НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені Ігоря СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра математичних методів захисту інформації

«До захисту допущено»
В.о. завідувача кафедри
_____ Сергій ЯКОВЛЄВ
«__» _____ 2022 р.

Дипломна робота

на здобуття ступеня бакалавра

зі спеціальності: 113 Прикладна математика
на тему: **«Методи підвищення ефективності криптографічних систем, які грунтуються на задачах факторизації»**

Виконав: студент 4 курсу, групи ФІ-84
Кислий Арсеній Олегович

Керівник: к.ф.-м.н., доцент Хмельницький М. О. _____

Консультант: _ _____

Рецензент: звання, степінь, посада Прізвище І.П. _____

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент _____

Київ — 2022

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені Ігоря СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра математичних методів захисту інформації

Рівень вищої освіти — перший (бакалаврський)
Спеціальність (освітня програма) — 113 Прикладна математика,
ОПП «Математичні методи криптографічного захисту інформації»

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ Сергій ЯКОВЛЄВ
«__» _____ 2022 р.

ЗАВДАННЯ
на дипломну роботу

Студент: Кислий Арсеній Олегович

1. Тема роботи: «*Методи підвищення ефективності криптографічних систем, які грунтуються на задачах факторизації*»,

керівник: к.ф.-м.н., доцент Хмельницький М. О.,

затверджені наказом по університету №__ від «__» _____ 2022 р.

2. Термін подання студентом роботи: «__» _____ 2022 р.

3. Вихідні дані до роботи: *опубліковані джерела за тематикою роботи*

4. Зміст роботи: *наведено математичний опис криптосистем, які ґрунтуються на задачі факторизації; проаналізовано розглянуті методи підвищення ефективності роботи криптосистем, а також вказано на переваги та недоліки кожного з методів.*

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо): *«Презентація доповіді»*

6. Дата видачі завдання: 10 вересня 2021 р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання	Примітка
1	Узгодження теми роботи із науковим керівником	01-15 вересня 2021 р.	Виконано
2	Огляд опублікованих джерел за тематикою дослідження	Вересень-жовтень 2021 р.	Виконано
3	Опрацювання теоретичних відомостей для роботи	Листопад 2021 р.	Виконано
4	Історичний огляд за темою дослідження	Грудень 2021 р.	Виконано
5	Математичний опис криптосистем, які ґрунтуються на задачі факторизації	Лютий 2022 р.	Виконано
6	Опис існуючих методів покращення роботи криптосистем	Квітень 2022 р.	Виконано
7	Проведення аналізу розглянутих методів	Квітень 2022 р.	Виконано
8	Виокремлення переваг та недоліків кожного з методів	Травень 2022 р.	Виконано
9	Оформлення роботи та підготовка презентації до захисту	Травень 2022 р.	Виконано

Студент _____ Кислий А.О.

Керівник _____ Хмельницький М. О.

РЕФЕРАТ

Кваліфікаційна робота містить: 59 сторінок, 1 рисунок, 2 таблиці, 22 джерела.

Метою дослідження є опис та аналіз існуючих методів підвищення ефективності алгоритмів, які ґрунтуються на задачі факторизації.

Об'єктом дослідження є інформаційні процеси в системах криптографічного захисту.

Предметом дослідження є методи підвищення ефективності криптографічних систем, які ґрунтуються на задачі факторизації.

У ході дослідження було розкрито в історичному контексті взаємозв'язок між іменами, ідеями та результатами, які були отримані різними науковцями. Також було наведено детальний математичний опис криптосистем, які ґрунтуються на задачі факторизації, а також детально описана робота методів підвищення їх ефективності.

Результатом роботи є проведений аналіз розглянутих методів підвищення ефективності описаних криптосистем. Було побудовано таблиці, які демонструють головну інформацію про кожен з методів, їх переваги та недоліки.

ЗАДАЧА ФАКТОРИЗАЦІЇ, МЕТОДИ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ КРИПТОСИСТЕМ, RSA, КРИПТОСИСТЕМА РАБІНА, КРИПТОСИСТЕМА ПАЙЄ, КРИПТОСИСТЕМА ОКАМОТО-УТІЯМИ

ABSTRACT

Qualification work contains: 59 pages, 1 figures, 2 tables and 22 sources.

The purpose of this work is to describe and analyze the existing methods of increasing efficiency of algorithms based on factorization problem.

The object of this research is information processes in cryptographic protection systems.

The subject of this work is methods of increasing efficiency of cryptographic systems based on factorization problem.

During the investigation it was revealed in the historical context the relationship between names, ideas and results obtained by different scientists. Also, a detailed mathematical description of cryptosystems based on factoring problems was given, as well as a detailed description of the work of methods to increase their efficiency.

The result of the work is the analysis of the considered methods of increasing the efficiency of the described cryptosystems. The tables were built that show the main information about each of the methods, their advantages and disadvantages.

FACTORIZATION PROBLEM, METHODS OF INCREASING EFFICIENCY OF CRYPTOGRAPHIC SYSTEMS, RSA, RABIN CRYPTOSYSTEM, PAILLIER CRYPTOSYSTEM, OKAMOTO-UCHIYAMA CRYPTOSYSTEM

ЗМІСТ

ВСТУП

Актуальність дослідження. Проблема захисту інформації шляхом її перетворення заради уникнення можливості зловмисника отримати доступ до неї, ще з давніх часів стала актальною. З плином часу криптографія застосовувалася в різних сферах, для вирішення тих чи інших проблем. І кожного разу список сфер застосування криптографії тільки розширювався.

На сьогодні, криптографія - це інструмент, який забезпечує конфіденційність, довіру, безпеку та багато інших важливих речей, які потребує сучасна людина.

Сучасні криптосистеми з кожним днем розвиваються та вдосконалюються, шляхом застосування різних методів, які б могли підвищити ефективність їх роботи. Та паралельно з розвитком інструментів, які здатні забезпечувати безпеку та захист інформації, розвиваються інструменти та методи для здійснення атак, які переслідують одну мету — отримати доступ до конфіденційної інформації. Тому дослідження та аналіз наявних методів підвищення ефективності криптографічних систем є актульною задачею на сьогодні. А результати аналізу можуть бути використані для забезпечення вирішення недоліків попередніх методів.

Метою дослідження є опис та аналіз існуючих методів підвищення ефективності алгоритмів, які ґрунтуються на задачі факторизації. Для досягнення мети треба вирішити такі завдання:

1) провести огляд опублікованих джерел за тематикою дослідження;

2) розглянути криптографічні системи, що ґрунтуються на задачі факторизації;

3) навести математичний опис розглянутих криптосистем;;

4) описати відомі методи підвищення ефективності цих криптосистем;

5) провести аналіз розглянутих методів та навести висновки.

Об'єктом дослідження є інформаційні процеси в системах криптографічного захисту.

Предметом дослідження є методи підвищення ефективності криптографічних систем, які ґрунтуються на задачі факторизації.

Практичне значення результатів полягає у тому, що вони можуть бути використані для подальшого дослідження методів підвищення ефективності алгоритмів, які ґрунтуються на задачі факторизації, а також їх вдосконалення.

1 ІСТОРИЧНИЙ ОГЛЯД ЗА ЗАДАНОЮ ТЕМОЮ ДОСЛІДЖЕННЯ

У даному розділі проведено огляд історичних подій та ідей, які призвели до появи криптосистем з відкритим ключем, їх вдосконалення та розвитку. Розглянуто ситуацію з цими криптосистемами на теперішній момент. А також описані варіанти їх застосування та актуальність на сьогодні.

1.1 Праця Рівеста, Шаміра та Адельмана

У листопаді 1976 Уітфілд Діффі і Мартін Хеллман опублікували статтю під назвою «Нові напрямки в криптографії» (англ. New Directions in Cryptography)[1]. Ця стаття повністю перевернула уявлення про криптографічні системи, заклавши основи криптографії з відкритим ключем.

Невдовзі було розроблено алгоритм Діффі – Хеллмана, який дозволяє двом учасникам виробити спільний секретний ключ, користуючись незахищеним каналом зв'язку. Але цей алгоритм не може вирішити проблему аутентифікації.

Праця Діффі та Хеллмана привернула увагу трьох вчених з Массачусетського інституту – Рональда Рівеста, Аді Шаміра та Леонарда Адельмана. Вивчивши статтю, науковці приступили до пошуку математичної функції, яка б дала можливість для реалізації моделі криптографічної системи з відкритим ключем, сформульовану Діффі та Хеллманом. На пошуки знадобилося витратити чимало часу та зусиль, проте розглянувши та опрацювавши більш ніж 40 варіантів, вченим все ж вдалося знайти алгоритм, який ґрунтувався на розходженні в тому, що знаходити великі прості числа легко, а розкладати на множники добуток

двох великих простих чисел – складно. Згодом алгоритм отримав назву RSA, за першими літерами прізвищ його творців. І у 1978 році було опубліковано опис нової криптосистеми під назвою RSA у журналі «Communications of the ACM»[2].

Алгоритм RSA став першим алгоритмом, який був би придатним для:

1) Шифрування;
2) Цифрового підпису.

Наразі RSA застосовується для багатьох криптографічних застосунків, включаючи: PGP, S/MIME, TLS/SSL, IPSEC/IKE та багато інших.

Не дивлячись на те, що алгоритм RSA набув чималої популярності та застосовується для багатьох криптографічних цілей, варто зазначити, що криптосистеми з відкритим ключем є суттєво повільнішими за симетричні криптосистеми, тому, для того аби збільшити та покращити ефективність застосування алгоритму RSA, його часто використовують для створення гібридних криптосистем, що в свою чергу дає можливість більш раціонально використовувати переваги криптосистеми.

Означення 1.1. Гібридна (комбінована) криптосистема — це така система шифрування, яка поєднує переваги криптосистем з відкритим ключем з продуктивністю симетричних криптосистем.

Тобто, повідомлення зашифровують за допомогою симетричних алгоритмів, бо в них шифрування відбувається з високою швидкістю і разом з тим короткі ключі, а RSA алгоритм використовують для шифрування самого симетричного ключа. Проте даний механізм потребує використання криптографічно стійкого генератора псевдовипадкових чисел для формування випадкового ключа симетричного шифрування.

Так як з плином часу, для взлому криптосистеми створювали та вдосконалювали різного роду атаки, то звичайним є той факт, що впродовж всього часу і навіть сьогодні створювалися різні методи покращення роботи алгоритму RSA.

Вчені всього світу аналізували роботу криптосистеми, для того аби покращити різні аспекти цього алгоритму. Так наприклад, проаналізувавши швидкість роботи процедури розшифрування, науковці Дж.Дж. Квісквотер та К.Коуврер[3] запропонували метод пришвидшення розшифрування, шляхом використання Китайської теореми про лишки. Або для підвищення швидкості генерування цифрового підпису, було описано метод RebalancedRSA[4], який значно покращував швидкість роботи процедур розшифрування та генерації цифрового підпису.

Інші методи покращення роботи алгоритму RSA будуть наведені далі у роботі.

1.2 Роботи Рабіна та Вільямса

Криптосистема Міхаеля Рабіна

Так чи інакше, поява криптосистеми RSA мала великий вплив для розвитку асиметричної криптографії в цілому. Багато науковців досліджували цей алгоритм під час проведення власних досліджень. Деякі вчені, намагалися вдосконалити роботу алгоритму RSA, але згодом їх розробки почали вважати окремими криптосистемами, до яких також почали створювати різні методи підвищення ефективності.

Так у 1979 році, тобто через рік після публікації опису RSA, Міхаел Рабін[5] вперше представив свою криптосистему.

Спочатку криптографічна система Рабіна створювалася саме як модифікація RSA, тому не дивно чому ці дві криптосистеми так схожі між собою, як у власне описі самих криптосистем, так і у недоліках які можуть виникати. Проте подібність цих двох систем не означає, що

обидві набули однакового успіху.

Криптосистема Рабіна стала першою криптосистемою з відкритим клчем, для якої було доведено, що відновити початковий текст з шифротексту так само важко, як і виконати процедури факторизації великих чисел. Точніше кажучи, складність відновлення пов'язана зі складністю вилучення квадратного корення за модулем складеного числа. Задача факторизації і задача вилучення квадратного корення еквівалентні.

Алгоритм шифрування в криптосистемі Рабіна надзвичайно ефективний і придатний для багатьох практичних застосунків, наприклад для шифрування за допомогою портативних пристроїв.

Основною перевагою криптосистеми Рабіна є те, що випадкове повідомлення можна точно відновити від зашифрованого лише якщо користувач, що розшифровує повідомлення спроможний до ефективної факторизації модуля.

Проте слід також зазначити, що криптосистема має певну низку недоліків, серед яких можна виділити такі:

1) обмеження до вхідних параметрів, що повинні задовольняти певній рівності;

2) в класичному алгоритмі Рабіна застосовується позиційна система числення, що призводить до зменшення швидкодії процесів шифрування та розшифрування, а також до збільшення часової складності при використанні багаторозрядних чисел.

Враховуючі усі недоліки можна сказати, що задача пошуку шляхів покращення та підвищення ефективності криптосистеми є актуальною і в наш час. Саме тому з плином часу продовжують з'являтися нові модифікації даного алгоритму.

Криптосистема запропонована Хью Вільямсом

Праця Міхаела Рабіна в свою чергу надихнула канадського

математика Хью Вільямса розробити власний метод вдосконалення роботи криптосистеми Рабіна. В різних джерелах по різному трактують працю Вільямса[6]: одні книжки описують її саме як метод, що здатний вирішити проблеми системи Рабіна, інші ж вказують її як самостійну криптосистему.

Метод Вільямса пропонує вирішення проблеми криптосистеми Рабіна шляхом застосування певних перетворень до повідомлення, а також пропонує робити додаткові обчислення символу Якобі. Саме за допомгою використання символу Якобі, Вільямс вирішує проблему Рабіна.

В свої основі метод канадського вченого спирається на задачу факторизації, так само як і RSA та алгоритм Рабіна. І хоча метод вирішує певні проблеми свого попередника, та все ж має власні значні недоілки, які також потребують вирішення.

Інші методи, що були розроблені для криптосистеми Рабіна, які в свою чергу вирішують ті проблеми, які виникли у криптосистемі Вільямса, будуть наведені та детально описані далі у роботі.

1.3 Послідовності Люка, як основа для нової криптосистеми

Криптосистема LUC – це криптосистема, яку описали у 1993 році дослідники Пітер Сміт та Майкл Леннон[7], так само як і криптосистема RSA, криптосистема LUC застосовується як для шифрування, так і для цифрового підпису. Характерною відмінністю даної криптосистеми є використання послідовностей Люка замість операції піднесення до степеню.

Означення 1.2. Послідовності Люка – це множина пар лінійних послідовностей другого порядку, які вперше були розглянуті французьким математиком Едуардом Люка. Серед відомих послідовностей Люка, які носять власні імена, виділяються числа

Фібоначчі, числа Люка і тд.

Сміт та Леннон провели детальний аналіз алгоритму RSA і вказали на суттєвий недолік криптосистеми, а саме на підробку підпису, за умови, що конкретні повідомлення вибрані зловмисником заздалегідь, підписуються жертвою.

Згодом науковці вирішили застосувати послідовності Люка замість піднесення до степеню. І хоча обчислювальні вимоги нової системи трохи більші ніж для алгоритму RSA, та дослідники змогли довести, що їх система криптографічно більш стійка, ніж криптосистема RSA.

Математики у своїй роботі описали нову криптосистему і довели, що виявлена ними проблема підробки підпису у системі RSA не стосується криптосистеми LUC, а тому нова криптосистема є криптографічно більш стійкою у порівнянні з алгоритмом RSA.

1.4 Гомоморфні криптографічні системи

У цьому підрозділі коротко описані відомості про криптосистему Окамото-Утіями та криптосистему Пайє, які мають гомоморфні властивості. Більш детально про гомоморфне шифрування буде вказано у другому розділі цієї роботи.

Робота японських вчених Окамото та Утіями

Криптосистема Окамото-Утіями, була запропонована у 1998 році Тацуакі Окамото та Сигенорі Утіямой[8]. Це імовірнісна криптосистема, побудована на основі логарифмічної функції L, визначеної над мультиплікативною групою $(Z/nZ)^*$, де $n = p^2q$, а p та q є великими простими числами.

В криптосистемі Окамото-Утіями безпека інформації, що передається по загальнодоступному каналу, підсилюється за допомогою семантичної безпеки через використання імовірнісної випадкової

величини, яка не дозволяє третій стороні визначити, які зашифровані дані були отримані з певного відкритого тексту.

Окамото-Утіями – це імовірнісна, частково гомоморфна криптосистема.

Означення 1.3. Імовірнісне шифрування полягає в тому, один і той самий початковий відкритий текст, перетворений на одному і тому ж ключі, призводить до появи безлічі різних шифрованих текстів.

Означення 1.4. Гомоморфне шифрування – це тип шифрування, який передбачає, що змінюючи зашифроване повідомлення певними алгебраїчними діями, буде змінюватися і відповідний відкритий текст, до того ж, не обов'язково знати сам відкритий текст.

Означення 1.5. Частково гомоморфна система – це така криптосистема, яка має властивість гомоморфності лише відносно функції додавання, або функції множення.

Криптосистема Окамото-Утіями може використовуватися для захисту системи електронного голосування для забезпечення безпеки та конфіденційності даних. А за допомогою гомоморфних властивостей алгоритму можна виконувати підрахунок голосів. Алгоритм може захистити інформацію, шляхом шифрування голосів виборців і в цей час система може робити підрахунки використовуючи зашифровану інформацію, без необхідності її розшифрування.

Праця науковця Паскаля Пайє

Французький криптограф Паскаль Пайє у 1999 році[9] вперше запропонував створений ним алгоритм. Криптосистема Пайє – це імовірнісна криптосистема, яка так само як і згадані раніше криптосистеми базується на складності розкладу складеного числа у добуток простих чисел.

Дана криптосистема є адитивною частково гомоморфною системою, тобто на основі знань тільки відкритого ключа і шифротекстів, що відповідають відкритим текстам m_1 та m_2, можна обчислити шифротекст відкритого тексту $m_1 + m_2$.

В своїй роботі Пайє описав припущення про складність обчислення кореня залишку від ділення по модулю і запропонував відповідний механізм використання цієї математичної проблеми в криптографічних цілях.

Криптосистема Пайє має велику кількість шляхів застосування, наприклад, її можна застосовувати для:

1) реалізації безпечних обчислень;
2) для збереження конфіденційності;
3) для безпечного електронного голосування;
4) для електронних транзакцій.

Однією з відмінних особливостей криптосистеми Пайє є властивість самоосліплення, під цим терміном розуміють здатність внесення змін до шифрованого тексту без змінення відкритого тексту. Це в свою чергу може бути використано при розробці систем електронної валюти.

Висновки до розділу 1

В даному розділі було коротко наведено загальну інформації про криптосистеми, які спираються на задачу факторизації, вказано на певні недоліки описаних криптосистем, що обумовлює актуальність вдосконалення цих криптосистем, шляхом розробки різніх методів, які будуть здатні підвищити ефективність роботи цих систем.

Також в загальному було описано дві криптосистеми з гомоморфними властивостями, наведені їх переваги та способи застосування.

2 МАТЕМАТИЧНИЙ ОПИС КРИПТОГРАФІЧНИХ СИСТЕМ

У другому розділі даної роботи наведено перелік необхідних означень, які будуть потрібними для кращого розуміння матеріалу. Проведено детальний математичний опис криптосистем з відкритим ключем на задачі факторизації, які були згадані у попередньому розділі, описано основні алгоритми: генерування ключів, шифрування та дешифрування. Також наведно перелік переваг та недоліків кожної з криптосистем, а також більш детально описано основні принципи гомоморфного шифрування та шляхів його застосування.

2.1 Перелік необхідних означень

Означення 2.1. Число називається **простим**, якщо воно ділиться лише на одиницю та само на себе.

Означення 2.2. Задача факторизації цілого числа – розкладання заданого числа на прості множники, розклад оператора на добуток декількох операторів нижчого порядку.

Задача факторизації імовірно є складною задачею.

Означення 2.3. Функція Ойлера – це відображення $\varphi : N \to N$, яке кожному числу $m \in N$ ставить у відповідність число $\varphi(m)$, що дорівнює кількості натуральних чисел $1 \leqslant a \leqslant m$, які є взаємно простими з m.

Означення 2.4. НСД (gcd) – найбільший спільний дільник двох або більше невід'ємних чисел, тобто це найбільше натуральне число, на яке діляться без остачі дані числа.

Означення 2.5. Китайська теорема про лишки — один з основних результатів елементарної теорії чисел. Використовуючи позначення

модульної арифметики її можна сформулювати таким чином. Нехай n_1,$n_2 \in N$; $(n_1,n_2) = 1$; x_1,$x_2 \in Z$. Тоді $\exists x \in Z$:

$$\begin{cases} x \equiv x_1 \bmod n_1 \\ x \equiv x_2 \bmod n_2 \end{cases}$$

Якщо x – деякий розв'язок даної системи порівнянь, то всі інші розв'язки даної системи будуть мати вигляд:

$$x + kn_1n_2,\ k \in Z$$

Зокрема, $\exists! x \in Z_{n_1 \cdot n_2}$, що є розв'язком системи порівнянь.

Означення 2.6. Число Блюма - це число виду $n = pq$, де числа p та q – різні прості числа виду $4k + 3$

Означення 2.7. Дужка Айверсона – це функція, яка повертає 1 якщо твердження істинне, і 0 якщо твердження є хибним.

Означення 2.8. Символ Лежандра $\frac{x}{p}$ для простого числа $p > 2$ і елемента $x \in Z$ називається величина, що приймає наступні значення:

$$(\frac{x}{p}) = \begin{cases} 0, & \text{якщо } x \text{ ділиться на } p; \\ 1, & \text{якщо } x \text{ є квадратичним лишком за модулем } p; \\ -1, & \text{якщо } x \text{ є квадратичним нелишком за модулем } p. \end{cases}$$

Означення 2.9. Символ Якобі – це узагальнення символу Лежандра, величина $(\frac{x}{n})$, що дорівнює $(\frac{x}{n}) = (\frac{x}{p_1})^{\alpha_1}(\frac{x}{p_2})^{\alpha_2}...(\frac{x}{p_s})^{\alpha_s}$, де $n = p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdot ...p_s^{\alpha_s}$, $(\frac{x}{p_i})$ – символ Лежандра

Означення 2.10. Число Кармайкла – це додатне складене число n, яке задовольняє умові $b^{n-1} \equiv 1 \pmod n$ для усіх цілих чисел b, що є взаємно простими з n.

2.2 Криптосистема RSA

Опис алгоритму

Криптографічні системи з відкритим ключем використовують односторонні функції, котрі мають такі властивості:

1) якщо x відомий, то $f(x)$ знайти відносно легко;
2) якщо відомий $y = f(x)$, то аби знайти x немає простого шляху.

В основі RSA[2] лежить складність задачі факторизації добутку двох великих простих чисел. Для того аби зашифрувати інформацію використовується операція піднесення до степені за модулем великого числа. Для розшифрування необхідно обрахувати функцію Ойлера від заданого великого числа, для того щоб її обрахувати необхідно знати розклад числа на множники.

У криптосистемах з відкритим ключем кожен учасник має відкритий та секретний ключ, який він створив самостійно. В системі RSA кожен ключ — це пара цілих чисел. Учасники обміну повідомленнями можуть повідомляти відкриті ключі кому завгодно, на відміну від секретних ключів, які вони мають тримати в повній таємниці і нікому їх не повідомляти, так як якщо повідомлення було зашифроване відкритим ключем, то розшифрувати його можна лише відповідним секретним ключем.

Криптографічна система RSA складається з 3 алгоритмів:

1) Алгоритм генерації ключів (відкритого і секретного);
2) Алгоритм шифрування;
3) Алгоритм розшифрування;

Алгоритм створення відкритого і секретного ключів

Шифрування у криптосистемі RSA починається з генерування відкритого та секретного ключів. Алгоритм генерування ключів виглядає так:

1) Довільно обираються два різних великих простих числа p, q, (тобто $p \neq q$);

2) Знаходиться їх добуток $N = p \cdot q$, цей добуток називається модулем;

3) Обчислюється значення функції Ойлера: $\varphi(N) = (p-1) \cdot (q-1)$, де n – не секретне; p,q,$\varphi(N)$ – секрет;

4) Обирається просте число e, таке що $1 < e < \varphi(N)$, e та $\varphi(N)$ – взаємно прості;

5) Обчислюється значення числа d, яке називається секретною експонентою:

$$d \cdot e = 1 \bmod \varphi(N) \implies d = e^{-1} \bmod \varphi(N)$$

значення числа d знаходиться за допомогою розширеного алгоритму Евкліда;

6) Пара (e, N) – називається відкритим ключем;

7) Пара (d, N) – називається секретним ключем.

Алгоритм шифрування повідомлення

Передача повідомлення M між користувачем А та користувачем Б в алгоритмі RSA відбувається наступним чином:

Вважається, що користувач Б хоче відправити повідомлення M до користувача А, вважають, що повідомлення M є цілим числом, таким що: $0 \leqslant M \leqslant N-1$ і $\gcd(M, N) = 1$

Алгоритм:

1) Застосовується відкритий ключ (e, n) абонента А;

2) Обирається відкритий текст M;

3) Шифрують повідомлення ключем абонента А за допомогою операції піднесення до степеню e по модулю N:

$$C = E(M) = M^e \bmod n$$

Алгоритм розшифрування повідомлення

Для того аби розшифрувати повідомлення M, абоненту А необхідно зробити наступні кроки:

1) Прийняти зашифроване повідомлення C;

2) Взяти власний секретний ключ (d,N);

3) Розшифрувати повідомлення секретним ключем, згідно з теоремою Ойлера:

$$M = D(C) = C^d \bmod n$$

Для кращого розуміння, на рисунку 2.1. можна побачити повну схему роботи алгоритму.

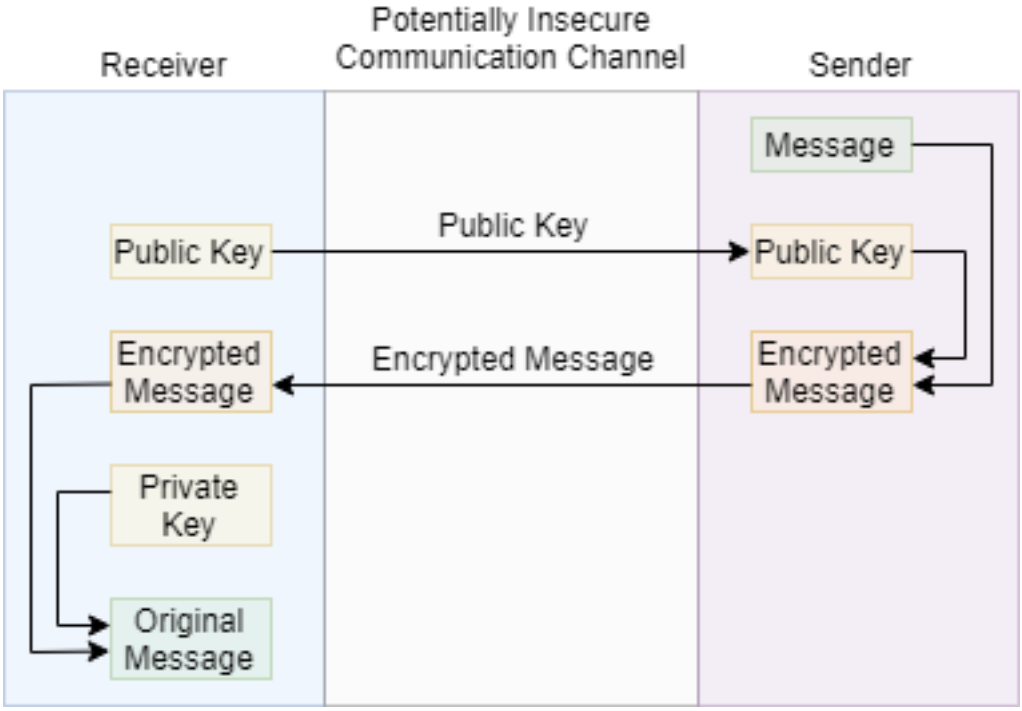

Рисунок 2.1 – Схема роботи криптосистеми RSA

Цифровий підпис

Як згадувалося в першому розділі, криптосистему RSA можна використовувати не лише для шифрування повідомлень, але й для цифрового підпису.

Припускають, що абонент А хоче відправити до абонента Б повідомлення, яке буде затверджене електронним цифровим підписом, для цього абонент А виконує наступні кроки:

1) Обирає відкритий текст M;

2) Створює цифровий підпис використовуючи власний секретний ключ

$$S = M^d \bmod N$$

3) Передає пару (S,M).

Для перевірки підпису S абоненту Б необхідно:

1) Взяти відкритий ключ абонента А;

2) Обчислити:

$$M = S^e \bmod N$$

3) Перевірити правильність підпису.

Безпека та оцінки складності обчислень у алгоритмі RSA

Як вже було згадано раніше, складність атаки на алгоритм RSA, ґрунтується на труднощі вирішення проблеми факторизації числа n. Досі не знайдено ефективного методу вирішення проблеми факторизації, саме це і забезпечує статус надійності RSA, однак варто пам'ятати, що твердження про залежність криптосистеми від задачі факторизації є гіпотетичним, тому що теоретично можливе існування алгоритму факторизації з поліноміальною складністю на звичайному комп'ютері.

Прийнятний рівень безпеки у протоколі RSA досягається при використанні відносно невеликого числа n. Актуальна довжина числа рівна 2048 біт, чи більше.

Оцінки складності обчислень:

Для довільних x та k складність обчислення функції RSA, тобто $y = x^e \bmod N$ за схемою Горнера оцінюється в кількості операцій множення за модулем N нерівністю: $L_1 \leqslant 2log_2N$

При невідомих значеннях p та q складність обернення функції RSA в загальному випадку записують так: $L_2 = O(\sqrt{N})$ – для простого алгоритму і $L_3 = exp\{(c_0 + o(1))ln^{1/3}N \cdot (lnln(N))^{2/3}\}$ – для кращого субекспоненціального алгоритму.

При відомих параметрах p та q складність обчислення оберненої функції оцінується такою нерівністю : $L_4 \leqslant 2log_2N$, тобто якщо знати параметри p та q, можна досить швидко обчислити і функцію RSA, і обернену функцію.

Недоліки криптосистеми

RSA, так само як і всі інші криптосистеми з відкритим ключем, має певні недоліки, серед яких:

1) Швидкість роботи. У порівнянні з симетричним шифрування, асиметричне помітно поступається у швидкості, адже у асиметричних криптосистемах використовуються ресурсоємні операції (піднесення одного великого числа у степінь іншого великого числа);

2) Необхідність захисту відкритих ключів від підміни;

3) Досі немає математичного доведення необоротності функції, яка використовується у криптосистемі.

Переважна більшість розроблених методів покращення роботи RSA спрямовані на вирішення саме цих недоліків. Детально ці методи будуть розглянуті далі у роботі.

2.3 Криптосистема Рабіна

Криптографічна система Рабіна[5] була запропонована, як перша асиметрична криптосистема із доказовою стійкістю. Дана система базується на використанні односторонньої функції Рабіна, яка має вигляд $f(x) = x^2 \bmod n$, де число N є добутком простих чисел p та q, обертання даної функції еквівалентно факторизації числа N. Криптосистема запропонована Міхаелем Рабіним, так само як і RSA, придатна як для шифрування, так і для цифрового підпису, проте варто зазначити, що схема Рабіна потребує проведення спеціальної процедури форматування вхідних повідомлень.

Як і будь-яка асиметрична криптосистема, система Рабіна використовує відкритий та секретний ключі. Відкритий використовується для шифрування повідомлень, може бути доступний для всіх, а секретний необхідний для розшифрування повідомлення, має триматися у повному секреті.

Алгоритм генерування ключів

1) Користувач А генерує два великих простих числа p та q, кожне число має бути виду $4k + 3$, водночас $p \neq q$;

2) А обчислює $N = p \cdot q$.

Число N – відкритий ключ користувача А, а пара чисел p та q – секретний ключ корисутвача.

Форматування відкритих повідомлень

Повідомлення, шифротексти та цифрові підписи у криптосистемі Рабіна є елементами кільця Z_n, для того аби у схемі Рабіна виконувалося обмеження яке висувається до відкритих повідомлень M, а саме $M > \sqrt{N}$, використовують форматування вхідних повідомлень.

$x = form(M,r)$ де r це якесь випадкове значення, процедура форматування повідомлень має гарантувати, що $x \in Z_n$ і що $x \geqslant \sqrt{N}$.

Приклад функції форматування:

$$x = 2^{l-2} + r \cdot 2^{l-65} + M$$

Де l – це довжина числа N у бітах, множина можливих вхідних повідомлень обмежується числами, довжина яких до $l - 66$ включно, а довжина параметру r встановлюється у 64 біти.

Алгоритм шифрування

Нехай N – відкритий ключ користувача А, x – відформатоване повідомлення яке треба зашифрувати. Аби зашифрувати повідомлення користувачу Б треба виконати наступні пункти:

1) Обчислити шифротекст:

$$y = x^2 \bmod N$$

2) Обчислити значення додаткових біт інформації:

$$c_1 = x \bmod 2, c_2 = [(\tfrac{x}{N} = 1]$$

квадратні дужки позначають символ Айверсона, а $(\frac{x}{N})$ – символ Якобі. Тобто $c_2 = 1$, якщо $(\frac{x}{N}) = 1$ і $c_2 = 0$, якщо $(\frac{x}{N}) = -1$. Трійка (y,c_1,c_2) формує зашифроване повідомлення.

3) Відправити отримане значення користувачу А.

Алгоритм розшифрування

Після того, як користувач А отримав шифртекст (y,c_1,c_2), йому необхідно обчислити 4 квадратні корені:

$$\sqrt{y} \bmod N$$

Коректним відкритим текстом x буде той корінь, для якого співпадуть обидва додаткові біти шифртексту c_1 та c_2.

Щоб отримати правильний відкритий текст M користувачу А треба видалити байти форматування.

Схема цифрового підпису

Криптосистему Рабіна, так само як і криптосистему RSA, можна використовувати для постановки цифрового підпису. Далі буде наведено відповідну процедуру.

Отже у системі Рабіна процедура постановки ЕЦП виглядає наступним чином:

1) Користувач А проводить процедуру форматування відкритого тексту M у повідомлення x;

2) Далі проводиться перевірка чи виконується рівність:

$$(\tfrac{x}{p}) = (\tfrac{x}{q}) = 1$$

3) Якщо рівність не виконується – відбувається повернення до пункту 1) і виконується повторне форматування з іншим значенням r;

4) Далі користувач А обчислює чотири квадратні корені:

$$\sqrt{x} \bmod N$$

і випадковим чином обирає один з них, як цифровий підпис S під повідомленням N.

Для того щоб, користувач Б перевірив підписане повідомлення (M,S), йому необхідно лише обчислити значення:

$$x = S^2 \bmod N$$

та перевірити чи є x відформатованим повідомленням M.

Загальна оцінка алгоритму

Роблячи висновки про криптосистему Рабіна, можна сказати, що у порівнянні з криптосистемою RSA, вона здатна демонструвати значне збільшення ефективності роботи. Проте разом з тим, вона має власні недоліки.

Серед переваг криптосистеми можна виділити такі:

1) Швидкість;

У порівнянні з алгоритмом RSA, криптосистема має значну перевагу у швидкості виконання алгоритму, через те що відбувається піднесення до квадрату при шифруванні повідомлення, замість зведення в степінь e.

2) Відновлення вхідного повідомлення;

У криптосистемі Рабіна, початкове повідомлення може бути відновлене від шифротексту лише за умови, що розшифрувальник спроможний на ефективну факторизацію відкритого ключа користувача.

3) Доказова стійкість криптосистеми.

Проте не дивлячись на всі переваги, система Рабіна не змогла здобути такого ж широкого визнання та застосування, як криптосистема RSA саме через свої недоліки, серед яких виділяються найголовніші:

1) Чотири результати розшифрування;

Певно це є головним недоліком криптосистеми, адже разом з правильним результатом розшифрування, користувач отримує ще 3 неправильних, це потребує додаткових дій та ресурсів.

2) Вразливість перед атакою на основі обраного шифрованого тексту;

3) Необхідність форматування вхідних повідомлень.

Але так само як і для її попередника, для криптосистеми Рабіна було розроблено чимало різних методів, які в першу чергу намагалися вирішити проблему вибору результату розшифрування.

2.4 Криптосистема LUC

Перш ніж почати опис криптосистеми LUC[7], доцільним буде більш детально розповісти про власне те, на чому ґрунтується дана криптосистема, а саме – послідовності Люка.

Важливо не плутати послідовності Люка з числами Люка. Послідовності Люка являють собою пари послідовностей (U_n) та (V_n). Прийнято використовувати послідовність $V_n(P,Q)$, але всі властивості будуть справедливими і для послідовності $U_n(P,Q)$.

Нехай a та b – ненульові цілі числа, α та β – корені наступного рівняння:

$$x^2 - ax + b = 0$$

Тоді маємо наступні результати:

$$\alpha = \tfrac{a+\sqrt{D}}{2} \text{ і } \beta = \tfrac{a-\sqrt{D}}{2}, \text{ де } D = a^2 - 4b$$

Послідовності Люка U_n та V_n записуються як:

$$U_n = \tfrac{\alpha^n - \beta^n}{\alpha - \beta}$$
$$V_n = \alpha^n + \beta^n$$

Прийнято вважати, що $U_0 = 0$, $U_1 = 1$, $V_0 = 2$, $V_1 = a$, а для $n \geqslant 2$ маємо наступне:

$$U_n = aU_{n-1} - bU_{n-2}$$
$$V_n = aV_{n-1} - bV_{n-2}$$

Для подальшого опису криптосистеми потрібно записати наступне твердження:

$$V_n(V_k(a,b),b^k) = V_{nk}(a,b)$$

Запишемо для $b = 1$:

$$V_n(V_k(a,1),1) = V_{nk}(a,1)$$

Для більш детального опису властивостей послідовностей Люка, які застосовуються в даній криптосистемі можна подивитися джерело[7].

Алгоритм генерації ключів

Для генерації відкритого та секретного ключа в криптосистемі LUC необхідно виконати наступні пункти:

1) Обрати два великих простих числа p та q, такі що $p \neq q$;

2) Обчислити добуток чисел p та q: $N = p \cdot q$;

3) Обрати випадкове число e, таке що:

$$\gcd(e,(p-1)\cdot(q-1)\cdot(p+1)\cdot(q+1)) = 1$$

4) Чотири можливих ключа розшифрування:

$$d = e^{-1} \bmod (\mathrm{lcm}((p+1),(q+1)))$$
$$d = e^{-1} \bmod (\mathrm{lcm}((p+1),(q-1)))$$
$$d = e^{-1} \bmod (\mathrm{lcm}((p-1),(q+1)))$$
$$d = e^{-1} \bmod (\mathrm{lcm}((p-1),(q-1)))$$

тут lcm – це найменше спільне кратне.

Таким чином: відкритим ключем в даному алгоритмі є числа e та N, а закритим – d та N.

Алгоритм шифрування повідомлення

Нехай e та N – відкритий ключ користувача А, M – повідомлення яке треба зашифрувати. Аби зашифрувати повідомлення користувачу Б потрібно виконати наступне:

1) Обчислити значення:

$$C = V_e(M,1) \bmod N$$

2) Відправити отримане значення користувачу А.

Алгоритм розшифрування повідомлення

Після отримання користувачем А, шифртексту C відправленого користувачем Б, користувачу А треба розшифрувати повідомлення використовуючи формулу:

$$M = V_d(C,1) \bmod N$$

Складнощі у використанні LUC криптосистеми

Під час роботи з даною криптосистемою виникають деякі складнощі, а саме:

1) Обчислення великих чисел Люка, стає достатньо складною задачею, через те, що числа Люка задаються рекурентно, а отже доводиться перебирати всі попередні числа.

Однак, за допомогою наступних двох властивостей, можна вирішити дану проблему.

$$1)\ V_{2n} = V_n^2 - 2b^n$$
$$2)\ 2V_{n+m} = V_nV_m + DU_nU_m$$

За допомогою цих властивостей, можна достатньо швидко отримати потрібне число, шляхом розкладу номеру елемента послідовності за степенями двійки.

2) Варіанти значень секретного ключа d. В даній криптосистемі існує 4 можливих значення для секретного ключа через те, що d залежить від повідомлення M.

Тому, отримуючи шифртекст C потрібно порахувати символи Лежандра:

$$\left(\frac{C^2-4}{p}\right) \text{ і } \left(\frac{C^2-4}{q}\right)$$

Обчислити ці значення можна за допомогою аналогу до алгоритму Евкліда, що може зайняти близько $O(log_2 p) + O(log_2 q)$ операцій.

Криптографічна стійкість криптосистеми LUC

Так само як і для RSA алгоритму, стійкість криптосистеми LUC базується на тому, що немає способу підрахунку $V_d(C,1) \bmod n$ не знаючи число d, чи способу знайти число d з чисел e та n. Друга задача більш складна, ніж відповідна задача для методу RSA, адже в LUC можливі чотири різні значення для d для кожної пари e та n, серед яких лише один може підійти до довільного шифротексту C.

2.5 Гомоморфне шифрування

Далі у роботі буде описано дві криптосистеми, які відносяться до так званих, гомоморфних систем шифрування. Тому, для кращого розуміння, що означає термін гомоморфне шифрування, як працює, чому воно корисне і де використовується – розглянемо основну інформацію за даним напрямком.

Гомоморфне шифрування – це тип шифрування, який передбачає, що змінюючи зашифроване повідомлення певними алгебраїчними діями, буде змінюватися і відповідний відкритий текст, до того ж, не обов'язково знати сам відкритий текст.

Вперше поняття «гомоморфне шифрування» було сформовано авторами алгоритму RSA – Рівестом та Адлманом, а також Майклом Дертузо у 1978 році[10]. Саме вони запропонували можливість виконання певних дій над зашифрованою інформацією без безпосереднього її розшифрування. Варто також зазначити, що RSA була першою асиметричною криптосистемою з гомоморфними властивостями, проте аби забезпечити безпеку RSA треба перед шифруванням провести додавання певної кількості випадкових біт до повідомлення, а через це

врачаються гомоморфні властивості.

Короткий опис гомоморфного шифрування:

Нехай k – ключ для шифрування, M_1,M_2 – відкриті тексти, E – функція для шифрування, D – функція для розшифрування, а символи $\oplus$ та $\otimes$ -– означають операції додавання та множення над зашифрованими повідомленнями відповідно.

Розрізняють Частково гомоморфні системи (дозволяє виконувати одну з операцій) та Повністю гомоморфні системи:

1) Система шифрування називається гомоморфною відносно операції додавання, якщо $D(E(k,M_1) \oplus E(k,M_2) = M_1 \oplus M_2$;

2) Система шифрування називається гомоморфною відносно операції множення, якщо $D(E(k,M_1) \otimes E(k,M_2) = M_1 \otimes M_2$;

3) Система шифрування називається повністю гомоморфною, якщо $D(E(k,M_1) \otimes E(k,M_2) = M_1 \otimes M_2$ і $D(E(k,M_1) \otimes E(k,M_2) = M_1 \otimes M_2$

Для того аби описати звичайну систему шифрування необхідно провести три операції:

1) Операція генерування ключів;

2) Операція шифрування;

3) Операція розшифрування.

Для того аби описати гомоморфну систему, необхідне проведення ще однієї операції – обчислення (**Eval**). Дана операція полягає в отриманні сервером деякої математичної функції f , використання цієї функції та приватного ключа для виконання обчислень над зашифрованим текстом та відправлення отриманих результатів отримувачу.

Області застосування гомоморфного шифрування

1) Хмарні обчислення; В хмарних обчисленнях головну роль займає продуктивність, тому є необхідність використовувати різні алгоритми для

різних операцій. Наприклад, використання RSA алгоритму для операції множення зашифрованих повідомлень, а для операції додавання є доцільним використання криптосистеми Пайє.

2) Електронне голосування;

Гомоморфне шифрування дає можливість проводити шифрування голосів виборців і підрахувати результати голосування, зберігаючи при цьому анонімність виборців.

3) Пошук інформації;

Даний тип шифрування може дозволяти користувачам отримувати інформацію з пошукових систем зі збереженням конфіденційності.

Гомоморфне шифрування є одним з найбільш перспективних методів шифрування станом на сьогодні, тому воно має чимало областей для використання і з кожним днем список цих областей тільки збільшується.

2.6 Криптосистема Окамото-Утіями

Перед початком опису криптосистеми Окамото-Утіями[8], варто спершу зазначити, що в даній криптосистемі використовується логарифмічна функція L, яка задана над підгрупою групи $(Z/nZ)^*$, де $n = p^2q$.

Нехай p це непарне просте число, а G – підгрупа $(Z/nZ)^*$:

$$G = \{x \in (Z/nZ)^* | x \equiv 1 \bmod p\}$$

Функцію L визначають на G таким чином:

$$L = \frac{x-1}{p}, \text{ де } x \in G$$

Функція L має гомоморфну властивість від множення до додавання, тому дану функцію називають логарифмічною функцією на G.

Також для опису криптосистеми необхідно знати наслідок з лемми, а саме:

$$M = \frac{L(y)}{L(x)} = \frac{y-1}{x-1} \bmod p$$

тут $x \in G$, $L(x) \neq 0 \bmod p$, $y = x^2 mod p^2$, $m \in Z/pZ$.

Більш детальний опис функції L та її властивостей можна побачити у роботі[8].

Алгоритм генерації ключів

Відкритий та секретний ключі у криптосистемі Окамото-Утіями генеруються наступним чином:

1) спочатку треба згенерувати два великих простих числа p та q;

2) далі треба обчислити число N як $N = p^2 q$;

3) випадковим чином обирається число $g \in \{2...n-1\}$, таке що $g^{p-1} \neq 0 \bmod p^2$

4) Обчислюється число $h = g^n \bmod n$

Трійка (n,g,h) формує відкритий ключ, а пара (p,q) – серктений.

Варто зазначити, що h – це додатковий параметр, який використовується для підвищення ефективності шифрування, так як h можна легко обчислити з g та n.

Алгоритм шифрування повідомлення

Відкритий текст $M < p$ зашифровується наступним чином:

1) обирається випадкове число $r \in \{1...n-1\}$;

2) обчислюється число $C = g^M h^r \bmod N$.

Значення C – отриманий шифротекст повідомлення M

Алгоритм розшифрування повідомлення

Для того аби розшифрувати зашифроване повідомлення c необхідно, використовуючи секретний ключ, виконати наступне:

1) Обчислити $a = \frac{(c^{p-1} \bmod p^2)-1}{p}$;

2) Обчислити $b = \frac{(g^{p-1} \bmod p^2)-1}{p}$;

3) Скориставшись розширеним алгоритмом Евкліда, необхідно обчислити обернене значення b' за модулем p:

$b' = b^{-1} \bmod p$

4) Обчислити $m = ab' \bmod p$.

Стійкість та властивості криптосистеми

Стійкість криптосистеми Окамото-Утіями базується на задачі факторизації числа N і потребує $O(log_3 N)$ бітових операцій.

Серед переваг криптосистеми, виділяються наступні:

1) система Окамото-Утіями є гомоморфною, тобто:

якщо $E(m)$ – функція шифрування повідомлення M, $M_1 + M_2 < p$. Тоді:

$E(M_1) \cdot E(M_2) = (g^{M_1} r_1^C)(g^{M_2} r_2^C) \bmod N = g^{M_1+M_2}(r_1 r_2)^C \bmod N = E(M_1 + M_2)$.

Ця властивість криптосистеми може бути використаною для процедури електронного голосування, а також інших криптографічних протоколів.

2) Ефективність криптосистеми. У порівнянні з системою RSA, швидкість шифрування криптосистеми Окамото-Утіями є меншою, проте швидкість процедури розшифрування є більшою аніж швидкість у схемі RSA.

3) Рандомізованість шифрованого тексту. Будь хто, навіть той хто не знає секретний ключ, може перетворити шифрований текст c у інший шифрований текст c', зберігаючи при цьому відкритий текст m, а зв'язок між c та c' можна приховати.

Така властивість є корисною для протоколів захисту конфіденційності.

2.7 Криптосистема Пайє

Криптосистема Пайє[9] використовує математичний апарат, який є подібним до того, який використовується у криптосистемі Окамото-Утіями, криптосистема так само є гомоморфною і ґрунтується на тому що факторизація числа є складною задачею.

Перед описом криптографічної системи, необхідно перевизначити логарифмічну функцію L, а саме:

Перевизначається множина

$$Z_{n,k} = \{x \in Z^*_{n^k} : x \bmod n = 1\}, \text{ при } k = 2$$

Тоді відображення L наступне:

$$L : Z_{n,k} \to Z_{n^{k-1}}$$
$$L(x) = \frac{x-1}{n}$$

для довільного $n \in N$.

Алгоритм роботи

Отже, процедура генерування ключів у алгоритмі Пайє проводиться наступним чином:

1) Обираються прості числа p та q, $p \neq q$, такі що: $\gcd(pq,(p-1)(q-1)) = 1$;

2) Обчислюються число N та функція Кармайкла λ, де $N = pq$, а $\lambda = \operatorname{lcm}(p-1,q-1)$;

3) Вибирається випадкове ціле число $g \in Z^*_{N^2}$ таке, що порядок g поділяється на N, а елемент $g' = L(g^{\lambda(N)} \bmod N^2)$ належить Z^*_N.

Пара (N,g) формує відкритий ключ;

Трійка (p,q,$\lambda(n)$) –секретний.

Алгоритм шифрування повідомлення

Нехай $M \in Z_N$ – відкритий текст який треба зашифрувати, тоді:

1) Обирається випадковим чином число $r \in Z_N^*$;

2) Обчислюється зашифрований текст: $C = g^M \cdot r^N \bmod N^2$.

Алгоритм розшифрування повідомлення

Для того аби розшифрувати зашифроване повідомлення c, необхідно обчислити:

$$M = \frac{L(C^{\lambda(N)} \bmod N^2)}{L(g^{\lambda(N)} \bmod N^2)}$$

Схема цифрового підпису

Криптосистему Пайє можна використовувати для постановки цифрового підпису, однак треба використовувати геш-функцію $h : N \mapsto \{0,1\}^k \subset Z_{N^2}^*$

Для того аби підписати відкрите повідомлення M треба обчислити насутпне:

$$s_1 = \frac{L(h(M)^{\lambda(N)} \bmod N^2)}{L(g^{\lambda(N)} \bmod N^2)}$$

$$s_2 = (h(M)g^{-s_1})^{\frac{1}{N}} \bmod \lambda$$

Пара (s_1,s_2) – цифровий підпис.

Перевірка підпису

Для перевірки цифрового підпису, треба перевірити чи виконується наступна умова:

$$h(M) = g^{s_1} s_2^N \bmod N^2$$

Властивості та області застосування криптосистеми

Розглянута криптосистема має найбільш цікаві гомоморфні властивості серед усіх відомих асиметричних криптосистем. Серед властивостей системи можна виділити:

1) при розшифруванні добутку двох шифротекстів отримаємо суму відповідних відкритих текстів:

$D(E(m_1,r_1) \cdot E(m_2,r_2) \bmod n^2) = m_1 + m_2$;

2) при домноженні зашифрованого повідомлення на g^{m_2} також отримаємо суму відповідних відкритих текстів:

$(E(m_1,r_1) \cdot g^{m_2} \mathrm{mod} n^2) = m_1 + m_2.$;

3) при піднесенні шифротексту в степінь, що дорівнює іншому шифротексту, в результаті розшифрування отримаємо добуток двох відкритих текстів:

$D(E(m_1,r_1)^{m_2} \mathrm{mod} n^2) = m_1 m_2 \mathrm{mod} n$,

$D(E(m_2,r_2)^{m_1} \mathrm{mod} n^2) = m_1 m_2 \mathrm{mod} n$.

Криптосистема Пайє застосовується для багатьох цілей, але частіше за все, враховуючи переваги гомоморфного шифрування, криптосистему як правило використовують для:

1) організації процесу проведення електронного голосування;

2) організації процесу проведення електронної лотереї;

3) розробки систем електронної валюти.

Висновки до розділу 2

В другому розділі цієї роботи було наведено детальний математичний опис криптосистем які базуються на задачі факторизації. Вказано на їхні переваги та недолікі, а також властивості а способи застосування.

Також було більш детально описано принципи гомоморфного шифрування, для яких задач воно застосовується та якими перевагами цей тип шифрування володіє.

3 АНАЛІЗ МЕТОДІВ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ РОЗГЛЯНУТИХ КРИПТОСИСТЕМ

У даному розділі продемонстровані основні методи покращення роботи криптосистем RSA, Рабіна, Пайє та Окамото-Утіями. Описані далі методи створені для того, аби забезпечити підвищення швидкості алгоритмів шифрування та розшифрування у криптосистемах, або збільшення швидкості процедури генерування цифрового підпису. Також в цьому розділі було проведено аналіз та порівняння розглянутих методів. Для методів підвищення ефективності криптосистем RSA та Рабіна наведено таблиці, в яких коротко наведено основну інформацію про ці методи, а також вказані переваги та недоліки розглянутих методів.

3.1 Методи покращення роботи RSA

Розробка криптосистеми RSA викликала не аби який спалах зацікавленості у даному алгоритмі, RSA стала надзвичайно популярною, саме тому існує чимала кількість методів покращення даного алгоритму, які за мету ставлять підвищення ефективності роботи криптосистеми та усунення недоліків оригінального алгоритму. Розглянуті в цій роботі методи, здебільшого спрямовані на пришвидшення роботи алгоритмів шифрування та дешифрування.

Модифікована схема розшифрування з використанням Китайської теореми про лишки

Науковці Дж.Дж. Квісквотер та К.Коуврер[3] запропонували власний метод покращення роботи криптосистеми RSA, який полягав у тому, що в алгоритмі розшифрування, застосовується Китайська теорема про лишки. Цю схему розшифрування згодом було названо методом

Квісквотера-Коуврера.

Суть цього методу полягає в тому, що з секретного ключа d обчислюють значення двох менших секретних ключів d_p та d_q, за допомогою яких проводиться процедура розшифрування, а результат отримується за допомогою використання Китайської теореми про лишки.

Алгоритми генерування ключів та шифрування відкритого повідомлення ідентичні до тих, які були описані у стандартній криптосистемі RSA.

Алгоритм розшифрування в свою чергу виглядає наступним чином:

1) Спочатку проводять обчислення секретних ключів d_p та d_q:

$$d_p = d \bmod (p-1) \text{ і } d_q = d \bmod (q-1)$$

2) Потім обчислюють значення:

$$M_p = C^{d_p} \bmod p \text{ і } M_q = C^{d_q} \bmod q$$

3) Скориставшись Китайською теоремою про лишки знаходять значення відкритого тексту з M_p та M_q

Детально процедуру застосування Китайської теореми про лишки було наведено у роботі Йогана Гросшадля[20].

У порівнянні зі стандартною схемою RSA, метод запропнований Квісквотером та Коуврером приблизно у 4 рази швидший. Такий приріст у швидкості забезпечується тим, що відбувається зменшення секретного ключа і операція піднесення до степеня відбувається по модулю довжини $\frac{N}{2}$ біт, так як числа p та q мають вдвічі менший порядок ніж число N.

Метод підвищення ефективності RSA з використанням складеного модулю

Наступний метод був запропонований вченими Т.Коллінзом, Д.Хопкінсом та С.Ленгфордом[11], суть якого полягала у тому, що замість звичайного модуля $n = pq$ у криптосистемі RSA, було

запропоновано використовувати складений модуль, який складався із k простих чисел.

Відмінності у порівнянні зі стандартним алгоритмом RSA полягають у схемі генерування відкритого та секретного ключів, а також у алгоритмі розшифрування повідомлення.

Варто зазначити, що кожне з k простих чисел повинно мати певний розмір, для того, аби захиститися від атак факторизації. Для того, аби даний метод був достатньо захищений, розмір числа k залежить від бітової довжини числа N. Для модуля довжини 1024 та 2048 біт рекомендовано використовувати значення $k = 3$, а для довжини 4096 та 8192 — $k = 4$.

Алгоритм **генерування ключів** у методі Коллінза у якості параметра приймає число k, що позначає кількість простих чисел, які будуть застосовуватися. Далі потрібно виконати наступні дії:

1) згенерувати k різних простих чисел p_1,p_2,...,p_k, де $k \geqslant 3$, а кожне з p_i має бути довжини $(\frac{N}{k})$ біт. Записати $N = \prod_{i=1}^{k} p_i$.

2) Обрати число e та обчислити $d = e^{-1} \bmod \varphi(N)$, потрбіно враховувати, що $\gcd(e,\varphi(N)) = 1$, $\varphi(N) = \prod_{i=1}^{k}(p_i - 1)$.

3) Для $1 \leqslant i \leqslant k$ проводиться обчислення значення $d_i = d \bmod (p_i - 1)$.

В даному методі пара (N,e) формує відкритий ключ, а набір $(N,d_1,d_2,...,d_k)$ – секретний ключ.

Для того аби **зашифрувати** відкрите повідомлення M потрібно скористатися відкритим ключем (N,e) і зашифрувати повідомлення так само як і в звичайній схемі RSA.

Алгоритм розшифрування у даному методі – це розширення до методу Квісквотера-Коуврера. Тобто, для того аби розшифрувати повідомлення c потрібно:

1) Обчислити $M_i = C^{d_i} \bmod p_i$ для кожного i, $1 \leqslant i \leqslant k$.

2) Далі застосовуємо Китайську теорему про лишки, для того аби з

усіх M_i отримати відкритий текст M.

Провдячи порівняння даного методу зі стандартним алгоритмом RSA можна дійти висновку, що розшифрування з використання Китайської теореми про лишки у стандартному RSA потребує два повних піднесення до степення за модулем $n/2$-біт, а у методі зі складеним модулем потрібно зробити k піднесень до степеня за модулем n/k біт.

Використовуючи стандартний алгоритм, для того аби розшифрувати відкритий текст M, потрібно $O(log^3 N)$ часу, а теоритичне збільшення швидкості даного методу над звичайним алгоритмом RSA дорівнює $\frac{k^2}{4}$. Тобто, для $k = 3$ метод складеного модуля у 2.25 рази швидший ніж класичний RSA.

Метод Rebalanced RSA

Описаний далі метод покращення криптосистеми RSA створений для підвищення швидкості розшифрування та генерування цифрового підпису, для того аби ці процедури були швидшими, ніж шифрування та перевірка цифрового підпису. Цей варіант криптосистеми базується на пропозиції Вінера[12]. Метод називається Rebalanced RSA[4]

В цьому методі обирається випадковим чином число d, таке що: d є достатньо великим числом за модулем N, але $d \bmod p-1$ і $d \bmod q-1$ – малі числа. Далі розглянемо більш детальний опис алгоритмів.

Алгоритм генерування ключів у методі Rebalanced приймає два секретних параметри n та k, де $k \leqslant \frac{n}{2}$. Зазвичай обираються значення $n = 1024$ і $k = 160$.

1) Спочатку генеруються два різних простих числа p та q, таких, що $\gcd(p-1,q-1) = 2$. І обчислюємо значення $N = pq$.

2) На наступному кроці обирають два k-бітних числа r_1 та r_2, таких що:

$$\gcd(r_1,p-1) = 1,\ \gcd(r_2,q-1) = 1 \text{ і } r_1 = r_2 \bmod 2$$

3) Потім проводять обчислення $d = r_1 \bmod p-1$ і $d = r_2 \bmod q-1$.

4) Знаходиться значення $e = d^{-1} \bmod \varphi(n)$.

Пара (N,e) формує відкритий ключ, а четвірка (p,q,r_1,r_2) – секретний ключ.

Шифрування відбувається аналогічно до стандартної схеми шифрування відкритого тексту M з використанням відкритого ключа (N,e) у RSA. Єдина відмінність полягає у розмірі відкритої експоненти e, в даному методі вона є набагато більшою, ніж у стандартному RSA.

Для того аби **розшифрувати** зашифроване повідомлення C треба скористатися секретним ключем і зробити наступне:

1) Обчислити значення $M_1 = C^{r_1} \bmod p$ і $M_2 = C^{r_2} \bmod q$.

2) Використовуючи Китайську теорему про лишки обчислити значення відкритого тексту M.

Теоретичне підвищення швидкості розшифрування повідомлення у методі Rebalanced RSA приблизно дорівнює $\frac{n}{2k}$. Тобто можемо підрахувати, що для модуля довжини 2048 біт та $k = 160$ дана модифікація теоретично збільшує швидкість процедури розшифрування у 6,4 рази.

Метод підвищення ефективності Rprime RSA

Метод Rprime[13] поєднує у собі два попередні методи – з використанням складеного модулю і Rebalanced RSA. Ідея даного методу полягає в тому, що для генерації ключів використовується метод Rebalanced, з поправкою на те, що замість замість двох простих чисел, буде отримано k простих чисел. Алгоритм шифрування аналогічний до того, що застосовується у стандартному RSA, а алгоритм розшифрування використовується той, що був задіяний у методі зі складеним модулем.

Алгоритм генереування ключів в даному методі виглядає наступним чином:

1) Випадковим чином генеруються l різних простих чисел p_1,p_2,...,p_l, такі що $\gcd(p_1 - 1, p_2 - 1, p_l - 1) = 2$ і обчислюється значення $N = p_1 p_2 ... p_l$.

2) Генерується l випадкових k-бітних чисел r_1,r_2,...,r_l, таких що:

$\gcd(r_1,p_1-1)=1$, $\gcd(r_2,p_2-1)=1$,...,$\gcd(r_l,p_l-1)=1$ і $r_1=r_2=...=r_l \bmod 2$.

3) Знаходиться значення d, таке що $d=r_1 \bmod p_1-1$,

$d=r_2 \bmod p_2-1$, ..., $d=r_l \bmod p_l-1$.

4) Обчислюється $e=d^{-1} \bmod \varphi(N)$.

Пара (N,e) формує відкритий ключ, а набір $(p_1,p_2,...,p_l,r_1,r_2,...,r_l)$ – секретний ключ.

Шифрування повідомлення відбувається аналогічно до процедури шифрування в стандартній криптосистемі RSA з використанням відкритого ключа (N,e), з поправкою на розмір відкритої експоненти e.

Для того аби **розшифрувати** повідомлення, проводять наступні кроки:

1) Для кожного i, $i=\overline{1,l}$, обчислюють значення $M_i=C^{r_i} \bmod p_i$.

2) Далі, скориставшись Китайською теоремою про лишки, знаходять значення відкритого тексту M.

Варто зазначити, що в модифікації Rprime RSA, збільшення швидкості процедури розшифрування залежить від кількості та розміру простих чисел, що є у відкритому ключі. Теоретичне пришвидшення розшифрування в даному методі рівне $\frac{log(n)l}{4k}$, тобто для модуля розміру 2048 біт розшифрування працює у 27 разів швидше, ніж у стандартній схемі RSA.

Порівняльний аналіз методів підвищення ефективності RSA

У таблиці (3.1) проведено порівняння розглянутих у роботі методів покращення ефективності криптосистеми RSA, наведено основну інформацію, переваги та недоліки кожного з методів.

Після проведення порівнянь розглянутих методів підвищення ефективності криптосистеми RSA можна зробити висновки:

1) Кожен з розглянутих методів має як свої переваги, так і

недоліки. Однозначно сказати, який метод є найкращим, а який найгіршим неможливо, тому що певні методи можуть бути застосовними до вирішення одних задач, а інші – до вирішення других.

2) Серед розглянутих методів покращення роботи криптосистеми RSA, найбільший приріст в швидкості дає метод Rrime, але не зважаючи на це, метод має свої недоліки.

Таблиця 3.1 – Порівняльний аналіз методів підвищення ефективності RSA

Метод покращення	Складність	Збільшення швидкості відносно оригінальної схеми	Переваги	Недоліки
Метод Квісквотера-Коуврера	$2 \cdot O((n/2)^2)$	$n^3/(2(n/2)^3)$	Основною перевагою даного методу є збільшення швидкості роботи процедури розшифрування майже в 4 рази.	В основному, ті самі недоліки, що й в стандартній криптосистемі RSA.
Метод Колінза	$k \cdot O((n/2)^3)$	$n^3/(k(n/k)^3)$	Процедура шифрування розбивається на довільну кількість піднесень до меншого степеню, що має гарний вплив на швидкість роботи.	Кожне просте число має бути певного розміру. Максимальна кількість простих чисел залежить від довжини модуля n.
Метод Reabalanced	$2 \cdot O(k(n/2)^2)$	$n^3/(2k(n/k)^2)$	Збільшення швидкості процедури розшифрування приблизно у 6 разів відносно стандартної RSA.	Пришвидшення розшифрування тягне за собою зменшення швидкості шифрування. Розмір k потрібно обирати не менше 160.
Метод Rprime	$l \cdot O(k(n/l)^2)$	$n^3/(l \cdot k(n/l)^2)$	Теоретично, швидкість розшифрування порівняно з стандартною схемою RSA більша у 27 разів.	Так як даний метод є поєднанням двох попередніх, то він має ті самі недоліки, що й вони.

3.2 Методи покращення роботи криптосистеми Рабіна

Криптосистема Рабіна хоч і не досягла успіху свого попередника у популярності, проте це зовсім не означає, що науковці взагалі не вдавалися до спроб вдосконалення цієї криптосистеми. Для алгоритму Рабіна було розроблено чимало методів покращення роботи(джерело), які в першу чергу були спрямовані на вирішення проблеми вибору правильного відкритого тексту після розшифрування повідомлення.

Метод Rabin-Williams

Хью Вільямс, автор вдосконаленої схеми для алгоритму Рабіна, намагавася вирішити проблему вибору відкритого тексту з 4 можливих варіантів при розшифруванні. Вдосконалити схему Рабіна науковець намагався шляхом застосування символу Якобі. В методі Вільямса числа p та q відповідно дорівнють $3(\mathrm{mod}8)$ та $7(\mathrm{mod}8)$ [6].

Вільямс пропонує робити кодування відкритого повідомлення $1 \leqslant M < \frac{N}{8} - 1$, як ціле число x, такого що x є парним і $(\frac{x}{N}) = 1$. Число x буде мати наступний вигляд:

$$x = P(M) = \begin{cases} 4(2M+1), & \text{якщо } (\frac{2M+1}{N}) = 1 \\ 2(2M+1), & \text{якщо } (\frac{2M+1}{N}) = -1 \end{cases}$$

Шифрування відкритого повідомлення M в даній схемі виконується наступним чином:

$$C = x^2 \bmod N$$

Процедура розшифрування у методі запропонованим Вільямсом потребує обчислення квадратних коренів, для отримання єдиного парного цілого числа x, такого що $1 < x < N$, $(\frac{x}{N}) = 1$ і $x^2 = C \bmod N$.

Якщо 8|x, тоді розшифрування повідомлення не виконується, в іншому випадку є дві можливі ситуації:

1) $m = \frac{(\frac{x}{2}-1)}{2}$, якщо $x \equiv 2 \bmod 4$;
2) $m = \frac{(\frac{x}{4}-1)}{2}$, якщо $x \equiv 0 \bmod 4$.

Розглянутий метод часто називають схемою Рабіна-Вільямса, він пропонує вирішення проблеми необхідності вибору відкритого тексту при процедурі розшифрування. Проте, не зважаючи на вирішення головного недоліку стандартної криптосистеми, метод помітно поступається схемі Рабіна у продуктивності, через те що метод Вільямса потребує обчислення символу Якобі.

Метод запропонований Куросавою

Наступний метод покращення криптографічної системи, запропонований К.Куросавою[14], так само як і схема Вільямса спрямований на те, аби вирішити проблему знаходження правильного відкритого тексту. Модифікована схема Куросави використовує два додаткових біти інформації, для того аби вказати на правильний квадратний корінь, а також у схемі застосовується обчислення символу Якобі, у алгоритмах шифрування та розшифрування.

Метод Куросави, так само які стандартна криптосистема Рабіна, використовує у якості секретного ключа пару великих простих чисел p та q, відмінність полягає у публічному ключі. Тут ним слугує пара, що складається з $N = pq$, та числа $l \in Z_N$, такого що:

$$(\tfrac{l}{p}) = (\tfrac{l}{q}) = -1$$

В якості відкритого повідомлення використовується M, таке що: $0 < M < N$ і $gcd(M,N) = 1$

Додаткові біти інформації у даній схемі визначаються так:

$$s = \begin{cases} 0, & \text{якщо } \frac{l}{M} > M \\ 1, & \text{якщо } \frac{l}{M} < M \end{cases}; \quad t = \begin{cases} 0, & \text{якщо } (\frac{M}{N}) = 1 \\ 1, & \text{якщо } (\frac{M}{N}) = -1 \end{cases}$$

Шифрування повідомлення у цій модифікації проводиться так:

$$C = M + (\frac{l}{M}) \bmod N$$

Для того аби **розшифрувати** повідомлення, потрібно розв'язати наступне квадратне рівняння:

$$M^2 - CM + l = 0$$

і обрати той корінь, який буде визначений парою (s,t)

Розглянутий метод слугує альтернативою до стандартної криптосистеми Рабіна. Схема Куросави працює для кожної пари простих чисел p та q. Цей метод вирішує проблему ідентифікації правильного відкритого тексту з чотирьох можливих варіантів. Проте, в процесах шифрування та розшифрування даного методу є необхідність обчислення символу Якобі, а це призводить до додаткових обчислювальних витрат.

Метод покращення Rabin-Boneh

Rabin-Boneh[15], або як його ще часто називають SAEP-Rabin — це метод підвищення ефективності криптосистеми Рабіна, який використовує паддінговий протокол шифрування SAEP(Simple Asymmetric Encryption padding)[16]. Ден Боне запропонував робити накладання деяких властивостей на відкритий текст перед процедурою шифрування. Цей метод вважається одним з найбільш оптимальних з точки зору ефективності роботи криптосистеми, так як в цьому методі не використовується символ Якобі і з великою імовірністю, при розшифруванні буде одержано правильний відкритий текст.

В методі Боне, так само як і в системі Рабіна обирають прості числа p та q, такі що $p \equiv q \equiv 3 \bmod 4$ і обчислюється значення $N = pq$.

Протокол SAEP потребує застосувати деякі обмеження на довжину модуля. Нехай $k \in N$– це секретний параметр, потрібно обрати числа p та q довжини $\frac{k}{2} + 1$ біт. Тоді модуль буде довжини $k + 2$ біт, тобто $2^{k+1} \leqslant N < 2^{k+2}$.

Як було сказано раніше, даний метод полягає в тому, що перед початком шифрування, відкритий текст доповнюється, тому потрібно виконати наступні кроки:

1) Нехай M – відкритий текст довжини l біт. До повідомлення додають l_0 нульових біт, щоб отримати наступне значення:

$$M^0 = M \parallel 0^{l_0}$$

2) Потім випадковим чином обирають ціле число r, довжина якого дорівнює l_1 біт. Далі застосовують геш-функцію $h : \{0,1\}^{l_1} \rightarrow \{0,1\}^{l_0+l}$ наступним чином аби отримати доповнене повідомлення:

$$y = (M^0 \oplus h(r)) \parallel r$$

Шифрування повідомлення в методі Rabin-Boneh відбувається шляхом піднесення y до квадрату за модулем N:

$$C = y^2 \bmod n$$

На довжину доповненого повідомлення накладається обмеження: $l \leqslant \frac{k}{4}$, $l + l_0 \leqslant \frac{k}{2}$ і $k = l + l_0 + l_1$. Довжина y дорівнює k біт, отже $2^{k-1} \leqslant y < 2^{k+1}$. Це обмеження накладається для виконання процедури розшифрування в даному методі.

В процедурі **розшифрування**, проводиться обчислення коренів c, після обчислення одразу можна видалити ті корені, які є більшими за $\frac{N}{2}$. Після цього залищається один, або два корені.

Якщо можливі два варіанти y_1 та y_2, то знаходять $y_1 = v_1 \parallel r_1$ і $y_2 = v_2 \parallel r_2$. Звідси знаходять m_1^0 та m_2^0. Для того аби вибрати правильне повідомлення проводять перевірку, чи l_0 останніх біт нулі.

Хоча метод SAEP-Rabin вважається одним з найбільш ефективних, варто також вказати, що не зважаючи на високу імовірність одержання правильних результатів, все ж існує певна імовірність того, що алгоритм дешифрування може невдало спрацювати.

Метод Rabin-p

Метод під назвою Rabin-p був запропонований у 2014 році[17] і вважається відносно новим методом покращення криптосистеми Рабіна. Ця схема є подібною до стандартної схеми Рабіна, проте вона є більш ефективною і покращує властивості стандартної криптосистеми. Rabin-p не використовує функцію доповнення у процедурах шифрування та розшифрування повідомлення, як більшість інших варіантів схеми Рабіна. Ще однією перевагою цього методу є те, що під час дешифрування результат отриманий тільки один, а не 4 як в стандартній криптосистемі, тобто Rabin-p безпомилково розшифровує повідомлення.

В даному методі використовується одне просте число для розшифрування, саме тому метод має таку назву. Використання лише одного просто числа в процедурі дешифрування значно зменшує затрати на обчислення.

На вхід алгоритм приймає число k – розмір секретного параметру. Далі для генерування ключів обирають два різних простих числа p та q, таких що $2^k < p,q < 2^{k+1}$ і обчислюють значення $N = p^2q$

Секретним ключем буде число p, а N– відкритим ключем.

Далі обирається відкрите повідомлення M для шифрування, таке що $0 < M < 2^{2k-1}$ і $\gcd(M,N) = 1$.

Шифрування відбувається як у стандартному алгоритмі Рабіна, а саме:

$$C = M^2 \bmod N$$

Дешифрування в даній схемі відбувається з використанням секретного ключа p. Для виконання розшифрування необхідно виконати наступні кроки:

1) Обчислити: $w = C \bmod p$;

2) Обчислити: $M_p = w^{\frac{p+1}{4} \bmod p}$;

3) Обчислити : $i = \frac{(c-M_p^2)}{p}$;

4) Обчислити: $j = \frac{i}{2M_p \bmod p}$;

5) Обчислити: $M_1 = M_p + jp$;

6) Перевірити, чи $M_1 < 2^{2k-1}$, якщо так тоді $M = M_1$;

7) Якщо ні, тоді $M = p^2 - M_1$.

Метод Rabin-p створювався для того, аби підвищити ефективність стандартної схеми Рабіна без використання символу Якобі, додаткових біт інформації або функції доповнення.

У 2018 році[22] проводились порівняння методу Rabin-p та різних криптосистем, таких як RSA та її модифікацій, де було досліджено, що Rabin-p є набагато більш ефективним, ніж RSA у контексті часу роботи та розміру вхідного повідомлення.

Порівняльний аналіз методів підвищення ефективності криптосистеми Рабіна

У таблиці (3.2) проведено порівняння розглянутих методів збільшення ефективності криптографічної системи Рабіна. В таблиці коротко наведена необхідна інформація, описана головна задача кожного з методів, а також переваги та недоліки цих методів показують, чи була виконана ця задача і якою ціною.

Проаналізувавши перечислені методи підвищення ефективності криптосистеми Рабіна, можна дійти висновку, що серед розглянутих методів найбільш оптимальним є метод Rabin-p, адже він помітно збільшує ефективність роботи стандартної криптосистеми. І хоча цей метод має недоліки, наприклад атака на основі підібраного шифротексту, проте цей недолік може бути вирішений шляхом накладання певних обмежень на відкритий текст.

Таблиця 3.2 – Порівняльний аналіз методів підвищення ефективності криптосистеми Рабіна

Метод покращення	Головна ціль методу	Переваги	Недоліки
Метод Williams	Вирішення проблеми вибору правильного відкритого тексту з 4 можливих варіантів	Запропонований метод вирішує проблему вибору відкритого тексту, а також цей метод не розкриває інформацію про зашифрований текст	Метод потребує виконувати обчислення символу Якобі, що є ресурсоємною операцією; метод поступається у продуктивності оригінальній схемі Рабіна
Метод Куросави	Усунення проблеми вибору правильного відкритого тексту	На числа p та q не накладаються обмеження, вони можуть бути довільними; вирішує проблему вибору правильного відкритого тексту	Додаткові обчислювальні затрати, через те що в алгоритмах шифрування та розшифрування проводиться обчислення символу Якобі
Метод Boneh	Підвищення швидкості роботи криптосистеми шляхом використання протоколу доповнення SAEP	З великою імовірністю буде одержано єдиний правильний відкритий текст після розшифрування повідомлення; ефективність даного методу вища за попередні методи, через те що не використовується символ Якобі	Існує імовірність того, що алгоритм дешифрування видасть неправильний результат відкритого тексту
Метод Rabin-p	Покращення властивостей стандартної крипстосистеми Рабіна	Відсутність некоректної роботи алгоритму розшифрування; значно менші затрати на обчислення	Вразливість перед атакою на основі підібраного шифротексту

3.3 Методи покращення роботи криптосистем Пайє та Окамото-Утіями

Методи підвищення ефективності криптографічних систем Пайє та Окамото-Утіями було вирішено об'єднати в один підрозділ, для кращого сприйняття матеріалу.

Застосування Китайської теореми про лишки

Для криптосистеми Пайє, так само як і для криптосистем Рабіна та

RSA застосовують Китайську теорему про лишки, для того аби підвищити швидкість роботи алгоритму розшифрування.

В даному методі, так само як і в оригінальній криптосистемі є три алгоритми: генерування ключів, шифрування та розшифрування. Перші два алгоритми не відрізняються від тих, що були наведені у стандартній криптосистемі Пайє, але алгоритм розшифрування відрізняється від того що був представлений раніше.

Функція L, яка була задана як $L = \{x < n^2 | x = 1 \bmod n\}$, застосовується для двох модулів p та q, наступним чином:

$$L_p = \{x < p^2 | x = 1 \bmod p\} \text{ і } L_q = \{x < q^2 | x = 1 \bmod q\}$$

Отже, $L_p(x) = \frac{x-1}{p}$, $L_q(x) = \frac{x-1}{q}$

Далі в методі значення $\mu = (L(g^\lambda \bmod n^2)^{-1}) \bmod n$ записують через суму значень h_p та h_q. Формули для h_p та h_q записують таким чином:

$$h_p = [L_p(g^{p-1} \bmod p^2)]^{-1} \bmod p \text{ і } h_q = [L_q(g^{q-1} \bmod q^2)]^{-1} \bmod q,$$

ці значення обчислюють попередньо.

Для того аби розшифрувати відкритий текст M використовуючи цей метод необхідно обчислити значення:

$$M_p = L_p[(C^{p-1} \bmod p^2)h_p \bmod p] \text{ і } M_q = L_q[(C^{q-1} \bmod q^2)h_q \bmod q]$$

і застосувати Китайську теорему про лишки для того аби отримати правильний відкритий текст M.

Детально ознайомитися з процедурою застосування Китайської теореми про лишки у даному методи можна тут[18].

Цей метод має позитивний вплив на швидкість проведення процедури розшифрування, адже всі необхідні операції проводяться за меншим модулем. Тому даний метод може бути застосовним для вирішення певних задач. Обчислювальна складність даного методу дорівнює $O(n^2)$.

Метод зменшення складності процедури розшифрування

Для криптосистеми Окамото-Утіями був запропонований метод зменшення складності процеури розшифрування до $O(log^2 n)$[19], при цьому зберігаючи еквівалентність до задачі факторизаціїї.

Метод полягає в тому, що число p обирається таким чином, що $p-1$ має великий коефіцієнт t, де розмір t не менше 160 біт і t є простим числом. В цьому методі записують: $p-1=tu$ і виконується наступне:

1) Випадковим чином обирається число g, таке що $g < n$, позначається $g_p = g^{p-1} \bmod p^2$, так само як це було у стандартній криптосистемі. Потім проводиться обчислення значення $G = g^u \bmod n$;

2) Далі випадковим чином обирається значення $g' < n$ і обчислюється $H = (g')^{nu} \bmod n$

Трійка (n,G,H) – утворює відкритий ключ, а секретний ключ це пара чисел (p,q).

Для **зашифрування** відкритого повідомлення M, обирається значення $r < n$ і обчислюється:

$$C = G^M H^r \bmod n$$

Розшифрування в даному методі виглядає аналогічно до того, що було представлено у стандартній криптосистемі.

Даний метод виконує свою голвну ціль та понижує кубічну складність алгоритму розшифрування криптосистеми Окамото-Утіями до квадратичної складності, зі збереженням еквівалентності до факторизації.

Для криптосистеми Окамото-Утіями було розроблено не так багато методів підвищення ефективності її роботи, тому в майбутньому варто звернути увагу на цю криптосистему, адже властивості цієї криптосистеми можуть бути корисними і застосовуватися у різних сферах.

Висновки до розділу 3

У даному розділі наведений детальний опис розглянутих методів підвищення ефективності, які застосовуються до криптосистем RSA, Рабіна, Окамото-Утіями та криптосистеми Пайє. Було наведено інформацію про основні відмінності цих методів від стандартних криптосистем, описано відмінності в алгоритмах, а також, після проведеного аналізу та порівняння, було вказано на переваги та недоліки кожного з методів.

Також у розділі, для криптосистем RSA та Рабіна, усі розглянуті методи були записані у таблиці, в яких проаналізовано кожен метод. У таблицях коротко вказано головну інформацію про досліджені методи, а саме:

1) складність роботи;
2) зміни відносно оригіналу;
3) переваги;
4) недоліки.

Після проведеного аналізу було показано, що кожен метод з одного боку покращує певні параметри криптосистеми і водночас зменшує ефективність інших. Тому однозначно виділити один найкращий метод для кожної криптосистеми серед усіх розглянутих — неможливо.

ВИСНОВКИ

Криптографічний захист інформації вже давно є невід'ємною частиною сучасного світу. Майже щодня відбуваються процеси боротьби методів захисту інформації з атаками зловмисників, які намагаються отримати доступ до конфіденційної інформації. Саме це спонукає криптографію продовжувати свій розвиток, і саме тому для різних криптографічних систем продовжують розробляти нові та вдосконалювати вже наявні методи підвищення ефективності їх роботи.

В цій роботі розкрито взаємозв'язок в історичному контексті між іменами, ідеями та результатами, які були отримані різними науковцями, що призвели до появи сучасних криптографічних алгоритмів та створених для них модифікацій.

Було наведено детальний математичний опис криптографічних систем, які ґгрунтуються на задачі факторизації, вказані властивості цих криптосистем, їх переваги та недоліки. Також наведено основну інформацію про принципи гомоморфного шифрування, для кращого розуміння роботи криптосистем Окамото-Утіями та Пайє, які власне і мають властивості гомоморфного шифрування.

У роботі детально описані розроблені методи підвищення ефективності розглянутих криптографічних систем, вказано у чому полягають відмінності від стандартних алгоритмів, а також проведено аналіз кожного з методів, після якого було вказано на переваги та недоліки кожного з них.

Також, для методів покращення роботи криптосистем RSA та Рабіна, були побудовані порівняльні таблиці, які демонструють головну інформацію про кожен метод, складність роботи алгоритмів, підрахунок покращень відносно стандартної криптосистеми, а також загальну інформацію про переваги та недоліків цих методів.

Результати даного дослідження можуть бути використанні для продовження подальшої роботи за заданою темою, а саме для покращення розглянутих методів підвищення ефективності.

ПЕРЕЛІК ПОСИЛАНЬ

1. W. Diffie and M. Hellman, "New directions in cryptography,"in IEEE Transactions on Information Theory, vol. 22, no. 6, pp. 644-654, November 1976

2. Rivest R. L., Shamir A., Adleman L. A method for obtaining digital signatures and public-key cryptosystems //Communications of the ACM. – 1978. – Vol. 21. – №. 2. – P. 120-126.

3. Quisquater J. J., Couvreur C. Fast decipherment algorithm for RSA public-key cryptosystem //Electronics letters. – 1982. – Vol. 18. – №. 21. – P. 905-907.

4. Boneh D., Shacham H. Fast variants of RSA //CryptoBytes. – 2002. – Vol. 5. – №. 1. – P. 1-9

5. M.O. Rabin, Digitalized signatures and public-key functions as intractable as factorization, Tech. Report MIT/LCS/TR-212, MIT Laboratory for Computer Science, 1979.

6. Hugh C. Williams. Some Public-Key Crypto-Functions as Intractable as Factorization, 1985.

7. P. J. Smith and M. J. J. Lennon, LUC: A new public key system, International Conference on Information Security (E. Graham Dougall, ed.), IFIP Transactions, vol. A-37, North-Holland, 1993, pp. 103–117.

8. T. Okamoto and S. Uchiyama, A new public-key cryptosystem as secure as factoring, EUROCRYPT 1998 (K. Nyberg, ed.), LNCS, vol. 1403, Springer, 1998, pp. 308–318.

9. P. Paillier, Public-key cryptosystems based on composite degree residuosity classes,EUROCRYPT 1999 (J. Stern, ed.), LNCS, vol. 1592, Springer, 1999, pp. 223–238

10. Rivest R. Data banks and privacy homomorphisms / R. Rivest, L. Adleman, M. Dertouzos. // Foundations of Secure Computation. – 1978. – C. 159–180.

11. Collins T. et al. Public key cryptographic apparatus and method :

пат. 5848159 США. – 1998.

12. Wiener M. J. Cryptanalysis of short RSA secret exponents //IEEE Transactions on Information theory. – 1990. – Vol. 36. – №. 3. – P. 553-558.

13. Cesar Alison Monteiro Paix˜ao. An efficient variant of the RSA cryptosystem // Cryptology ePrint Archive, Report 2003/159. 2003.

14. Kurosawa, K., Ito, T., and Takeuchi, M. (1991). Public key cryptosystem using a reciprocal number with the same intractability as factoring a large number. Cryptologia, 12(4):225–233.

15. Boneh, D. (2001). Simplified oaep for the rsa and rabin functions. In In Advances in Cryptology-Crypto 2001, pages 275–291. Springer.

16. K. T. Kim and M. H. Son, "SAEP: Secure and Accurate and Energy-Efficient Time Synchronization in Wireless Sensor Networks,"2009 Eighth International Symposium on Parallel and Distributed Computing, 2009, pp. 117-120

17. Asbullah, Muhammad Asyraf and Kamel Ariffin, Muhammad Rezal (2014) Design of Rabin-like cryptosystem without decryption failure. Malaysian Journal of Mathematical Sciences, 10 (spec. Aug.). pp. 1-18. ISSN 1823-8343; ESSN: 2289-750X

18. H. Xiao, Y. Huang, Y. Ye, and G. Xiao, “Robustness in Chinese Remainder Theorem,” arXiv, 2007, pp. 1–12.

19. Coron, J.-S., Naccache, D., Paillier, P.: Accelerating Okamoto-Uchiyama public-key cryptosystem. Electronics Letters 35(4), 291-292 (1999)

20. J. Grossschadl, "The Chinese Remainder Theorem and its application in a high-speed RSA crypto chip,"Proceedings 16th Annual Computer Security Applications Conference (ACSAC’00), 2000, pp. 384-393

21. Steven D. Galbraith: Mathematics of Public Key Cryptography. Version 2.0. Cambridge University Press, 2018.

22. M. Adli, Perbandingan Efisiensi Algoritma Rabin-P dan Algoritma RSA pada Pengamanan File Txt Berbasis Desktop, Deg. Thesis, Univ. Sumatera Utara, Indonesia, 2018.