберігається на носіях такої інформації, якщо це призвело до її витоку.

1.3. Аналіз методу PGP

Умовні позначення, що застосовуються в даному розділі:

Ka - сеансовий ключ;

KRa - закритий ключ *A*;

KUa - відкритий ключ *A*;

EP - шифрування в схемі з відкритим ключем;

DP - розшифрування в схемі з відкритим ключем;

EC - шифрування в схемі з закритим ключем;

DC - розшифрування в схемі закритим ключем;

H - функція хешування;

|| - конкатенація;

E_{KUb} - шифрування з використанням особистого ключа користувача *B;*

E_{KRa} - шифрування з використанням відкритого ключа користувача *A;*

E_{Ks} - шифрування з використанням сеансового ключа;

Z - стиснення з застосуванням алгоритму *ZIP*;

R64 - перетворення *radix*-64 в *ASCII*.

PGP розшифровується як *Pretty Good Privacy*[21]. Це програмне забезпечення для шифрування, призначене для забезпечення конфіденційності, безпеки та автентифікації систем онлайн-зв'язку. Філ Циммерман (*Phil*

Zimmerman) – автор першої програми *PGP*, і за його словами, вона стала вільно доступною через зростаючий соціальний попит на приватність.

З моменту створення в 1991 році було створено багато версій програмного забезпечення *PGP*. Хоча спочатку використовувався лише для захисту повідомлень електронної пошти та вкладень, *PGP* зараз застосовується до широкого кола випадків використання, включаючи цифрові підписи, повне шифрування диска та захист мережі.

PGP - одна з перших широко доступних програм для реалізації криптографії з відкритим ключем. Це гібридна криптосистема, яка використовує як симетричне, так і асиметричне шифрування для досягнення високого рівня безпеки.

Система *PGP* заснована на алгоритмах, які витримали перевірку практикою і вважаються виключно надійними. Зокрема, в пакет включені алгоритми шифрування з відкритим ключем *RSA*, *DSS* і алгоритм Діффі-Хеллмана, алгоритми традиційного шифрування *CAST*-128, *IDEA* і *TDEA*, а також алгоритм хешування *SHA*-1.

Забезпечення автентичності

Послідовність дій при підписанні документа (рис. 1.1):

- створення повідомлення;
- обчислення 160-бітного хеш-коду повідомлення з використанням *SHA*-1;
- шифрування отриманого хеш-коду за допомогою алгоритму *RSA* на секретному ключі відправника, додавання отриманого шифротексту в початок повідомлення;
- розшифрування і відновлення хеш-коду одержувачем з використанням відкритого ключа відправника;
- генерація нового хеш-коду від отриманого повідомлення і його порівняння з розшифрованим повідомленням, при збігу кодів, повідомлення вважається справжнім.

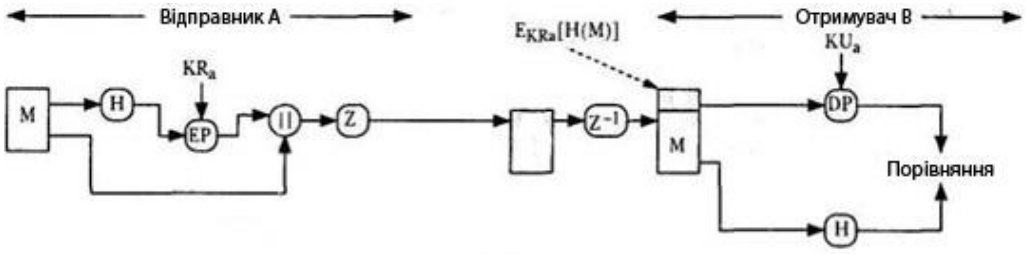

Рис. 1.1 Схема обчислення цифрового підпису

Комбінація алгоритмів *RSA* та *SHA*-1 забезпечує ефективну схему цифрового підпису[22]. З огляду на надійність *RSA*, користувач може бути впевнений в тому, що тільки лише власник секретного ключа міг створити даний підпис. Надійність *SHA*-1 дає одержувачу повідомлення впевненість в тому, що ніхто інший не зміг би створити повідомлення з даним хеш-кодом і, відповідно, з підписом з оригінального повідомлення. Також можливе створення цифрових підписів з використанням комбінації *DSS* і *SHA*-1.

Забезпечення конфіденційності

Схема *PGP* гарантує конфіденційність, що забезпечується шифруванням повідомлень. Для даної мети застосовуються алгоритми симетричного шифрування *CAST*-128, *IDEA* або *3DES*. В тому числі, може використовуватися режим зворотного зв'язку по 64-бітовим блокам шифротекста (режим *CFB*).

У *PGP* кожен ключ застосовується лише один раз. Для кожного повідомлення генерується власний ключ у вигляді випадкового 128-бітного числа. Таким чином, ключ є одноразовим (сеансовим). У зв'язку з тим, що кожен ключ застосовується лише один раз, він приєднується до повідомлення і передається разом з ним. Для захисту ключа, проводиться його шифрування за допомогою відкритого ключа одержувача (рис. 1.2).

Послідовність дій при шифруванні документа:

- генерація відправником повідомлення і 128-бітного сеансового ключа;
- шифрування повідомлення з допомогою *CAST*-128, *IDEA* або *3DES* на отриманому сеансовому ключі;
- шифрування сеансового ключа за допомогою алгоритму *RSA* на відкритому ключі одержувача, приєднання отриманого шифротекста до початку повідомлення;

- розшифрування одержувачем зашифрованого сеансового ключа на його закритому ключі;
- розшифрування одержувачем повідомлення на отриманому сеансовому ключі.

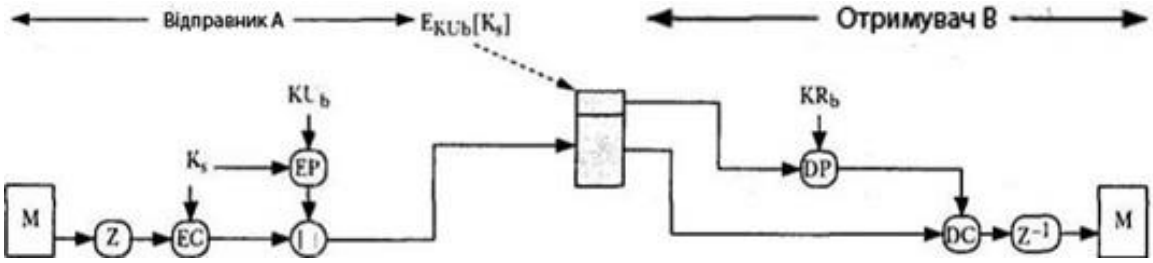

Рис. 1.2 Схема шифрування повідомлення відкритим ключем

В якості альтернативи *RSA* в *PGP* може використовуватися алгоритм обміну ключами Ель-Гамаля.

Забезпечення автентичності та конфіденційності в сукупності

Для сукупного забезпечення автентичності та конфіденційності повідомлення (рис.3.3), спочатку для нього генерується підпис, який додається в початок повідомлення у відкритому вигляді. Потім відкритий текст повідомлення і підпис шифруються за допомогою алгоритму *CAST*-128, *IDEA* або *3DES*, а сеансовий ключ - за допомогою *RSA* (або схеми Ель-Гамаля).

Така схема краще зворотної, коли спершу шифрується вміст повідомлення, а потім отриманий шифротекст підписується. У загальному випадку, зручніше зберігати підпис разом з відкритим текстом повідомлення. Також, з точки зору зручності верифікації третьою стороною, при використанні прямої схеми, третій стороні не потрібно знати ключ шифрування для перевірки справжності.

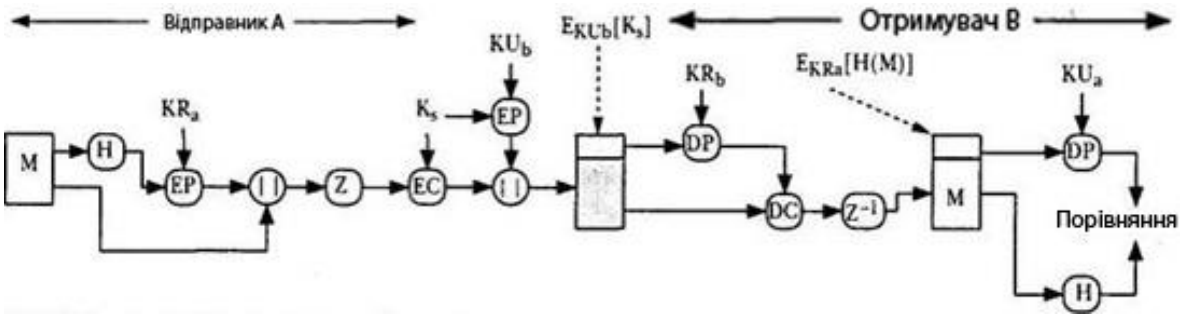

Рис. 1.3 Схема для забезпечення конфіденційності і автентичності повідомлення

Стиснення

Після етапу генерації підпису в *PGP* відбувається стиснення повідомлення з використанням алгоритму *ZIP* для зменшення обсягу даних при передачі пошти і зберіганні прикріплених файлів.

Генерація підпису передує стисненню з наступних причин:

- Можливість зберігання повідомлення з підписом в стислому вигляді. Підписання стислої версії документа спричиняє необхідність його зберігання в стислому вигляді або ж постійного стиснення для проведення процедури верифікації.
- Алгоритм стиснення в *PGP* не є детермінованим і різні його версії можуть застосовувати різні типи стиснення. Таким чином, однакові файли, стисненні різними версіями *PGP*, можуть відрізнятися. Застосування функцій хешування і підпису після стиснення повідомлення призвело б до необхідності використання однакових алгоритмів стиснення у всіх версіях *PGP*.
- Також, шифрування вже стисненого повідомлення виправдано посиленням його криптографічного захисту.

Сумісність на рівні електронної пошти

При використанні *PGP* шифрується, як мінімум, частина блоку, що передається. Якщо потрібен тільки цифровий підпис - шифрується профіль повідомлення; якщо необхідна конфіденційність - відбувається шифрування і повідомлення і значення цифрового підпису.

У підсумку, частина вхідного блоку повідомлення, або весь блок цілком, являє собою потік довільних байтів. Однак, більшість сервісів електронної пошти дозволяють використовувати лише блоки, що складаються з *ASCII* символів. Для виконання цієї вимоги, *PGP* містить механізм конвертації 8-бітного двійкового потоку в потік *ASCII*-символів із застосуванням *radix*-64: кожна група двійкових даних перетворюється в 4 *ASCII*-символу, а потім до них

додається контрольна сума (*CRC*), яка дозволяє виявити помилки при передачі даних.

Перетворення повідомлення в *radix*-64 збільшує його довжину на 33%, проте, відкритий текст стискається, а розміри підпису і сеансового ключа досить малі. Стиснення, таким чином, компенсує збільшення розміру повідомлення при його конвертації в *radix*-64. Конвертація повідомлення відбувається без урахування його вмісту, навіть якщо воно містить тільки *ASCII*-текст. Таким чином, підписане незашифроване повідомлення буде незрозуміле випадковому спостерігачеві. На рис. 1.4 представлений зв'язок між описаними вище компонентами.

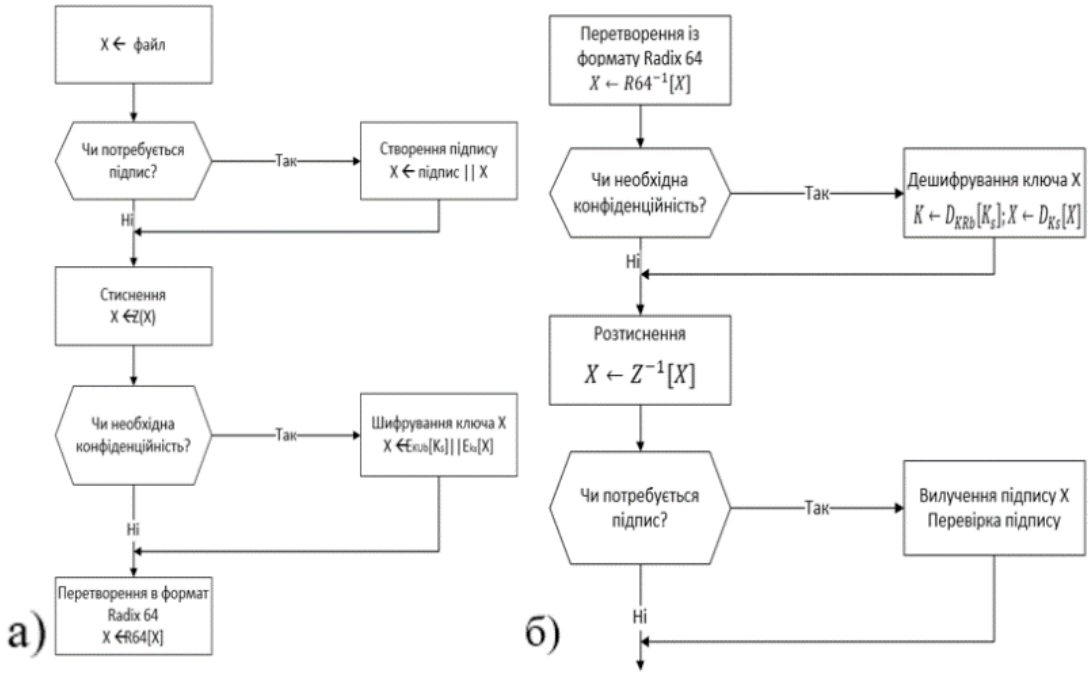

Рис. 1.4 а) Загальна схема відправлення повідомлення; б) Загальна схема отримання повідомлення

Сегментація і зворотна збірка повідомлення

Часто засоби електронної пошти накладають обмеження на розмір повідомлення. Будь-яке довге повідомлення повинно розбиватися на блоки, які передаються окремо. У *PGP* занадто довгі повідомлення розбиваються на блоки достатньої довжини для передачі повідомлення. Розбиття на блоки відбувається після виконання всіх інших операції, включаючи перетворення в *radix*-64.

Завдяки поєднаному використанню симетричного та асиметричного шифрування, *PGP* дозволяє користувачам безпечно обмінюватися інформацією та криптографічними ключами через Інтернет. Як гібридна система, *PGP* виграє як від безпеки асиметричної криптографії, так і від швидкості симетричного шифрування. Окрім безпеки та швидкості, цифрові підписи забезпечують цілісність даних та достовірність відправника. Протокол *OpenPGP* дозволив появу стандартизованого конкурентного середовища, а рішення *PGP* зараз пропонуються багатьма компаніями та організаціями. І все-таки всі програми *PGP*, які відповідають стандартам *OpenPGP*, сумісні між собою. Це означає, що файли та ключі, створені в одній програмі, можуть без проблем використовуватись в іншій. Що стосується недоліків, то системи *PGP* не такі прості у використанні та розумінні, особливо для користувачів з невеликими технічними знаннями. Крім того, довга довжина відкритих ключів багато хто вважає досить незручною.

1.4. Висновки до розділу 1

В розділі 1 було проаналізовано існуючі закони, статті та нормативні документи. Аналіз показав, що на теперішній час в Україні розроблена досить велика законодавча база щодо використання електронної пошти, основана на широкій правовій і нормативній базі та створена інфраструктура, що має забезпечити надійний захист інформації у державі. Також має місце відповідальність за завдані збитки під час реалізації загроз, а саме адміністративні та кримінальні покарання.

Було досліджено та проаналізовано метод *PGP,* з чого бачимо, що він показує себе як дуже надійний метод, який поєднує у собі як шифрування(асиметричне та симетричне) повідомлення так і його цифровий підпис, що забезпечує високий рівень захисту інформації. Існуючі реалізації цього методу вирішують поставлену задачу, а саме, надійне шифрування

повідомлення та його цифровий підпис, зрозумілі у користуванні та забезпечують високу надійність.

РОЗДІЛ 2. АНАЛІЗ ВРАЗЛИВОСТЕЙ, ЗАГРОЗ ТА АТАК НА PGP

2.1. Вразливості PGP

Хоча *PGP* вважається досить надійним варіантом для захисту інформації, проте і в нього є декілька моментів, на які треба звернути уваги.

Перш за все великою проблемою є те, що він розроблений у далекому 1991, а с того часу розвиток технологій далеко пішов уперед. З цього випливає, що деякі криптоалгоритми які у ньому застосовуються, вже втратили свою актуальність чи потребують модифікації. Інша проблема яка стосується цього моменту, це те, що PGP декларує сумісність із великою кількістю різних криптоалгоритмів, а це не є найбільш вдалим виходом. *PGP* підтримує *ElGamal*, *RSA*, р-криві *NIST*, *Brainpool*, *Curve25519*, *SHA*-1, *SHA*-2, *RIPEMD160*, *IDEA*, *3DES*, *CAST5*, *AES*. І це ще не повний список. Якщо за останні 20 років ми дізналися три важливі речі про криптографію, принаймні, дві з них, це те, що узгодження і сумісність – не ведуть до добра. Як правило, недоліки криптосистем з'являються в реалізаціях, а не примітивах, і експансивна криптосумісність збільшує кількість реалізацій.

Або у вас зворотна сумісність з програмою 90-х років, або у вас хороша криптографія; не можна отримати і те, і інше.

Другий момент це використання його разом із електронною поштою. Електронна пошта не надто добрий варіант для захищеного листування. *PGP* сам по собі припускає витік метаданих. При звичайному використанні повідомлення безпосередньо пов'язані з ідентифікаторами ключів, які по всій мережі довіри *PGP* пов'язані з ідентифікатором користувача. Крім того, досить велика частина користувачів *PGP* використовує сервери ключів, які самі

видають в мережу, які користувачі *PGP* обмінюються повідомленнями один з одним.

Найбільш успішна реалізація атаки на електронну пошту яка використовувала *PGP* (та *S/MIME*) це так званий *Efail.* У 2018 році група академіків, на чолі з професором Університету прикладних наук в Мюнстері Себастьяном Шинцель (*Sebastian Schinzel*), попередила про критичні вразливості в складі *PGP* і *S/MIME*. Відомо, що дірки в *PGP і S/MIME* дозволяють прочитати зашифровані таким чином повідомлення в форматі звичайного тексту. Гірше того, проблема поширюється і на старі листи, відправлені та отримані раніше.

Виявилося, що з самими технологіями і криптографією все в порядку, а проблеми криються в імплементації та навколишнього екосистемі. Зокрема, уразливі поштові клієнти (*Thunderbird, Outlook, Apple Mail*) і *PGP*-плагіни для них (*Enigmail*, *Gpg4win* і *GPG Tools*, відповідно).

Третя та наймеш можлива вразливість, це вразливість самого алгоритму шифрування (наприклад *RSA*). Зловмисник який бажає читати повідомлення *PGP*, ймовірно, буде використовувати більш прості засоби, ніж стандартний криптоаналіз, наприклад, криптоаналіз з гумовим шлангом або криптоаналіз чорного мішка (наприклад, установка деякої форми троянського коня або програмного забезпечення/обладнання для реєстрації натискань клавіш на цільовому комп'ютері для захоплення зашифрованих зв'язок ключів та їх паролі). ФБР вже використав цю атаку проти *PGP* в своїх дослідженнях. Однак будь-які такі уразливості застосовні не тільки до *PGP*, але і до будь-якого звичайного програмного забезпечення для шифрування.

2.2. Загрози PGP

2.2.1. Загальні загрози інформаційній безпеці

Загрози інформаційній безпеці проявляються не самостійно, а через можливий контакт із прогалинами в системі захисту або факторами вразливості. Загроза призводить до збоїв у роботі систем на конкретному носії.

Основні уразливості викликані наступними факторами:

- Недоліки програмного або апаратного забезпечення
- Різні характеристики структури автоматизованих систем в потоці інформації
- Деякі операційні процеси системи є недостатніми
- Неточність протоколів та інтерфейсу обміну інформацією
- Складні умови роботи та умови, в яких знаходиться інформація.

Найчастіше джерела загроз спрацьовують для отримання незаконної вигоди після пошкодження інформації. Однак можливий також випадковий ефект загроз через недостатній захист та масовий напад загрозливого фактора.

Вразливі місця можуть бути:

- Об’єктивні.
- Випадкові.
- Суб’єктивні.

Випадкові загрози

Ці фактори змінюються залежно від непередбачених обставин та особливостей інформаційного середовища. Їх майже неможливо передбачити в інформаційному просторі, але ви повинні бути готовими до швидкого їх усунення. Інженерно-технічне розслідування або атака реагування допоможуть пом'якшити наступні проблеми:

1. Несправності системи:
   - Викликані несправностями технічних засобів на різних рівнях обробки та зберігання інформації (в тому числі відповідальних за роботу системи та доступ до неї).
   - Несправності та застарілі елементи (розмагнічування носіїв даних, таких як дискети, кабелі, з'єднувальні лінії та мікросхеми).

- Несправності різного програмного забезпечення, що підтримує всі ланки в ланцюжку зберігання та обробки інформації (антивіруси, прикладні та сервісні програми).
- Несправності допоміжного обладнання інформаційних систем (збої в передачі електроенергії).

2. Фактори, що послаблюють інформаційну безпеку:

- Пошкодження комунікацій, таких як водопостачання, електрика, вентиляція та каналізація.
- Несправності огороджувальних пристроїв (огорожі, стіни в будинках, корпус обладнання, де зберігається інформація).

Об’єктивні загрози

Вони залежать від технічної конструкції обладнання, яке встановлюється на об'єкті, що потребує захисту, а також його характеристик. Уникнути всіх цих факторів неможливо, але часткове їх усунення можна досягти за допомогою інженерних методів у наступних випадках:

1. Пов’язані з технічними засобами викидів:

- Електромагнітні (бічне випромінювання та сигнали від кабельних ліній, елементи технічних засобів).
- Звукові (акустичні або з вібраційними сигналами).
- Електричні (ковзання сигналів у ланцюги електричної мережі через індукцію в лінії та провідники через нерівномірний розподіл струму).

2. Активоване:

- Шкідливе програмне забезпечення, незаконні програми, технологічні виходи з програм, які разом називаються «інструментами імплантації».
- Апаратні імплантати: вводяться безпосередньо в телефонні лінії, електричні мережі або приміщення.

3. Завдяки характеристикам об’єкта, що охороняється:

- Розташування об’єкта (видимість та відсутність контрольованої зони навколо інформаційного об’єкта, наявність вібрацій або

звуковідбивних елементів навколо об’єкта, наявність віддалених елементів об’єкта).

- Організація каналів обміну інформацією (використання радіоканалів, оренда частот або використання спільних мереж).

4. Ті, що залежать від характеристик носіїв:

- Деталі з електроакустичними модифікаціями (трансформатори, телефонні пристрої, мікрофони та гучномовці, дроселі).
- Елементи під впливом електромагнітного поля (носії, мікросхеми та інші елементи).

Суб’єктивні загрози

У більшості випадків уразливість цього підтипу виникає внаслідок неадекватних дій співробітників на рівні розвитку системи зберігання та захисту. Усунення таких факторів можливо за допомогою апаратного та програмного забезпечення:

1. Неточності та грубі помилки, що порушують інформаційну безпеку:

- На етапі завантаження готового програмного забезпечення або попередньої розробки алгоритму, а також під час його використання (можливо, під час щоденного використання або під час введення даних).
- При управлінні програмами та інформаційними системами (труднощі в навчанні роботі з системою, індивідуальне налаштування служб, маніпулювання інформаційними потоками).
- Під час використання технічного обладнання (під час увімкнення або вимкнення, використання пристроїв для передачі або отримання інформації).

2. Несправності системи в інформаційному середовищі:

- Режим захисту персональних даних (проблема може бути спричинена звільненими працівниками або поточними працівниками в неробочий час, коли вони отримують несанкціонований доступ до системи).

- Режим безпеки та безпеки (при доступі до об'єктів або технічних пристроїв).
- Під час роботи з пристроями (неефективне використання енергії або неналежне обслуговування обладнання).
- Під час роботи з даними (зміна інформації, її збереження, пошук та знищення даних, усунення дефектів та неточностей).

2.2.2. Загрози безпосередньо для PGP

- Ризик неправильного використання - з електронною поштою завжди залишається ризик того, що хтось надішле вам конфіденційну інформацію в чистому тексті - просто тому, що може, тому що це простіше, тому що вони ще не мають вашого відкритого ключа і не турбуються про це, або просто помилково. Можливо, навіть тому, що вони знають, що таким чином можуть розлютити вас - і виправдовуються, вдаючи за некомпетентність. Деякі люди навіть встигають відповідати незашифрованими на зашифроване повідомлення, хоча програмне забезпечення *PGP* повинно заважати їм це робити.[28]
- Формат *OpenPGP* - завдяки легко виявленому формату повідомлень OpenPGP це легка вправа для будь-якого виробника апаратного забезпечення Deep Packet Inspection запропонувати можливість виявлення повідомлень, зашифрованих PGP, де завгодно в потоці Інтернет-комунікацій, не тільки в межах SMTP. Отже, використовуючи PGP, ви робите себе видимим. Stf пропонує використовувати формат обгортання, який неможливо виявити.
- Дані про спілкування - Навіть якщо ви використовуєте *PGP*, можна простежити, з ким ви говорите, коли і як довго. Можна здогадатися, про що ви говорите, тим більше, що деякі з вас вкладуть щось значуще

в незашифрований заголовок. *PGP* пропонує засоби шифрування рядка заголовку.

- Відсутність прямої секретності – зловмисник зберігає повну колекцію всіх повідомлень *PGP*, що надсилаються через Інтернет, на випадок, якщо одного разу необхідні приватні ключі можуть потрапити до його рук. Це має сенс, оскільки *PGP* не має прямої секретності.. Технічно *PGP* здатний освіжити підключі, але це настільки втомливо, що не практикується - не кажучи вже про те, що практикується так, як повинно бути: принаймні щодня.
- Час модернізувати саму криптографію – зловмисник також може чекати дня, коли криптографія *RSA* буде зламана, і всі зашифровані повідомлення будуть читатись із зворотною силою. Той, хто зафіксував якомога більше трафіку *PGP*, одного разу отримає від цього стратегічні переваги. За словами криптографів, цей день може бути ближчим, ніж вважають прихильники *PGP*, оскільки незабаром криптографія *RSA* може бути зламана.
- Використання міжсерверної магістралі - подорожуючи усталеними та перевіреними шляхами електронної пошти, *PGP* надмірно суперекспонується. Було б набагато краще, якби той самий *PGP* передавався від комп'ютера до комп'ютера безпосередньо. Можливо, навіть вбудований у картину, фільм чи музичний твір за допомогою стеганографії.
- Мережа довіри, якій не можна довіряти – вона э загальнодоступна для здобуття даних; має багато одиничних точок відмови (соціальні центри з компрометованими ключами); погано масштабується для глобального використання. Отже, насправді це ще три причини не використовувати *PGP*, але оскільки ви можете використовувати *PGP* без мережі довіри, ми будемо вважати їх однією.
- *PGP* поєднує невідмову та автентифікацію.

- Статистичний аналіз - вгадування розміру повідомлень. Особливо для чатів та віддаленого керування комп’ютером відомо, що розмір і частота невеликих зашифрованих фрагментів може спостерігатися досить довго, щоб здогадатися про вміст. Це проблема з *SSH* більше, ніж з *PGP*, але також *PGP* був би розумнішим, якби повідомлення були заповнені певними стандартними розмірами, завдяки чому вони виглядали б єдиними.
- Обмін повідомленнями за допомогою PGP недоцільний. Це громіздка процедура конфігурації та неефективна, оскільки кожна копія перешифровується. Ви також можете спільно використовувати один і той же ключ, але це інша громіздка процедура конфігурації. Інструменти спілкування наступного покоління автоматизують створення та розподіл ключів групових сеансів, тому вам не потрібно турбуватися.
- Зберігання чернетки в чистому тексті на сервері - одна особа впроваджує *IMAP*, інша реалізує підтримку *PGP*, і вони ніколи не натрапляють один на одного і усвідомлюють, що за замовчуванням поведінка поштового агента, який підтримує обидва, - робити те, що ні в якому разі не слід робити: відправляти незашифровану пошту на сервер. Це робить усі зусилля щодо використання *PGP* марними.
- *DNS* та *X.509* вимагають стільки роботи. Це може здатися не пов’язаним, але *PGP* базується на електронній пошті, і електронна пошта без потреби примушує до нас залежність від *DNS* і *X.509* (стандарт сертифікації *TLS* і *HTTPS*, який змушує нас потребувати сертифікатів, підписаних органом влади, а потім може в будь-якому випадку обдурити і зламати). І те, і інше коштує грошей для участі в них, і їм слід докладно керувати.
- Можливі цілеспрямовані атаки на ключові ідентифікатори *PGP* - *PGP* має погану звичку використовувати усічені відбитки пальців як ідентифікатори ключів, упорядковуючи ключі в своїй базі даних за

коротким ідентифікатором ключа та маючи справу з ключами з тим самим коротким ідентифікатором ключа, що, ймовірно, однаковий, хоча створити нову пару ключів не так складно який вирішує той самий ідентифікатор ключа, що і існуючий. Здається, це проблема навіть при довгих ключових ідентифікаторах. Використовуючи короткі ідентифікатори для обслуговування ключів, реалізації програмного забезпечення *PGP* роблять це неправильно. Одним із можливих наслідків цього є те, що користувачів можуть обдурити, прийнявши помилковий ключ заміни з сервера ключів, або якимось іншим чином заплутали управління ключами до такої міри, що пошкодила шлях зв'язку, який раніше був безпечним і дозволив людині посередині втрутитись.

2.3. Атаки на PGP

2.3.1. Зламати асиметричне шифрування

- Можна підбирати можливі ключі за допомогою (відомого) відкритого ключа одержувача до знаходження збіги. Ефективність цього нападу значно зменшується за допомогою довільної надлишкової інформації, що вводиться в процесі симетричного шифрування.
- Спробувати математично зламати асиметричне шифрування.
- Криптоаналіз асиметричного шифрування. Криптоаналіз одного зашифрованого тексту дасть загальний підхід для зламу *RSA*.
- Таймінг-атаки на *RSA*. Однак ці атаки вимагають низкоуровневого контролю за комп'ютером одержувача в той час, як він розшифровує послання.

2.3.2. Зламати симетричний ключ

- Зламати симетричний ключ за допомогою атаки «в лоб». Всі алгоритми шифрування за допомогою симетричного ключа для *PGP* мають ключі розміром не менше 128 біт. Це робить нереальною лобову атаку.
- Криптоаналіз шифрування за допомогою симетричного ключа Алгоритми шифрування з допомогою симетричного ключа (підтримувані *PGP 5.x*): *IDEA*, 3-*DES*, *CAST*-5, *Blowfish* і *SAFER-SK* 128. Ефективні методи для загального криптоаналіза цих алгоритмів не відомі.
- Змусити (обманом) відправника використовувати відкритий ключ одержувача, чий закритий ключ відомий, для шифрування повідомлення
- Змусити відправника повірити, що якийсь підроблений ключ (секретний ключ якого відомий) - це ключ адресата.
- Переконати відправника зашифрувати послання не одним-єдиним ключем: справжнім ключем одержувача і іншим, секретний ключ якого відомий.
- Зробити так, щоб повідомлення було зашифровано деяким іншим відкритим ключем, походження якого відправнику невідомо Цього можна домогтися, запустивши програму, яка змусить користувача повірити, що використовується правильний ключ, тоді як насправді шифрування проводиться іншим ключем.
- Якщо адресат сліпо підписує зашифрований ключ, то він мимоволі відкриває незашифрований ключ. Ключ досить короткий, тому хешування не обов'язково перед підписуванням. Або якщо хеш-функція повідомлення відповідає зашифрованого ключа, то одержувачу можна запропонувати підписати повідомлення (або його хеш-функцію).

- Незашифрований симетричний ключ повинен зберігатися десь в пам'яті під час шифрування і дешифрування. Якщо пам'ять доступна, це дає привід заволодіти ключем і прочитати послання.

2.3.3. Добути секретний ключ одержувача

- Розкладання на модулі *RSA*. Цей спосіб вимагає рішення безлічі теоретичних питань, які в даний час представляються дуже складними.
- Отримати секретний ключ одержувача з його зв'язки ключів
- Здобути зашифровану зв'язку ключів одержувача
- Скопіювати його з жорсткого диска користувача
- Скопіювати його резервну копію
- Контроль мережевого трафіку
- Впровадити вірус або закладку для розкриття копії зашифрованого секретного ключа

2.3.4. Атака Efail

По суті, атака має на увазі, що зловмисник, який отримав в своє розпорядження зашифровані листи, оснастить їх *HTML*-тегами і хитрістю змусить оригінального відправника (або одного з реципієнтів) відкрити стало шкідливим послання. Експлуатація проблем тісно пов'язана з тим, як поштові клієнти і їх плагіни обробляють *HTML* і посилання на зовнішні джерела, наприклад, зображення і стиль, завантажені з зовнішніх *URL*.

Справа в тому, що поштові клієнти, як правило, сконфігуровані таким чином, щоб автоматично дешифрувати вміст захищених листів. Якщо клієнт при цьому ще й автоматично завантажує дані з зовнішніх джерел, цим можуть скористатися атакуючі, надіславши своєї мети модифіковану версію зашифрованого послання і, зрештою, отримавши його розшифровану копію.

Так, атакуючий може використовувати теги *img* або *style*, розмістивши їх в незашифрованому частині *HTML*-листів (а саме в *MIME*-заголовках), як показано на ілюстрації нижче. Фактично, лист перетворюється в *multipart HTML*-повідомлення, і всередині тега міститься зашифрований текст. У підсумку, коли вразливий клієнт буде розшифровувати це послання, він перейде до автоматичній обробці *HTML* і відправить вже дешифрований текст зловмисникові в рамках використаного тега.

На даний момент розробники, кажуть, що ця вразливість вже виправлена у новіших версіях поштових клієнтів.

2.3.5. Інші методи атак

- Прочитати послання після того, як воно буде розшифровано одержувачем
- Скопіювати повідомлення з жорсткого диска або з віртуальної пам'яті комп'ютера
- Копіювати повідомлення з резервної копії, що зберігається на магнітній стрічці
- Контроль мережевого трафіку
- Використовувати кошти прийому електромагнітного випромінювання для зчитування повідомлення, виведеного на екран
- Отримання повідомлення з пристроїв виведення
- Отримати текст з паперової роздруківки
- Отримати текст з фоточутливого барабана принтера
- Підслухати передачу інформації з комп'ютера на принтер
- Отримати інформацію з пам'яті принтера.

2.4. Висновки до розділу 2

Отже, проаналізувавши можливі вразливості, загрози та атаки для *PGP*, можна зробити висновок, що загалом чогось особливо критичного або специфічного немає. Для доступу до даних необхідно зламати асиметричний чи симетричний ключ, що при коректному їх використанні виглядає неможливим. Проблемою залишається людський фактор, але це стосується будь-якої сфери життя. Найбільш критичним виявляється атака Efail, проте по-перше вона стосується лише використання електронної пошти, із декількома типами додатків. По-друге розробники цих додатків заявляють, що дана вразливість вже виправлена.

Загалом при розумному використані *PGP* показує себе із дуже гарного боку, та не несе за собою критичних недоліків вже понад 30 років.

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДУ PGP

3.1. Опис алгоритмів які використовуються у PGP

В класичній реалізації *PGP*[24] для шифрування сеансового ключа та ЕЦП використовують *RSA* або алгоритм Ель-Гамаля, проте маючи на меті прийти до універсальності будемо використовувати тільки алгоритм *RSA* з довжиною ключа 4096 біт.

Для шифрування самого тексту використовують симетричні алгоритми шифрування такі як *AES, CAST5, 3DES, IDEA, Twofish, Blowfish, Camellia*. Проте багато з цих алгоритмів чи застаріли, чи втратили свою актуальність. Будемо використовувати *AES* з довжиною ключа 256 біт.

Для цифрового підпису використовують алгоритми хешування такі як *MD5*, *SHA*-1, *RIPEMD*-160, *SHA*-256, *SHA*-384, *SHA*-512. Будемо використовувати *SHA*-3 (*Keccak*) з довжиною ключа 512 біт.

3.1.1. Алгоритм RSA

RSA (абревіатура від прізвищ *Rivest, Shamir* і *Adleman*) - криптографічний алгоритм з відкритим ключем, який базується на обчислювальної складності задачі факторизації великих цілих чисел[20].

Алгоритм RSA включає в себе чотири етапи: генерація ключів, передача ключів, шифрування і розшифрування. Криптографічні системи з відкритим ключем використовують так звані односторонні функції.

Одностороння функція (англ. *One-way function*) - математична функція, яка легко обчислюється для будь-якого вхідного значення, але завдання знаходження аргументу по заданому значенню функції відноситься до класу *NP*-повних задач. Під односторонністю розуміється не теоретична

односпрямованість, а практична неможливість обчислити зворотнє значення, використовуючи сучасні обчислювальні засоби, за доступний для огляду інтервал часу.

В основу криптографічної системи з відкритим ключем[18] *RSA* покладена складність завдання факторизації похідної двох великих простих чисел. Для шифрування використовується операція піднесення до степеня за модулем великого числа. Для дешифрування (зворотної операції) за розумний час необхідно вміти обчислювати функцію Ейлера від даного великого числа, для чого необхідно знати розкладання числа на прості множники. У криптографічній системі з відкритим ключем кожен учасник має як відкритий ключ (англ. *Public key*), так і закритий ключй (англ. *Private key*). У криптографічній системі *RSA* кожен ключ складається з пари цілих чисел. Кожен учасник створює свій відкритий і закритий ключ самостійно. Закритий ключ кожен з них тримає в секреті, а відкриті ключі можна повідомляти будь-кому або навіть публікувати їх. Відкритий і закритий ключі кожного учасника обміну повідомленнями в криптосистемі *RSA* утворюють «узгоджену пару» в тому сенсі, що вони є взаємно зворотними. Тобто для будь-яких допустимих пар відкритого і закритого ключів (*p, s*) існують відповідні функції шифрування $E_p(x)$ та дешифрування $D_s(x)$ такі, що для будь-якого повідомлення $m \in M$, де M – множина допустимих повідомлень, $m = D_s\left(E_p(m)\right) = E_p(D_s(m))$.[2]

Створення відкритого та закритого ключів:

1. Вибираються два різних випадкових простих числа p і q заданого розміру (наприклад, 4096 біта кожне).
2. Обчислюється їх добуток $n = p \cdot q$, яке називається модулем.
3. Обчислюється значення функції Ейлера від числа n:

$\varphi(n) = (p-1) \cdot (q-1)$.

4. Вибирається ціле число $e(1 < e < \varphi(n))$, взаємно просте зі значенням функції $\varphi(n)$. Зазвичай в якості *e* беруть прості числа, що містять невелику кількість одиничних біт в двійковій запису.

- Число e називається відкритою експонентою.
- Час, необхідний для шифрування з використанням швидкого зведення в ступінь, пропорційно числу одиничних біт в *e*.
- Занадто малі значення *e*, наприклад 3, потенційно можуть послабити безпеку схеми *RSA*.

5. Обчислюється число *d*, мультиплікативно зворотне до числа *e* по модулю $\varphi\ (n)$, тобто число, яке задовольняє порівнянні:

$$d \cdot e \equiv 1\ (mod \varphi\ (n)).$$

6. Пара $\{e, n\}$ публікується в якості відкритого ключа *RSA*.

7. Пара $\{d, n\}$ грає роль закритого ключа *RSA* і тримається в секреті.

Передача ключів.

Припустимо, що Відправник хоче відправити Одержувачу інформацію. Якщо вони вирішують використовувати *RSA*, Відправник повинен знати відкритий ключ отримувача для того щоб зашифрувати повідомлення, а Одержувач повинен використовувати свій закритий ключ для розшифрування повідомлення. Щоб дозволити Відправникові відправляти свої зашифровані повідомлення, Одержувач передає свій відкритий ключ Відправникові через надійний, але не обов'язково секретний маршрут. Закритий ключ Отримувача ніколи нікому не передається. [1]

Шифрування.

Припустимо, Відправник хоче послати Одержувачу повідомлення m. Повідомленнями є цілими числами в інтервалі від 0 до *n-1*, тобто $m \in Zn$.[3]

Алгоритм:

- Взяти відкритий ключ (e, n) отримувача

- Взяти відкритий текст *m*
- Зашифрувати повідомлення з використанням відкритого ключа отримувача:

$$c = E(m) = m^e \bmod n$$

Дешифрування.

Алгоритм:

- Прийняти зашифроване повідомлення *c*
- Взяти свій закритий ключ (d, n)
- Застосувати закритий ключ для розшифрування повідомлення:

$$m = D(c) = c^d \bmod n$$

Криптографічна стійкість.

Стійкість алгоритму ґрунтується на складності обчислення зворотної функції до функції шифрування.

$$c = E(m) = m^e \bmod n$$

Для обчислення *m* по відомим *c*, *e*, *n* потрібно знайти такий *d*, щоб

$$e \cdot d \equiv 1 \ (mod \varphi(n)),$$

тобто

$$d \equiv e - 1 \ (mod \varphi(n)).$$

Обчислення зворотного елемента по модулю не є складним завданням, проте зловмисникові невідомо значення $\varphi(n)$. Для обчислення функції Ейлера від відомого числа n необхідно знати розкладання цього числа на прості множники. Знаходження таких множників і є складним завданням, а знання цих множників - «потайними дверцями», яка використовується для обчислення *d* власником ключа[16].

У 2010 році групі вчених з Швейцарії, Японії, Франції, Нідерландів, Німеччини і США вдалося успішно обчислити дані, зашифровані за допомогою криптографічного ключа стандарту RSA довжиною 768 біт. Знаходження простих співмножників здійснювалося загальним методом решета числового поля. За словами дослідників, після їх роботи в якості надійної системи

шифрування можна розглядати тільки RSA-ключі довжиною 1024 біта і більш. Причому від шифрування ключем довжиною в 1024 біт варто відмовитися в найближчі три-чотири роки. З 31 грудня 2013 року браузери Mozilla перестали підтримувати сертифікати що засвідчують центрів з ключами RSA менше 2048 біт. А на сьогоднішній день доцільно користуватися ключами довжиною 4096 бітів.

3.1.2. Алгоритм AES

Advanced Encryption Standard (*AES*) - симетричний алгоритм блочного шифрування, прийнятий урядом США як стандарт в результаті конкурсу, проведеного між технологічними інститутами. Він замінив застарілий *Data Encryption Standard (DES)*, який більше не відповідав вимогам мережевої безпеки, що ускладнилися в XXI столітті[27].

Алгоритм AES представляє блок даних у вигляді двовимірного байтового масиву розміром $4x4$. Всі операції проводяться над окремими байтами масиву, а також над незалежними стовпцями і рядками.[5]

У кожному раунді алгоритму виконуються наступні перетворення:

1. Операція *SubBytes*, що представляє собою табличну заміну кожного байта масиву даних.
2. Операція *ShiftRows*, яка виконує циклічний зсув вліво всіх рядків масиву даних, за винятком нульовий. Зрушення *i*-го рядка масиву (для $i = 1,2,3$) проводиться на *i* байт.
3. Операція *MixColumns*. Виконує множення кожного стовпця масиву даних на фіксований поліном

   $a(x): a(x) = 3x^3 + x^2 + x + 2$

   Множення виконується по модулю

   $x^4 + 1$

4. Операція *AddRoundKey* виконує накладення на масив даних матеріалу ключа. А саме, на *i*-ий стовпець масиву даних ($i = 0 \dots 3$) побітової логічною операцією «виключне або» (*XOR*) накладається певне слово розширеного ключа W_{4r+i}, де *r* - номер поточного раунду алгоритму, починаючи з 1.

Кількість раундів алгоритму *R* залежить від розміру ключа наступним чином: 128 біт – 10; 192 біт – 12; 256 біт – 14.

Перед першим раундом алгоритму виконується попереднє накладення матеріалу ключа за допомогою операції *AddRoundKey*, яка виконує накладення на відкритий текст перших чотирьох слів розширеного ключа $W_0 \dots W_3$. Останній же раунд відрізняється від попередніх тим, що в ньому не виконується операція *MixColumns*[13].

Дешифрування.

Розшифрування виконується застосуванням зворотних операцій в зворотній послідовності. Відповідно, перед першим раундом розшифрування виконується операція *AddRoundKey* (яка є зворотною самої себе), що виконує накладення на шифротекст чотирьох останніх слів розширеного ключа, тобто $W_{4R} \dots W_{4R+1}$. Потім виконується *R* раундів розшифрування, кожен з яких виконує наступні перетворення:

1. Операція *InvShiftRows* виконує циклічний зсув вправо трьох останніх рядків масиву даних на ту ж кількість байт, на яке виконувався зсув операцією *ShiftRows* при зашифрованими.
2. Операція *InvSubBytes* виконує побайтово зворотну табличну заміну
3. Операція *AddRoundKey*, як і при шифруванні, виконує накладення на оброблювані дані чотирьох слів розширеного ключа $W_{4R} \dots W_{4R+3}$. Однак, нумерація раундів *r* при розшифрування проводиться в зворотній бік – від *R-1* до 0.
4. Операція *InvMixColumns* виконує множення кожного стовпця масиву даних аналогічно прямої операції *MixColumns*, однак,

множення проводиться на поліном $a^{-1}(x)$, визначений таким чином:

$$a^{-1}(x) = Bx^3 + Dx^2 + 9x + E.$$

Аналогічно шифруванню, останній раунд дешифрування не містить операцію *InvMixColumns*.

Розширення ключа.

AES використовує ключі шифрування трьох фіксованих розмірів: 128, 192, і 256 біт. Залежно від розміру ключа, конкретний варіант алгоритму AES може позначатися як *AES*-128, *AES*-192 і *AES*-256 відповідно[19].

Завдання процедури розширення ключа полягає у формуванні потрібного кількість слів розширеного ключа для їх використання в операції *AddRoundKey*. Як було сказано вище, під «словом» тут розуміється 4-байтний фрагмент розширеного ключа, один з яких використовується в первинному накладенні матеріалу ключа і по одному - в кожному раунді алгоритму. Таким чином, в процесі розширення ключа формується $4 * (R + 1)$ слів.[23]

Розширення ключа виконується в два етапи, на першому з яких виробляється ініціалізація слів розширеного ключа (що позначаються як W_i): перші N_k (N_k- розмір вихідного ключа шифрування *K* в словах, тобто 4, 6 або 8) слів W_i (тобто $i = 0 \dots N_{k-1}$) формуються їх послідовним заповненням байтами ключа.

Наступні слова W_i формуються такою послідовністю операцій для кожного $i = N_k \dots 4 * (R + 1) - 1$:

1. Ініціалізується тимчасова змінна *T*: $T = W_{i-1}$.
2. Дана змінна модифікується в такий спосіб:
   - якщо *i* кратно N_k, то: $T = SubWord\left(RotWord\,(T)\right)$ÅRCn$_i$/ N_k;

     константи *RCn* представляють собою слова, в яких всі байти, крім першого є нульовими, а перший байт має значення $2^{n-1} mod 256$.

     Операція *RotWord* побайтно обертає вхідний слово на 1 байт вліво.;

- якщо N_k = 8 і $imodN_k = 4$, то: $T = SubWord\ (T)$;
- в інших випадках модифікація змінної *T* не виконується.

3. Формується і-е слово розширеного ключа: $W_i = W_i - N_k$ÅT.

3.1.3. Алгоритм SHA-3

SHA-3 (Secure Hash Algorithm Version 3), також іменований *Keccak* (Кечак), являє собою односпрямовану функцію для створення цифрових відбитків обраної довжини (в стандарті прийняті 224, 256, 384 або 512 біт) з вхідних даних будь-якого розміру, розробленим групою авторів на чолі з Йоаном Дайменом в 2008 році і прийнятим в 2015 в якості нового стандарту *FIPS*. Алгоритм працює за допомогою функції перемішування зі стисненням до обраного розміру "криптографічного губкою"[29].

Keccak заснований на конструкції *Sponge*. Це означає, що для отримання хешу потрібно виконати наступні нехитрі дії: Взяти вихідне повідомлення *M* і доповнити його до довжини кратної r. У вигляді формули їх можна зобразити таким чином: $M = M \,||\, 0x01 \,||\, 0x00 \,||\, .. \,||\, 0x00 \,||\, 0x80$. Тобто до повідомлення дописується одиничний байт, необхідну кількість нулів і завершується байт із значенням 0x80. [26]

Все вищесказане справедливо тільки для випадків, коли додається більше одного байта.Однак в разі, якщо необхідно доповнити всього один байт, то досить додати лише 0x81.

Потім для кожного блоку *Mi* довжиною *r* біт виконуємо:

1. Додавання за модулем 2 з першими *r*-битами набору початкових станів *S*. Перед використанням цієї функції всі елементи *S* дорівнюватимуть нулю.
2. *N* раз застосовуємо до отриманих в результаті даними функцію *f*. Набором початкових станів *S* для блоку $Mi + 1$ буде результат останнього раунду блоку *Mi*.

3. Після того як всі блоки *Mi* закінчяться взяти підсумковий результат і повернути його в якості хеш-значення.

Хеш-функція *Keccak* реалізована таким чином, що функцію перестановки *f*, яка застосовується для кожного блоку *Mi*, користувач може вибирати самостійно з набору визначених функції $b = \{f-25, f-50, f-100, f-200, f-400, f-800, f-1600\}$.

Для використання функції *f-800*, необхідно вибрати такі *r* і *c*, щоб виконувалося рівність $r + c = 800$.

Крім того, змінюючи значення *r* і *c*, ви тим самим змінюєте кількість раундів вашої хеш-функції.[30]

Оскільки кількість раундів обчислюється за формулою $n = 12 + 2l$, де $2l = (b / 25)$. Так для $b = 1600$, Кількість раундів дорівнює 24.

Однак хоча користувач в праві вибирати для своєї реалізації будь-яку із запропонованих авторами функцій, слід відзначити що в якості стандарту *SHA-3* прийнята тільки функція *Keccak-1600* і автори всіляко рекомендують користуватися тільки нею. Так як основні значень для хешів різної довжини автори вибрали наступні параметри:

SHA-224: r = 1156, c = 448 (повернути перші 28 байт результату)

SHA-256: r = 1088, c = 512 (повернути перші 32 байт результату)

SHA-384: r = 832, c = 768 (повернути перші 48 байт результату)

SHA-512: r = 576, c = 1024 (повернути перші 64 байт результату)

Схема *SHA*-3 (*Keccak*) складається з двох етапів:

- *Absorbing* (вбирання). Оригінал тексту *M* піддається багатораундовим перестановкам *f*.
- *Squeezing* (віджимання). Вивід отриманого в результаті перестановок значення *Z*.

На етапі *Absorbing* виробляється обчислення хеш значення. А на етапі *Squeezing* вивід результатів до тих пір поки не буде досягнута необхідна довжина хешу.

3.2. Відмінності класичної реалізації PGP від запропонованого варіанту

1. Класична реалізація *PGP* має на меті забезпечити сумісність із великою кількість алгоритмів шифрування та хешування. В той час коли багато з них вже застарілі, чи не актуальні. Саме тому у запропонованому варіанті для створення ключової пари використовується тільки алгоритм *RSA* із довжиною ключа 4096 біт. Також одним із нововведень є використання для кожного користувача двох ключових пар. Одна з яких використовується тільки для підписання документа, а інша для шифрування.
2. Для створення електронно-цифрового підпису, а саме його хешу, використовується сучасний алгоритм *SHA*-3 з довжиною ключа 512 біт.
3. Для шифрування відкритого тексту застосовується алгоритм *AES* із довжиною ключа 256 біт.
4. Для створення сеансового ключа використовується стійкий генератор псевдовипадкових чисел.
5. Використання цифрового підпису та шифрування тільки разом, адже використання чогось одного не забезпечує повноти захисту інформації.

3.3. Програмний модуль на основі PGP

Для реалізації цього програмного модуля було обрано мову програмування – *Java*. Симетричне та асиметричне шифрування, хешування та створення сеансового ключа (за допомогою стійкого генератора псевдовипадкових чисел) виконувалось за допомогою бібліотек *Java* –

java.security та *javax.crypto*. Для швидкого розгортання та використання використовувався *Java Spring*.

На рис. 3.1. зображена блок-схема розробленої програми.

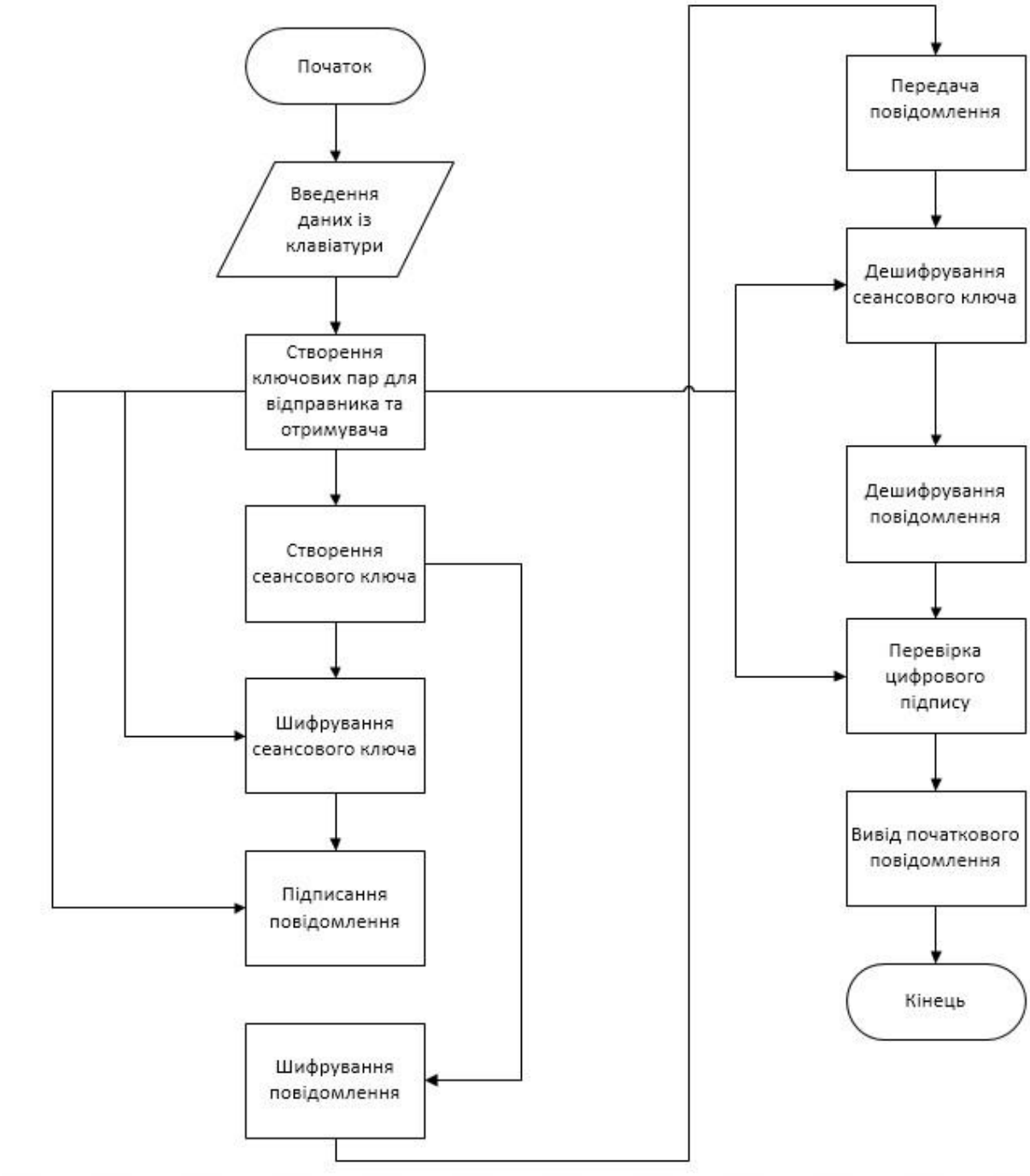

Рис. 3.1. Блок-схема програми

В якості повідомлення використовується текст введений з клавіатури, що показано на рис. 3.2.

```
Please, enter data that you want to encrypt:
Denysenko Dmytro BI-242M NAU
```

Рис. 3.2. Створення повідомлення

3.3.1. Генерація ключей

Спочатку генеруємо дві пари ключей *RSA* 4096 біт для відправника та одержувача, що показано на рис 3.3.

```
System.out.println("1 - Generating RSA keys.".toUpperCase());

KeyPair senderKeyPairForEncryption = createKeyPair("SENDER ENCRYPTION");

KeyPair senderKeyPairForSigning = createKeyPair("SENDER SIGNING");

KeyPair receiverKeyPairForEncrypting = createKeyPair("RECEIVER ENCRYPTION");

KeyPair receiverKeyPairForSigning = createKeyPair("RECEIVER SIGNING");
```

Рис. 3.3. Генерація двох пар ключів для відправника та одержувача

3.3.2. Створення сеансового ключа

За допомогою стійкого генератора псевдовипадкових чисел створюємо сеансовий ключ, що показано на рис 3.4.

```
System.out.println("2 - Generating session key.".toUpperCase());
SecretKey secretKey = CryptoUtil.generateAESSecretKey();
String sessionKeyString = encoder.encodeToString(secretKey.getEncoded());
System.out.println("Session key: " + sessionKeyString + "\n");
```

```
2 - GENERATING SESSION KEY.
Session key: ybPRTt0dx1J+W6db5c/eF/6zJRM2PSz358q3fm5+UxM=
```

Рис. 3.4. Генерація сеансового ключа

3.3.3. Шифрування сеансового ключа за допомогою RSA

За допомогою *RSA*, відкритим ключем отримувача(*RECEIVER ENCRYPTION RSA PUBLIC KEY*) шифруємо сеансовий ключ, що показано на рис 3.5.

```
System.out.println("3 - Encrypting session key with RSA.".toUpperCase());
byte[] encryptedWithRSA = pgpService.encryptDataWithPublicKey(
receiverKeyPairForEncrypting.getPublic(), sessionKeyString.getBytes());
System.out.println(encoder.encodeToString(encryptedWithRSA) + "\n");
```

```
3 - ENCRYPTING SESSION KEY WITH RSA.
Uds77yc5Pz/lVSSrp9zIGsOvSZr4eZ+FThlIdTnCjEYllT9Ee2IBXKnLRRxtb7nh2hUo2n4K11zRXQt5+OpmhPY2h
nYWEOKF8fzdnAvwjlGxN+Laq5yvgL9zofz+NQ+kdDXc5gQXklBSHYlQ9p9hPAV/5X57o/nOT8/JdmGN4KtkEXF97B
GKFcQDaqC+DpuW+1mPXRjIGa9FPBFT05DPPTcIX6MYXa9Dv5P2PUYjs6NnPic0DIOCLl2uaIXcR1gkRyCxUDHFhTR
nZdgw6jOYIfJ6YLdkMlsi+OnWUdhpLrLBLBs7XJyWvDkz/mQMN7AD+YsRBa0h+1RztMHiF2qllA==
```

Рис. 3.5. Шифрування сеансового ключа

3.3.4. Підписання повідомлення

За допомогою *SHA*-3 створюється дайджест повідомлення який додається до початку повідомлення та шифрується відкритим ключем отримувача (*RECEIVER SIGNING RSA PUBLIC KEY*), що показано на рис 3.6.

System.out.println("4 - Creating digital signature for the message.".toUpperCase());

byte[] signature = pgpService.sign(receiverKeyPairForSigning.getPublic(), message.getBytes(StandardCharsets.UTF_8));

System.out.println("Signature created: " + encoder.encodeToString(signature) + "\n");

String digestWithData = encoder.encodeToString(signature) + "\n" + message;

System.out.println("5 - Adding digest to the top of data.".toUpperCase());

System.out.println("Data: " + digestWithData + "\n");

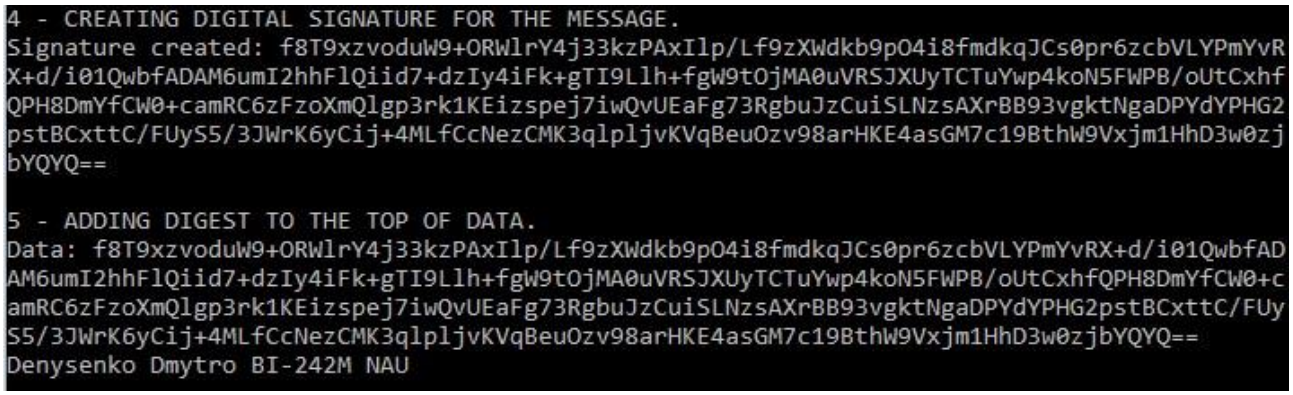

Рис. 3.6. Підписання повідомлення

3.3.5. Шифрування повідомлення

За допомогою *AES* сеансовим ключем шифруємо дані, що показано на рис 3.7.

System.out.println("6 - Encrypt message with session key using AES.".toUpperCase());

byte[] encryptedWithAES = pgpService.encryptDataWithSessionKey(

secretKey, digestWithData.getBytes(StandardCharsets.UTF_8));

System.out.println("Encrypted message with session key using AES: "+ encoder.encodeToString(encryptedWithAES) + "\n");

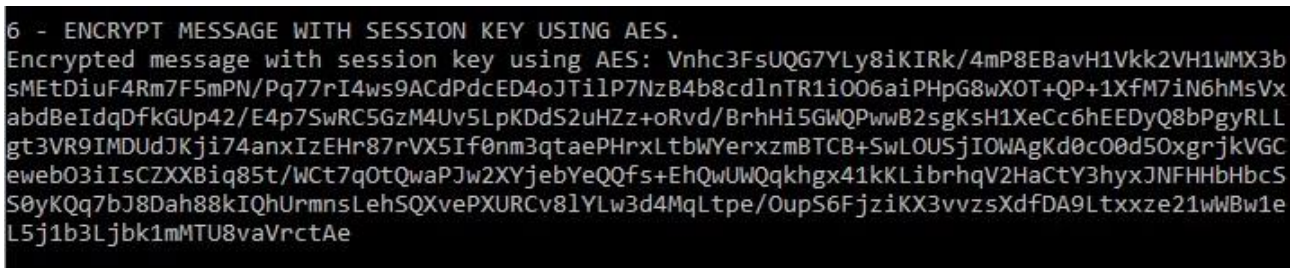

Рис. 3.7. Шифрування повідомлення за допомогою *AES*

3.3.6. Дешифрування сеансового ключа

Після того як отримувач отримав повідомлення, він розшифрує сеансовий ключ за допомогою особистого ключа (*RECEIVER ENCRYPTION PRIVATE KEY*), що показано на рис. 3.8.

```
System.out.println("7 - Decrypting session key with RSA.".toUpperCase());
byte[] decryptedWithRSA = pgpService.decryptDataWithPrivateKey(
receiverKeyPairForEncrypting.getPrivate(), encryptedWithRSA);
String sessionKey = new String(decryptedWithRSA, StandardCharsets.UTF_8);
System.out.println("Session key: " + sessionKey + "\n");
```

Рис. 3.8. Дешифрування сеансового ключа

3.3.7. Дешифрування повідомлення

За допомогою розшифрованого сеансового ключа проводиться дешифрування повідомлення, що показано на рис. 3.9.

```
SecretKey secretKeyDecrypted = CryptoUtil.getSecretKeyFromDecodeKey(decoder.decode(sessionKey));
System.out.println("8 - Decrypting data using session key.".toUpperCase());
```

byte[] decryptedMessage = pgpService.decryptDataWithSessionKey(secretKeyDecrypted, encryptedWithAES);

String decryptedData = new String(decryptedMessage);

System.out.println("Decrypted data: " + decryptedData + "\n");

```
8 - DECRYPTING DATA USING SESSION KEY.
Decrypted data: f8T9xzvoduW9+ORWlrY4j33kzPAxIlp/Lf9zXWdkb9pO4i8fmdkqJCs0pr6zcbVLYPmYvRX+d
/i01QwbfADAM6umI2hhFlQiid7+dzIy4iFk+gTI9Llh+fgW9tOjMA0uVRSJXUyTCTuYwp4koN5FWPB/oUtCxhfQPH
8DmYfCW0+camRC6zFzoXmQlgp3rk1KEizspej7iwQvUEaFg73RgbuJzCuiSLNzsAXrBB93vgktNgaDPYdYPHG2pst
BCxttC/FUyS5/3JWrK6yCij+4MLfCcNezCMK3qlpljvKVqBeuOzv98arHKE4asGM7c19BthW9Vxjm1HhD3w0zjbYQ
YQ==
Denysenko Dmytro BI-242M NAU
```

Рис. 3.9. Дешифрування повідомлення

3.3.8. Перевірка цифрового підпису

Дешифровуваши повідомлення, перевіряємо вірність цифрового підпису за допомогою особистого ключа отримувача (*RECEIVER SIGNING PRIVATE KEY)*, що показано на рис. 3.10.

System.out.println("9 - Verifying signature.".toUpperCase());

String digest = decryptedData.split("\n") [0];

String text = decryptedData.split("\n") [1];

boolean verified = pgpService.verify(receiverKeyPairForSigning.getPrivate(), text.getBytes(StandardCharsets.UTF_8), decoder.decode(digest));

System.out.println("Signature verified: " + verified + "\n");

```
9 - VERIFYING SIGNATURE.
Signature verified: true
```

Рис. 3.10. Перевірка правильності цифрового підпису

3.3.9. Результат роботи програми

Якщо, цифровий підпис співпадає, то виводимо отримане повідомлення на екран користувача, що показано на рис. 3.11.

if (verified) {

System.out.println("10 - Printing initial message.");

System.out.println("Decrypted message: " + text);}

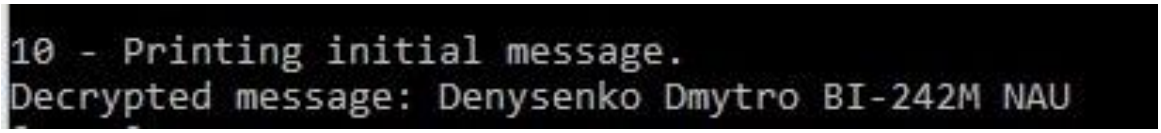

Рис. 3.11. Вивід отриманого повідомлення

3.4. Висновки до розділу 3

У розділі 3 було реалізовано та перевірено на практиці програмний модуль на основі методу *PGP*. Ключовою відмінністю від стандарту стало використання найбільш сучасних та стійких алгоритмів шифрування та хешування, а саме *RSA* із ключем довжиною 4096 біт, *AES* із ключем довжиною 256 біт, *SHA*-3 із ключем довжиною 512 біт. Також великою відмінністю стало використання двох ключових пар для кожного із учасників, а саме пара ключів для шифрування та пара ключів для підписання документа.

У порівняні із стандартною реалізацією *PGP*, з однією ключовою парою, швидкість шифрування не зазнала втрат, в той час як надійність захисту інформації виросла в декілька раз. Загальний час роботи програми для шифрування та шифрування зайняв 16 секунд, що є більш ніж гарним показником, дивлячись на рівень захищеності даних.

Отже, результат є досить задовільний та має право на використання, проте видно напрямки у яких треба роботи покращення, особливо, що торкається простоти використання.

ВИСНОВКИ

Захист інформації актуальне та сучасне питання, саме тому створення програмного модуля із сучасними алгоритмами шифрування і цифрового підпису мають велике значення.

Основні результати роботи полягають:

- Проаналізована законодавча та нормативно-правова база в сфері захисту інформації. Аналіз показав, що на теперішній час в Україні розроблена досить велика законодавча база щодо використання електронної пошти, основана на широкій правовій і нормативній базі та створена інфраструктура, що має забезпечити надійний захист інформації у державі. Проаналізовано алгоритм роботи методу PGP.
- Проаналізовано алгоритм роботи методу *PGP*.
- Досліджені можливі вразливості, загроза та атаки на метод *PGP*. Для доступу до даних необхідно зламати асиметричний чи симетричний ключ, що при коректному їх використанні виглядає неможливим. Проблемою залишається людський фактор, але це стосується будь-якої сфери життя.
- Проаналізовані алгоритми шифрування *RSA, AES* та алгоритм хешування *SHA*-3. Їх сучасні реалізації показують себе як надійний спосіб для захисту інформації
- За допомогою мови програмування Java створено програмний модуль на основі PGP, що має у свої меті створення можливості для захищеного спілкування повідомленнями.

Аналіз нормативно-правової бази України показав, що існує певний обсяг створених та прийнятих документів, проте проблема в тому, що багато із нормативних актів прийняті ще на початку сторіччя або ще раніше. Необхідно оновлювати та покращувати законодавчі акти, щоб вони відповідали сучасним

реаліям. Існуюча адміністративна та кримінальна відповідальність не покриває усі можливі загрози пов’язані із захистом інформації.

Метод PGP існує вже майже 30 років, проте до сих пір показує себе як надійний, та майже незламний. Випадки втрату даних пов’язані із використанням поштових клієнтів, а не являються проблемою самого методу.

Реалізований метод використовує найсучасніші та стійкі методи захисту інформації, а саме *RSA* із ключем довжиною 4096 біт, *AES* із ключем довжиною 256 біт, *SHA*-3 із ключем довжиною 512 біт.

Важливим вдосконаленням є використання замість однієї ключової пари – двох. Однієї для захисту цифрового підпису, іншої для захисту сеансового ключа, що має на меті підвищення захищеності даних у разі розкриття одного із ключів. Швидкість використання даного методу не погіршилась через використання цього нововведення.

Загалом програмний модуль можна використовувати для захищеного спілкування за допомогою електронної пошти чи інших додатків. Рекомендовано використовувати його як для приватного листування так і для корпоративного, що забезпечить високий рівень захищеності інформації.

Майбутній розвиток даної реалізації полягає у створенні зручного користувацького інтерфейсу, інтеграції цього методу у поштові клієнти та мессенджери. Також необхідно весь час спостерігати за розвитком криптоаналізу та створенням нових алгоритмів захисту інформації, щоб своєчасно покращувати програмне забезпечення і рівень захищеності інформації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Аграновский А.В. Практическая криптография: алгоритмы и их программирование / А.В. Аграновский, Р.А. Хади. – М.: СОЛОН-Пресс, 2009. – 256 с.: ил.
2. Блінцов В.С. Математичні основи криптології + CD: Навчальний посібник для студ. вищих навч. закл. / В.С. Блінцов, .Л. альчевський. – Миколаїв: Національний ун-т кораблебудування ім. адмірала Макарова, 2006. – 232 с.: іл.
3. Горбенко І.Д. Прикладна криптологія. Теорія. Практика. Застосування : монографія / І.Д. Горбенко, І.І. Горбенко. – Харків: Видавництво "Форт", 2012. – 880 с.: іл
4. Домарев В.В. "Безопасность информационных технологий. Методология создания систем защиты" – К: ООО "ТИД "ДС", 2002 – 688с.
5. Денисенко Д. Г. Дослідження переваг та недоліків OpenPGP та побудова схеми шифрування / Денисенко Д. Г. Карловський С. Є. Телющенко В. А. // Наукові обрії – 2020. – с. 56-58
6. Денисенко Д. Г. Метод захисту інформації під час використання електронної пошти / Динамиката на съвременната наука – 2020. - с.66-68
7. Задірака В.К. Комп'ютерна криптологія : Підручник / В.К. Задірака, О.С. Олексюк. – К.: Тернопільська академія народного господарства; НАН України; Інститут кібернетики ім. В.М. лушкова, 2002. – 504 с.: іл
8. Закон України "Про державну таємницю" // ВВР. – 1994. – № 16. – Ст. 93. Вводиться в дію Постановою ВР № 3856-XII від 21.01.94, ВВР, 1994, № 16, ст.94. Із змінами, внесеними згідно із Законами № 1169-VII від 27.03.2014, ВВР, 2014, № 20-21, ст.746 // № 1170-VII від 27.03.2014, ВВР, 2014, № 22, ст.816 // № 374-VIII від 12.05.2015, ВВР, 2015, № 28, ст.245 // № 576-VIII від 02.07.2015, ВВР, 2015, № 36, ст.360 // № 1798-VIII від

21.12.2016, ВВР, 2017, № 7-8, ст.50 // № 2509-VIII від 12.07.2018, ВВР, 2018, № 35, ст.267

9. Закон України "Про електронний цифровий підпис" // ВВР. – 2003. – № 36. – Ст. 276. Із змінами, внесеними згідно із Законами № 879-VI від 15.01.2009, ВВР, 2009, № 24, ст.296 // № 5284-VI від 18.09.2012, ВВР, 2013, № 37, ст.488 // № 222-VIII від 02.03.2015, ВВР, 2015, № 23, ст.158 //№ 1666-VIII від 06.10.2016, ВВР, 2016, № 47, ст.800
10. Закон України "Про електронні документи та електронний документообіг" //ВВР. – 2003. – № 36. – Ст. 275. Із змінами, внесеними згідно із Законами № 2599-IV від 31.05.2005, ВВР, 2005, № 26, ст.349 // № 1170-VII від 27.03.2014, ВВР, 2014, № 22, ст.816 // № 1206-VII від 15.04.2014, ВВР, 2014, № 24, ст.885 // № 675-VIII від 03.09.2015, ВВР, 2015, № 45, ст.410 //№ 2155-VIII від 05.10.2017, ВВР, 2017, № 45, ст.400
11. Закон "Про захист інформації в інформаційно-телекомунікаційних системах" // ВВР. – 1994. - №31 - ст.286. Із змінами, внесеними згідно із Законами N 879-VI (879-17) від 15.01.2009, ВВР, 2009, № 24, ст.296 // № 1180-VI (1180-17) від 19.03.2009, ВВР, 2009, № 32-33, ст.485 // № 767-VII (767-18) від 23.02.2014, ВВР, 2014, № 17, ст.593 // № 1170-VII (1170-18) від 27.03.2014, ВВР, 2014, № 22, ст.816.
12. Закон "Про захист персональних даних" // ВВР. – 2010. - № 34. - ст. 481. Із змінами, внесеними згідно із Законами № 4452-VI від 23.02.2012, ВВР, 2012, № 50, ст.564 // № 5491-VI від 20.11.2012, ВВР, 2013, № 51, ст.715 // № 245-VII від 16.05.2013, ВВР, 2014, № 12, ст.178 // № 383-VII від 03.07.2013, ВВР, 2014, № 14, ст.252 // № 1774-VIII від 06.12.2016, ВВР, 2017, № 2, ст.25 // № 2168-VIII від 19.10.2017, ВВР, 2018, № 5, ст.31.
13. Закон України "Про інформацію" // ВВР. – 1992. – № 48. – Ст. 650. Вводиться в дію Постановою ВР від 02.10.92 №2658-12 // ВВР. – 1992. – №48. – Ст. 651. Із змінами, внесеними згідно із Законами: № 1642-III від 06.04.2000, ВВР, 2000, № 27, ст.213 // № 3047-III від 07.02.2002, ВВР, 2002, № 29, ст.194 // № 2724-VI від 30.11.2010, ВВР, 2011, № 12, ст.86 //

№ 2756-VI від 02.12.2010, ВВР, 2011, № 23, ст.160 // № 317-VIII від 09.04.2015, ВВР, 2015, № 26, ст.219 // № 1405-VIII від 02.06.2016, ВВР, 2016, № 28, ст.533 // № 1774-VIII від 06.12.2016, ВВР, 2017, № 2, ст.25

14. Закон "Про Національну програму інформатизації" // ВВР. – 1998. - № 27-28. - ст.181. Із змінами, внесеними згідно із Законами № 2684-III від 13.09.2001, ВВР, 2002, № 1, ст.3 // № 2289-VI від 01.06.2010, ВВР, 2010, № 33, ст.471 // № 5463-VI від 16.10.2012, ВВР, 2014, № 4, ст.61 // № 922-VIII від 25.12.2015, ВВР, 2016, № 9, ст.89.
15. Іванов М.А. Криптографические методы защиты информации в компьютерных системах и сетях / М.А. Иванов – М.: КУДИЦОБРАЗ, 2001. – 363 с.: ил.
16. Кодекс України про адміністративні правопорушення: Кодекс України від 7 грудня 1998 р. № 8074-10 // Відомості Верховної Ради УРСР – 1984. – 18 грудня. – С. 330.
17. Конституція України. [Електронний ресурс] / Офіційний веб-сайт Верховної Ради України. - Режим доступу: http://portal.rada.gov.ua/.
18. Коутинхо С. Введение в теорию чисел. Алгоритм RSA / С. Коутинхо. – М.: Постмаркет, 2001. – 328 с.: ил.
19. Кримінальний кодекс України: Кодекс України від 5 квітня 2001 р. № 2341-III // Відомості Верховної Ради України – 2001. – 29 червня. – С. 447.
20. Молдовян Н.А. Криптография с открытым ключом / Н.А. Молдовян, А.А. Молдовян. – СПб.: БХВ-Петербург, 2005. – 288 с.: ил.
21. Петров А.А. Компьютерная безопасность. Криптографические методы защиты / А.А. Петров. – М.: Издательство ДМК, 2000. – 448 с.: ил.
22. Ричард Э. Смит Аутентификация: от паролей до открытых ключей. : Пер. с англ. – М. : Издательский дом ”Вильямс”, 2002. – 432 с. : ил. – Парал. тит. англ.
23. Сайт компанії *OpenPGP*. [Електронний ресурс] / - Режим доступу: https://www.openpgp.org/.

24. Соколов А.В., Степанюк О.М. Защита от компьютерного терроризма. Справочное пособие. – СПб.: БХВ – Петербург; Арлит 2002. – 496 с.
25. Стандарт криптографической защиты – AES. Конечные поля / А.С. Зензин, М.А. Иванов ; под ред. М.А. Иванова. – М.: КУДИЦОБРАЗ, 2002. – 176 с.: ил.
26. Столингс Вильям, Криптография и защита сетей: принципы и практика, 2-е издание: пер. с английского – М, : Издательский дом «Вильямс», 2001.
27. Указ президента України " Про Положення про порядок здійснення криптографічного захисту інформації в Україні" // 22.05.1998, Л. Кучма № 505/98. Із змінами, внесеними згідно з Указами Президента // № 1019/98 (1019/98) від 15.09.98 // № 1229/99 (1229/99) від 27.09.99 // № 333/2008 (333/2008) від 11.04.2008 // № 693/2009 (693/2009) від 28.08.2009.
28. Фороузан Б.А. Криптография и безопасность сетей: Учебное пособие / Б.А. Фороузан; пер. с англ. Под ред. А.Н. Берлина. – М.: Интернет-Университет Информационных Технологий: БИНОМ. Лаборатория знаний, 2010. – 784 с.: ил.
29. Хорошко В.А., Чекатков А.А. Методы и средства защиты информации К.: Юниор, 2003. – 504с.
30. Черемушкин А.В. Криптографические протоколы. Основные свойства и уязвимости: учеб. пособие для студ. учреждений высш. проф. образования / А.В. Черемушкин. – М.: Издательский центр "Академия", 2009. – 272 с.: ил.
31. Чугунков И.В., Иванов М.А. Криптографические методы защиты информации в компьютерных системах и сетях: Учебное пособие / Под ред. М.А. Иванова. М.: НИЯУ МИФИ, 2012. – 400 с.: ил.
32. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си / Б. Шнайер Б. – М.: Триумф, 2002. – 797 с.: ил.