МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чернігівський національний технологічний університет

Кафедра інформаційних та комп'ютерних систем

ЗАХИСТ ІНФОРМАЦІЇ В КОМП'ЮТЕРНИХ СИСТЕМАХ
МЕТОДИЧНІ ВКАЗІВКИ

до виконання розрахунково-графічних робіт

з дисципліни «Захист інформації в комп'ютерних системах»

для студентів спеціальності

123 - «Компютерна інженерія»

денної форми навчання

Обговорено і
рекомендовано
на засіданні кафедри
інформаційних та
комп'ютерних систем
Протокол № 6
від 12 січня 2019р.

Чернігів 2019 рік

Захист інформації в комп'ютерних системах Методичні вказівки до виконання розрахунково-графічних робіт для студентів спеціальності 123 «Комп'ютерна інженерія» денної форми навчання / Укл.: Гур'єв В.І., Казнадій С.П., Усов Я.Ю. – Чернігів: ЧНТУ, 2019. – 27 с.

Укладачі: ГУР'ЄВ ВОЛОДИМИР ІВАНОВИЧ, кандидат технічних наук, доцент, професор кафедри кібербезпеки та математичного моделювання
КАЗНАДІЙ СВІТЛАНА ПЕТРІВНА, ст. викладач кафедри інформаційних та комп'ютерних систем
УСОВ ЯРОСЛАВ ЮРИЙОВИЧ, викладач кафедри кібербезпеки та математичного моделювання

Рецензент: СКІТЕР ІГОР СЕМЕНОВИЧ , кандидат физ.-мат. наук, доцент кафедри
інформаційних технології та програмної інженерії

Відповідальний: Завідувач кафедри інформаційних та комп'ютерних систем С.В.Зайцев

ЗМІСТ

ВСТУП

На відміну від симетричних криптосистем, у яких процедура розшифрування легко відновлюється за процедурою шифрування та навпаки, у схемі шифрування з відкритим ключем неможливо визначити процедуру розшифрування, якщо відома процедура шифрування. Більш точно, час роботи алгоритму, який обчислює процедуру розшифрування настільки великий, що його неможна виконати на довільних сучасних комп'ютерах, як й на комп'ютерах майбутнього. Такі схеми шифрування називаються асиметричними.

Розрахунково-графічна робота складається з 4 завдань:

1. Дешифрування повідомлення, яке зашифроване за криптосистемою RSA.

2. Ідентифікація користувачів за системою Гиллоу-Куінскуотера.

3. Розділення та відновлення секрету для сітки з n абонентів за схемою поділу секрету Шаміра.

4. Підпис та верифікація повідомлення за схемою електронного цифрового підпису Ель-Гамаля.

АЛГОРИТМ RSA

RSA — криптографічний алгоритм з відкритим ключем.

RSA став першим алгоритмом такого типу, що придатний й для шифрування й для цифрового підпису. Алгоритм використовується у великій кількості криптографічних прикладень.

Історія

Опис RSA був надрукований в 1977 році Рональдом Райвестом (Ronald Linn Rivest), Аді Шаміром (Adi Shamir) и Леонардом Адлеманом (Leonard Adleman) из Массачусетського Технологічного Інституту (MIT). Система була названа за першими літерами їхніх призвищ.

Британський математик Кліффорд Кокс (Clifford Cocks), який працював у центрі урядового зв'язку (GCHQ) Великобританії, описав аналогічну систему в 1973 році у внутрішніх документах центру, но ця робота не була розкрита до 1977 року й Райвест, Шамір и Адлеман розробили RSA незалежно від праці Кокса.

У 1983 році MIT був виданий патент 4405829 США, термін дії якого закінчився **21 вересня 2000 року**.

У 1977 році авторами RSA була зашифрована фраза «The Magic Words are Squeamish Ossifrage» («Секретні слова — це бридливий ягнятник»). За розшифрування було запрпановано винагороду в 100 долларів США. Фраза была розшифрована в 1993—1994 роках. Більш ніж 600 добровольців жертвували процесорний час близько 1600 машин (дві з яких були факс-машинами) більше шести місяців. Координація проходила через Інтернет, і це був перший з подібних проектів розподілених обчислень. Отриману винагороду переможці пожертвували до фонда вільного програмного забезпечення.

Опис алгоритму

Безпека алгоритму RSA заснована на важкості задачі розкладання на множники. Алгоритм використовує два ключа — відкритий (public) и секретний (private), разом відкритий та відповідний ключі утворюють пару ключів (keypair). Відкритий ключ не потрібно зберігати в таємниці, він використовується для шифрування даних. Якщо повідомлення було зашифроване відкритим ключем, то розшифрувати його можна тільки **відповідним ключем**.

1.2.1. Генерація ключів

Для того, щоб згенерувати пару ключів виконуються наступні дії:

1. Вибираються два великих випадкових простих числа p і q
2. Обчислюється їхній добуток $n = pq$
3. Обчислюється Функція Ейлера $\varphi(n) = (p-1)(q-1)$
4. Вибирається ціле e таке, що $1 < e < \varphi(n)$ і e взаємно просте з $\varphi(n)$
5. За допомогою розширеного алгоритму Евклида знаходиться число d таке, що $ed \equiv 1 \pmod{\varphi(n)}$ це означає, що $de = 1 + k\varphi(n)$ при деякому цілому k.

Число n називається модулем, а числа e і d — відкритою та секретною експонентами відповідно. Пара чисел $(n,\ e)$ є відкритою частиною ключа, а d — секретною. Числа p і q після генерації пари ключів можуть бути знищені, але в жодному випадку не повинні бути розкриті.

Шифрування та розшифрування

Для того, щоб зашифрувати повідомлення з $m < n$ обчислюється $c = m^e \bmod n$.

Число c використовується у якості шифртекста. Для розшифрування треба обчислити

$m = c^d \bmod n$.

Неважко з'ясувати, що при розшифруванні ми отримаємо вихідне повідомлення:

$c^d \equiv (m^e)^d \equiv m^{ed} \pmod{n}$

З умови

$ed \equiv 1 \pmod{\varphi(n)}$

слідує, що

$ed = k\varphi(n) + 1$ для деякого цілого k, отже

$m^{ed} \equiv m^{k\varphi(n)+1} \pmod{n}$

Згідно з теоремою Ейлера:

$m^{\varphi(n)} \equiv 1 \pmod{n}$,

тому

$m^{k\varphi(n)+1} \equiv m \pmod{n}$

$c^d \equiv m \pmod{n}$

1.2.2. Приклад

Згенеруємо пару ключів. Виберемо два простих
$p = 3557$
$q = 2579$

Обчислимо модуль
$n = p \cdot q = 3557 \cdot 2579 = 9173503$
Обчислимо функцію Ейлера
$\varphi(n) = (p-1)(q-1) = 3556 \cdot 2578 = 9167368$
Покладемо відкритий показник рівним 3
$e = 3$
За допомогою розширеного алгоритму Евклида обчислимо секретний показник
$d = 6111579$
Відмітимо, що
$e \cdot d \bmod \varphi(n) = 3 \cdot 6111579 \bmod 9167368 = 1$

Пара чисел (3, 9173503) утворює відкриту частину ключа, а число 6111579 є секретним ключем.
Зашифруємо за допомогою відкритого ключа число $m = 111111$:
$c = m^e \bmod n = 111111^3 \bmod 9173503 = 4051753$

Шифртекстом є число 4051753.

Для розшифрування обчислюємо
$$c^d \bmod n = 4051753^{6111579} \bmod 9173503 = 111111$$

в результаті отримаємо вихідне повідомлення.

СХЕМА ІДЕНТИФІКАЦІЇ ГІЛЛОУ-КУІСКУОТЕРА

Алгоритм ідентифікації з нульовим передаванням знання, який розробили Л.Гіллоу (Louis Guillou) і Ж.Куіскуотер (Jean-Jacques Quisquater)), має декілька кращі характеристики, ніж інші схеми ідентифікації. У цьому алгоритмі обмін між сторонами А і В та аккредитації у кожному обміні доведено до абсолютного мінімуму - для кожного доведення потрібний тільки один обмін з однією акредитацією. Але об'єм потрібних обчислень для цього алгоритму більший ніж для схеми Фейге-Фіата-Шаміра.

Нехай сторона A – інтелектуальна картка, яка повинна довести свою справжність стороні B, яка буде перевіряти. Ідентифікаційна інформація сторони A представлє собою бітовий рядок I, який містить ім'я власника картки, строк дії, номер банківського рахунку і т.ін. Фактично ідентифікаційні дані можуть займати достатньо довгий рядок, тому їх хеширують до значення I.

Рядок I є аналогом відкритого ключа. Іншою відкритою інформацією, яку використовують всі картки, що приймають участь у даному додатку, є модуль n та показник степеню V. Модуль n є добутком двох секретних простих чисел.

Секретним ключем сторони A є величина G, яка обирається таким чином, щоб виконувалося співвідношення

$$I * GV \equiv 1 \pmod{n}.$$

Сторона A отправляє стороні B свої ідентифікаційні дані I. У подальшому вона повинна довести стороні B, що ці ідентифікаційні дані належать безпосередньо їй. Щоб домогтися цього, сторона A повинна переконати сторону B, що їй відомо значення G.

Протокол доведення справжності A без передачі стороні B значення G:

1.Сторона A обирає випадкове ціле r, таке, що $1 < r \leq n - 1$. Вона обчислює

$$T = rV \bmod n$$

та відправляє це значення стороні B.

2. Сторона B обирає випадкове ціле d, таке, що $1 < d \leq n - 1$, та відправляє це значення d стороні A.

3. Сторона A обчислює

$$D = r * Gd \bmod n$$

та відправляє це значення стороні B.

4. Сторона B обчислює значення

T' = DV Id mod n.

Якщо T ≡ T' (mod n), то перевірка справжності успішно завершена. Математичні викладки, які використані у цьому протоколі не дуже складні:

T' = DV Id = (r Gd)V Id = rV GdV Id = rV (IGV)d = rV ≡ T(mod n).

оскільки G обчислювалося таким чином, щоб виконувалося співвідношення IGV ≡ 1 (mod n).

РОЗДІЛЕННЯ СЕКРЕТУ

Кожна частка секрету — це площина, а секрет представляє собою точку перетину трьох площин. Дві частки секрету дозволяють отримати лінію, на якій лежить секретна точка.

В криптографії під розділенням секрету (англ. Secret sharing) розуміють будь-який метод розподілу секрету серед групи учасників, кожному з яких каждому из которых дістається частка секрету. Секрет потім може відтворити коаліція участників. Частка секрету сама по собі не несе ніякої інформації.

У схемі розділення секрету приймає участь один дилер і n гравців. Спочатку секрет знаходится у дилера, потім дилер обчислює частки секрету і роздає їх учасникам, кожному участнику по одній частці. Зазвичай для схеми розділення секрету визначений поріг t: будь-яка група з t або більше гравців, зібравшись разом, може відновити секрет, але ніяка група з менше ніж t гравців не зможе. Таку схему розділення секрету називають (t, n)-порогова схема (англ. (t, n)-threshold scheme).

3.1 Мотивація

Надійна схема розділення секрету дозволяє розподілити частки секрету так, що будь-яка коаліциия з менш ніж t гравців не зможе отримати абсолютно ніякої інформації про секрет. Розглянемо наївну схему розділення секрету, у якій секретне слово «пароль» ділиться на частки «па----», «--ро--» і «----ль». Припустимо Д у хакера немає ні якої частки секрету, ни одной доли секрета, тоді йому доведеться пробувати усі

можливі шестилітерні паролі, а це 336=1,2 млрд комбінацій. Якщо хакер зміг отримати одну частку, то йому тепер потрібно вгадати тільки чотири літери, а це 334=1,2 млн комбинацій, що набагато менше. Така система надійна, так як коаліція з менш ніж t гравців може отримати значну інформацію про секрет.

Тривіальна схема

Існує множина (t, n) - схем для яких t = n, тобто для відтворення секрету потрібні всі частки, наприклад:

- Закодуємо секрет як ціле число *s*. Дамо усім гравцям крім одного по випадковому числу r_i, а останньому гравцеві дамо число $r_n=s-r_1-r_2-\ldots-r_{n-1}$. Для відтворення секрету потрібно додати всі числа r_i.
- Закодуємо секрет як байт *s*. Дамо усім гравцям крім одного по випадковому байту r_i, а останньому гравцеві дамо байт $r_n=s \ XOR \ r_1 \ XOR \ r_2 \ XOR \ \ldots \ XOR \ r_{n-1}$. Для відтворення секрету потрібно застосувати до всіх чисел r_i операцію XOR.

Коли обсяг використовуваної пам'яті некритичний, то можна використовувати схемы такого роду для будь-якої підмножини гравців та для кожної можливої коаліції гравців. Наприклад, маємо трьох гравців: Аліса, Боб та Керол. Для створення схеми у якій будь-які два гравці можуть відтворити секрет, ми можемо створити три різних (2,2) схеми Кожна схема дасть нам дві частки секрету. Дві частки першої схеми дамо Алісі та Бобу, дві частки другої схеми — Бобу і Керол, а дві частки третьої схеми — Алісі і Керол. Однак такий підхід дуже швидко стає неефективним разом із збільшенням **n** і **t**.

3.2 Приклад t ≠ n

Прпустимо Допустим, рада директорів компанії Кока-Кола бажає захистити секретну формулу їх напою. Президент компанії повинний мати доступ до формули, а також будь-які л 3 з 12 членів ради директорів повинні зуміти зібравшись разом отримати формулу. Таку задачу можна розв’язати (3,15) схемою розділення секрету, де президент отримає 3 частки секрету, а кожний член радиа директорів — по одній частці.

3.3 Схема Блеклі

Дві непаралельні прямі на площині перетинаються у одній точці. Три непаралельні площини у просторі теж перетинаються у одній точці. Взгалі n n-вимірних гіперплощин завжди перетинаються у одній точці. Одна з координат цієї точки буде секретом. Якщо закодувати секрет як

декілька координат точки, то вже по одній частці секрету (одній гіперплощині) можна буде отримати якусь інформацію про секрет, тобто про взаємозалежності координат точки перетину. За допомогою схеми Блекли можна створити (t, n)-схему розділення секрету для довільних t і n: для цього необхідно покласти розмірність простору рівну t, та кожному із n гравців дати одну гіперплощину, що проходить через секретну точку.

Тоді будь-які **t** із **n** гіперплощин будуть однозначно перетинатися у секретній точці. Схема Блеклі менш ефективна ніж схема Шміра: у схемі Шаміра кожна частка має такий же розмір як і секрет, а у схемі Блеклі кожна частка у **t** разів більше. Існують поліпшення схеми Блеклі, які дозволяють підвищити її ефективність.

3.4 Схема Шаміра

Основна ідея схеми Шаміра заключається у тому, що двох точок достатньо для задання прямої, трьох точок – для задання параболи, чотирьох точок – для кубічної параболи, і так далі. Для задання многочлену степеня n потрібно n+1 точок. Припустимо, потрібно створити (k, n)-порогову схему Шаміра (k<n) для розділення секретного числа S. Вибиремо k – 1 випадкових коефіцієнтів a_1, …, a_{k-1}, а також нехай a_0=S. Візьмемо многочлен $f(x) = a_0 + a_1x_1 + a_2x_2 + \ldots + a_{k-1}x_{k-1}$. Частками секрету будуть n пар: : (i, f(i)), де i=1…n. Маючи k часток секрету можна обчислити всі коефіцієнти многочлена, наприклад за допомогою інтерполяційного многочлена Лагранжа, а отже і секрет $S=a_0$.

АЛГОРИТМ ЦИФРОВОГО ПІДПИСУ ЕЛЬ ГАМАЛЯ (EGSA)

Назва EGSA походить від от слів El Gamai Signature Algorithm (алгоритм цифрового підпису Ель Гамаля). Ідея EGSA основана на том, що для обґрунтування практичної неможливості фальсифікації цифрового підпису може бути використана більш складна обчислювальна задача, ніж розкладання на множники великого цілого числа, задача дискретного логарифмування. Крім того Ель Гамалю вдалося уникнути явної слабкості алгоритму цифрового підпису RSA, пов'язаної з можливістю підробки цифрового підпису під деякими повідомленнями без визначення секретного ключа.

Для того, щоб генерувати пару ключів (відкритий ключ секретний ключ), спочатку вибирають деяке велике просте ціле число P і велике ціле число G, причому G<P. Відправник і одержувач підписаного документа використовують при обчисленнях однакові великі цілі числа P (~10308 або ~21024) та G (~10154 або ~2512), які не є секретними. Відправник вибирає випадкове ціле число X, $1<X\leq (P-1)$, та обчислює

$Y = G^X \bmod P$.

Число Y є відкритим ключем, що використовується для перевірки підпису відправника. Число Y відкрито передається всім потенціальним одержувачам документів.

Число X є секретним ключем відправника для підписування документів

отправителя для подписывания документов и должно храниться в секрете.

Для того чтобы подписать сообщение M, сначала отправитель хэширует его с помощью хэш-функции h(·) в целое число m:
$m = h(M), 1 < m < (P-1)$

и генерирует случайное целое число K, $1< K< (P -1)$, такое, что K и (P-1) являются взаимно простыми. Затем отправитель вычисляет целое число a:
$a = G^K \bmod P$

и, применяя расширенный алгоритм Евклида, вычисляет с помощью секретного ключа X целое число b из уравнения

$m = X* a + K * b \pmod{(P-1)}$

Пара чисел (a, b) образует цифровую подпись S:
S = (a, b)
проставляемую под документом M.

Тройка чисел (M, a, b) передается получателю, в то время как пара чисел (X,K) держится в секрете.

После приема подписанного сообщения (M, a, b) получатель должен проверить, соответствует ли подпись S = (a, b) сообщению M. Для этого получатель сначала вычисляет по принятому сообщению M число

m = h(M),

т.е. хэширует принятое сообщение M.

Затем получатель вычисляет значение

$A = Y * a^b \pmod P$

и признает сообщение M подлинным, если, и только если

$A = G^m \pmod P$.

Иначе говоря, получатель проверяет справедливость соотношения

$Y^a a^b \pmod P = G^m \pmod P$.

Можно строго математически доказать, что последнее равенство будет выполняться тогда, и только тогда, когда подпись S = (a, b) под документом M получена с помощью именно того секретного ключа X, из которого был получен открытый ключ Y. Таким образом, можно надежно удостовериться, что отправителем сообщения M был обладатель именно данного секретного ключа X, не раскрывая при этом сам ключ, и что отправитель подписал именно этот конкретный документ M.

Следует отметить, что выполнение каждой подписи по методу Эль Гамаля требует нового значения K, причем это значение должно выбираться случайным образом. Если нарушитель раскроет когда-либо значение K, повторно используемое отправителем, то он сможет раскрыть секретный ключ X отправителя.

4.1 Пример

Выберем: числа P = 11, G = 2 и секретный ключ X = 8. Вычисляем значение открытого ключа:

$Y= G^X \bmod P =Y= 2^8 \bmod 11 = 3.$

Предположим, что исходное сообщение M характеризуется хэш-значением m = 5.

Для того чтобы вычислить цифровую подпись для сообщения M, имеющего хэш-значение m = 5, сначала выберем случайное целое число K=9. Убедимся, что числа K и (P-1) являются взаимно простыми.

Действительно, НОД(9, 10)= 1.

Далее вычисляем элементы a и b подписи:

$a = G^K \bmod P* 2^9 \bmod 11 = 6,$

элемент b определяем, используя расширенный алгоритм Евклида:

$m = X * a + K * b \pmod{(P-1)}.$

При m = 5, a=6, X=8, K=9, P=11 получаем
5 = (6 * 8 + 9 * b) (mod 10)
или 9 * b ≡ - 43 (mod10).

Решение: b = 3. Цифровая подпись представляет собой пару: a=6, b=3.

Далее отправитель передает подписанное сообщение. Приняв подписанное сообщение и открытый ключ Y=3, получатель вычисляет хэш-значение для сообщения M: m=5, а затем вычисляет два числа:
1) $Y^a a^b \pmod P = 3^6 * 6^3 \pmod{11} = 10 \pmod{11}$;
2) $G^m \pmod P = 2^5 \pmod{11} = 10 \pmod{11}$.

Так как эти два целых числа равны, принятое получателем сообщение признается подлинным.

ПРИМЕЧАНИЯ

Расширенный алгоритм Евклида

При заданных неотрицательных целых числах a и b этот алгоритм определяет вектор (u_1, u_2, u_3), такой, что

$$a * u_1 + b * u_2 = u_3 = НОД (a, b).$$

В процессе вычисления используются вспомогательные векторы (v_1, v_2, v_3), (t_1, t_2, t_3). Действия с векторами производятся таким образом, что в течение всего процесса вычисления выполняются соотношения

$$a * t_1 + b * t_2 = t_3, \quad a * u_1 + b * u_2 = u_3, \quad a * v_1 + b * v_2 = v_3.$$

Для вычисления обратной величины $a^{-1} \pmod n$ используется частный режим работы расширенного алгоритма Евклида, при котором
b = n, НОД(a,n) = 1, и этот алгоритм определяет вектор (u_1, u_2, u_3), такой, что

$$u_3 = 1, \quad a * u_1 + n * u_2 = НОД(a, n) = 1,$$
$$(a * u_1 + n * u_2) \bmod n \equiv a * u_1 \pmod n \equiv 1,$$
$$a^{-1} \pmod n \equiv u_1 \pmod n.$$

Шаги алгоритма:
1. Начальная установка.
Установить (u_1,u_2,u_3) := (0,1, n), (v_l, v_2, v_3) := (1, 0, a).

2. u_3 = 1?. Если u_3 = 1, то алгоритм заканчивается.

3. Разделить, вычесть.
Установить $q:=[u_3/v_3]$.
Затем установить
$(t_1, t_2, t_3) := (u_1,u_2,u_3) - (v_1, v_2, v_3) * q$,
$(u_1,u_2,u_3):= (v_1, v_2, v_3)$,
$(v_1, v_2, v_3) := (t_1, t_2, t_3)$.

Возвратиться к шагу 2.

5.1.1 Пример

Заданы модуль n=23 и число a = 5. Найти обратное число a^{-1} (mod 23), т.е. x=5-1(mod23).

Используя расширенный алгоритм Евклида, выполним вычисления, записывая результаты отдельных шагов в таблицу 5.1.

Таблица 5.1 – Шаги расширенного алгоритма Евклида

q	u_1	u_2	u_3	v_I	v_2	v_3
4 1 1	0 1 -4 5 -9	1 0 1 -1 2	n = 23 5 3 2 1	1 -4 5 -9	0 1 -1 2	a = 5 3 2 1

При $u_3 = 1$, $u_1 = -9$, $u_2 = 2$

$(a * u_1 + n * u_2) \bmod n = (5 * (-9) + 23 * 2) \bmod 23 = 5* (-9) \bmod 23 \equiv 1$,

$a^{-1} (\text{mod } n) = 5^{-1} (\text{mod } 23) = (-9) \bmod 23 = (-9 + 23) \bmod 23 = 14$.

Итак, $x = 5^{-1} (\text{mod } 23) = 14 (\text{mod } 23) = 14$.

Алгоритм быстрого возведения в степень

Алгоритм быстрого возведения в степень — алгоритм, предназначенный для возведения числа x в натуральную степень n за меньшее число умножений, чем это требуется в определении.

Алгоритм не всегда оптимален. Например, при n=15 требуется 6 умножений, хотя на самом деле возведение в 15-ую степень можно выполнить за 5 умножений.

5.2.1 Теоретические основы алгоритма

Пусть $m = (\overline{m_k m_{k-1} ... m_1 m_0})_2$ — двоичное представление степени *n*. Тогда $m = m_k \cdot 2^k + m_{k-1} \cdot 2^{k-1} + ... + m_1 \cdot 2 + m_0$, где $m_k = 1, m_i \in \{0, 1\}$и

$x^n = x^{((...((m_k \cdot 2 + m_{k-1}) \cdot 2 + m_{k-2}) \cdot 2 + ...) \cdot 2 + m_1) \cdot 2 + m_0} = ((...(((x^1)^2 \cdot x^{m_{k-2}})^2 ...)^2 \cdot x^{m_1})^2 \cdot x^{m_0}$

.

Таким образом, алгоритм быстрого возведения в степень сводится к мультипликативному аналогу схемы Горнера.

$$\begin{Bmatrix} s_k = x \\ s_i = s_{i+1} \cdot x^{m_i} \\ 0 \le i \le k-1 \end{Bmatrix}$$

5.2.2 Оценка сложности

Чтобы узнать, сколько умножений потребуется для возведения числа x в степень n алгоритмом быстрого возведения в степень, нужно произвести вычисления по следующей формуле: k = H + 2(E − 1), где H — количество нулей, а E — количество единиц в двоичной записи числа n.

Так, для возведения числа в сотую степень этим алгоритмом потребуется всего лишь 8 умножений.

Таким образом количество умножений равно O(*ln*n).

Вычисление степени числа a по модулю n a^x mod n можно выполнить как ряд умножений и делений. Существуют способы сделать это быстрее. Поскольку эти операции дистрибутивны, быстрее произвести возведение в степень как ряд последовательных умножений, выполняя каждый раз приведение по модулю. Это особенно заметно, если работать с длинными числами (200 бит и более).

5.2.3 Пример

Нужно вычислить a^8 mod n, не следует применять примитивный подход с выполнением семи перемножений и одного приведения по модулю громадного числа:

(a * a * a * a * a * a * a * a) mod n

Вместо этого выполняют три малых умножения и три малых приведения по модулю:

$((a^2 \bmod n)^2 \bmod n)^2 \bmod n$.

Тем же способом вычисляют
$a^{16} \bmod n = (((a^2 \bmod n)^2 \bmod n)^2 \bmod n)^2 \bmod n$.

Вычисление $a^x \bmod n$, где x не является степенью 2, лишь немного сложнее. Двоичная запись числа x позволяет представить число x как сумму степеней 2: $x = 25_{(10)} = 1\ 1\ 0\ 0\ 1_{(2)}$, поэтому $25 = 2^4 + 2^3 + 2^0$

Тогда $a^{25} \bmod n = (a * a^{24}) \bmod n = (a * a^8 * a^{16}) \bmod n =$
$= a * ((a^2)^2)^2 * (((a^2)^2)^2)^2 \bmod n = ((((a^2 * a)^2)^2)^2 * a) \bmod n$.

При разумном накоплении промежуточных результатов потребуется только шесть умножений:

$(((((((a^2 \bmod n) * a) \bmod n)^2 \bmod n)^2 \bmod n)^2 \bmod n) * a) \bmod n$

Этот метод уменьшает трудоемкость вычислений до 1,5xk операций в среднем, где k-длина числа в битах.

ВАРИАНТЫ ЗАДАНИЙ НА РГР

Задание №1

Сообщение S зашифровано по криптосистеме RSA с открытым ключом n и e, кроме того известна функция Эйлера $\varphi(n)$. Дешифруйте сообщение. Исходные данные по вариантам заданы в таблице 6.1.

Таблица 6.1 - Шифрование по криптосистеме RSA. Варианты заданий.

№ вар.	S	n	e	$\varphi(n)$
1	1721-301-619-1207	4187	977	4056
2	3890-2087-3110	5029	821	4876
3	0304-0903-3376-3508	4171	853	4032
4	2531-2930-732-2260	3589	713	3456
5	1981-276-1106	3599	703	3480
6	3890-2087-3110	5029	821	4876
7	5229-3908-1176-4103	5429	2099	5280
8	208-1497-793-96	2173	113	2080
9	3527-3312-542-1208	4189	911	4060

10	1499-3601-3292-2902	4453	937	4320
11	2294-4193-3345-852	4757	941	4620
12	1093-3587-2558-3981	5063	967	4920
13	782-1771-1476-4121	5029	887	4876
14	330-4126-3160-1828	5293	997	5148
15	1945-2991-286-1986	4717	827	4576
16	1149-444-3729-373	4187	821	4056
17	1721-301-619-1207	4187	977	4056
18	3890-2087-3110	5029	821	4876
19	0304-0903-3376-3508	4171	853	4032
20	2531-2930-732-2260	3589	713	3456
21	1981-276-1106	3599	703	3480
22	3890-2087-3110	5029	821	4876
23	5229-3908-1176-4103	5429	2099	5280
24	208-1497-793-96	2173	113	2080
25	3527-3312-542-1208	4189	911	4060
26	1499-3601-3292-2902	4453	937	4320
27	2294-4193-3345-852	4757	941	4620
28	1093-3587-2558-3981	5063	967	4920
29	782-1771-1476-4121	5029	887	4876
30	330-4126-3160-1828	5293	997	5148

Примечание 1. В задании для преобразования слов, состоящих из букв русского алфавита (33 буквы) и пробелов, используются соответствия между символами и числами. Пробел заменяется на 1, буква А – на 2, буква Б – на 3 и т. д., буква Я заменяется на 34. Соответствующие числа, записанные в обратном порядке, образуют числовое представление открытого текста. Например, слову ДОМ будет соответствовать число 151706.

Таблица 6.2 – Алфавит

<пробел>	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Ь	Э	Ю	Я											
31	32	33	34											

Примечание 2. Для вычисления секретного ключа d рекомендуется использовать расширенный алгоритм Евклида, а для возведения чисел в степень по модулю n целесообразно использовать алгоритм быстрого возведения в степень.

Задание №2

Провести идентификацию для абонента А абонентом В используя систему идентификации Гиллоу-Куинскуотера. Исходные данные по вариантам заданы в таблице 6.3

Таблица 6.3 -Система идентификации Гиллоу-Куинскуотера. Варианты заданий.

№ вар.	n, (p,q)	G	v	r	d
1	1909,(23,83)	97	19	100	33
2	1921,(17,113)	101	43	85	13
3	4453,(73,61)	53	17	15	21
4	4717,(53,89)	71	29	15	23
5	4171,(43,97)	67	31	8	12
6	5353,(53,101)	73	37	24	14
7	3547	71	7	11	8
8	2953	113	19	17	10
9	4187,(53,79)	47	23	17	16
10	4189,(71,59)	61	13	15	18
11	4453,(73,61)	53	17	22	21
12	4757,(67,71)	59	19	13	7
13	5063,(61,83)	41	7	19	9
14	5029,(47,107)	43	11	7	8
15	5293,(79,67)	37	5	9	10
16	4717,(53,89)	71	29	21	23
17	1909,(23,83)	97	19	100	33
18	1921,(17,113)	101	43	85	13
19	4453,(73,61)	53	17	15	21
20	4717,(53,89)	71	29	15	23
21	3547	71	7	11	8
22	2953	113	19	17	10
23	4187,(53,79)	47	23	17	16
24	4189,(71,59)	61	13	15	18
25	4453,(73,61)	53	17	22	21
26	4757,(67,71)	59	19	13	7

27	5063,(61,83)	41	7	19	9
28	5029,(47,107)	43	11	7	8
29	5293,(79,67)	37	5	9	10
30	4717,(53,89)	71	29	21	23

Примечание 3. J вычисляется из сравнения $J \cdot G^{v} \equiv 1 \bmod n$ при помощи расширенного алгоритм Евклида.

Задание №3

Выполнить разделение секрета для сети из 4-х абонентов по схеме разделения секрета Шамира. Восстановить секрет для 2 групп абонентов. Исходные данные по вариантам заданы в таблице 6.4. Поле Fp состоит из элементов {0, 1, 2, 3, 4}. Секрет S=a0.

Таблица 6.4 - Схема разделения секрета Шамира. Варианты заданий.

№ вар.	Многочлен f(x)	Группы абонентов для восстановления секрета	
1	$f(x)=2x^2+3x+2$	3-ий, 4-ый, 1-ый абон.	2-ой, 3-ий, 1-ый абон.
2	$f(x)=3x^2+2x+3$	1-ый, 2-ой, 3-ий абон.	1-ый, 3-ий, 4-ый абон.
3	$f(x)=3x^2+4x+1$	1-ый, 2-ой, 3-ий абон.	2-ой, 3-ий, 4-ый абон.
4	$f(x)=2x^2+2x+4$	1-ый, 3-ий, 4-ый абон.	1-ый, 2-ой, 3-ий абон.
5	$f(x)=x^2+3x+2$	3-ий, 4-ый, 1-ый абон.	2-ой, 3-ий, 1-ый абон.
6	$f(x)=x^2+2x+3$	1-ый, 2-ой, 3-ий абон.	1-ый, 3-ий, 4-ый абон.
7	$f(x)=x^2+4x+1$	1-ый, 2-ой, 3-ий абон.	2-ой, 3-ий, 4-ый абон.
8	$f(x)=x^2+2x+4$	1-ый, 3-ий, 4-ый абон.	1-ый, 2-ой, 3-ий абон.
9	$f(x)=2x^2+3x+2$	3-ий, 4-ый, 1-ый абон.	2-ой, 3-ий, 1-ый абон.
10	$f(x)=3x^2+2x+3$	1-ый, 2-ой, 3-ий абон.	1-ый, 3-ий, 4-ый абон.
11	$f(x)=3x^2+4x+1$	1-ый, 2-ой, 3-ий абон.	2-ой, 3-ий, 4-ый абон.
12	$f(x)=2x^2+2x+4$	1-ый, 3-ий, 4-ый абон.	1-ый, 2-ой, 3-ий абон.
13	$f(x)=x^2+3x+2$	3-ий, 4-ый, 1-ый абон.	2-ой, 3-ий, 1-ый абон.
14	$f(x)=x^2+2x+3$	1-ый, 2-ой, 3-ий абон.	1-ый, 3-ий, 4-ый абон.
15	$f(x)=x^2+4x+1$	1-ый, 2-ой, 3-ий абон.	2-ой, 3-ий, 4-ый абон.
16	$f(x)=x^2+2x+4$	1-ый, 3-ий, 4-ый абон.	1-ый, 2-ой, 3-ий абон.
17	$f(x)=2x^2+3x+2$	3-ий, 4-ый, 1-ый абон.	2-ой, 3-ий, 1-ый абон.
18	$f(x)=3x^2+2x+3$	1-ый, 2-ой, 3-ий абон.	1-ый, 3-ий, 4-ый абон.
19	$f(x)=3x^2+4x+1$	1-ый, 2-ой, 3-ий абон.	2-ой, 3-ий, 4-ый абон.
20	$f(x)=2x^2+2x+4$	1-ый, 3-ий, 4-ый абон.	1-ый, 2-ой, 3-ий абон.
21	$f(x)=x^2+3x+2$	3-ий, 4-ый, 1-ый абон.	2-ой, 3-ий, 1-ый абон.
22	$f(x)=x^2+2x+3$	1-ый, 2-ой, 3-ий абон.	1-ый, 3-ий, 4-ый абон.
23	$f(x)=x^2+4x+1$	1-ый, 2-ой, 3-ий абон.	2-ой, 3-ий, 4-ый абон.
24	$f(x)=x^2+2x+4$	1-ый, 3-ий, 4-ый абон.	1-ый, 2-ой, 3-ий абон.

25	$f(x)=2x^2+3x+2$	3-ий, 4-ый, 1-ый абон.	2-ой, 3-ий, 1-ый абон.
26	$f(x)=3x^2+2x+3$	1-ый, 2-ой, 3-ий абон.	1-ый, 3-ий, 4-ый абон.
27	$f(x)=3x^2+4x+1$	1-ый, 2-ой, 3-ий абон.	2-ой, 3-ий, 4-ый абон.
28	$f(x)=2x^2+2x+4$	1-ый, 3-ий, 4-ый абон.	1-ый, 2-ой, 3-ий абон.
29	$f(x)=x^2+3x+2$	3-ий, 4-ый, 1-ый абон.	2-ой, 3-ий, 1-ый абон.
30	$f(x)=x^2+2x+3$	1-ый, 2-ой, 3-ий абон.	1-ый, 3-ий, 4-ый абон.

Задание №4

Используя схему электронной цифровой подписи Эль-Гамаля подпишите и верифицируйте сообщение M. Исходные данные заданы в таблице 6.5.

Таблица 6.5 - Схема ЭЦП Эль-Гамаля. Варианты заданий.

№ вар.	p	q	x	m=h(M)	k
1	1409	983	67	999	443
2	1423	991	71	800	449
3	1759	953	92	1995	757
4	1451	1031	72	1033	823
5	1447	983	87	1741	907
6	2707	947	81	1021	997
7	833	2477	301	1097	163
8	997	1499	83	1329	617
9	1399	977	64	1003	439
10	2053	1231	89	1995	919
11	1759	953	92	1027	757
12	2243	1103	81	2091	929
13	1979	991	73	1871	661
14	1381	857	57	1121	809
15	1667	1009	71	1539	431
16	1523	967	86	1329	797
17	1409	983	67	999	443
18	1423	991	71	800	449
19	1759	953	92	1995	757
20	1451	1031	72	1033	823
21	1447	983	87	1741	907
22	2707	947	81	1021	997
23	833	2477	301	1097	163

24	997	1499	83	1329	617
25	1399	977	64	1003	439
26	2053	1231	89	1995	919
27	1759	953	92	1027	757
28	2243	1103	81	2091	929
29	1979	991	73	1871	661
30	1381	857	57	1121	809

ПРИМЕРЫ РЕШЕНИЯ ЗАДАНИЙ

Задание №1

Сообщение 1617-1273-1187 зашифровано по криптосистеме RSA с открытым ключом n=4841 и e=947, кроме того известна функция Эйлера $\varphi(n) = 4692$. Дешифруйте сообщение.

7.1.1 Решение

1. Зная открытый ключ e и функцию Эйлера, $\varphi(n)$ найдем секретный ключ d. Для этого решим сравнение $ed \equiv 1 \bmod \varphi(n)$ с помощью расширенного алгоритм Евклида.

Таблица 7.1 – Поиск d

e	947
n	4841
fi	4692

d
4583

Таблица 7.2 – Использование расширенного алгоритма Евклида

q	u1	u2	u3	v1	v2	v3
-	0	1	4692	1	0	947
4	1	0	947	-4	1	904
1	-4	1	904	5	-1	43
21	5	-1	43	-109	22	1
43	-109	22	1			

В результате получим d=4583.

Таблица 7.3 – Разложение на множители

15	14	13	12	11	10	9	8	7	6	5
32768	16384	8192	4096	2048	1024	512	256	128	64	32
0	0	0	1	0	0	0	1	1	1	1
0	0	0	4096	0	0	0	256	128	64	32

4	3	2	1	0
16	8	4	2	1
0	0	1	1	1
0	0	4	2	1

2. Теперь можно дешифровать сообщение, используя формулу $M = S^d \bmod n$. Получим

Таблица 7.4 - Возведение в степень

Расчет степени	ч1	ч2	ч3
4583	1918	1511	1907
4582	3480	3515	907
2291	4129	2617	3691
2290	2691	196	570
1145	164	2458	3233
1144	3922	4398	4754
572	1003	4407	3918
286	1884	3202	3298
143	3053	930	1095
142	1400	632	1983
71	2036	2037	2729
70	1579	1884	247
35	3162	1788	1763
34	2885	1340	1694
17	825	2471	665
16	2668	888	3459
8	2074	2274	551
4	1259	2136	3393
2	549	3635	238
1	1617	1273	1187

$1617^{4583} \bmod 4841 = 1918$;

$1273^{4583} \bmod 4841 = 1511$;
$1187^{4583} \bmod 4841 = 1907$.

Таким образом, сообщение имеет вид 1918-1511-1907. Заменим числа буквами и получим *рп-ми-ре*. Переставим буквы (см. примечание 1 к заданию №1) и получим открытый текст: *пример*.

3. Ответ: ПРИМЕР.

Задание №2

Провести идентификацию для абонента А абонентом В используя систему идентификации Гиллоу-Куинскуотера.

n=4841, G=61, v=5.

7.2.1. Решение

1. Абоненту А необходимо вычислить открытый ключ J. Вначале вычисляется $G^v \bmod n = 61^5 \bmod 4841 = 1554$. Решив сравнение $J \cdot 1554 \equiv 1 \bmod 4841$, находим J=4246.

2. Теперь можно выполнять протокол:

Таблица 7.5 – Протокол идентификации Гиллоу-Куинскуотера

Шаг	Абонент А		Абонент В	
	Действие	Параметры	Действие	Параметры
1	Отправил	n=4841, v=5, J=4246	Получил	n=4841, v=5, J=4246
2	Выбор r $(1 < r \le n-1)$; вычисление T	r=7; $T = 7^5 \bmod 4841 = = 2284$		
3	Отправил	T=2284	Получил	T=2284
4			Выбор и отправление d $(1 < d \le n-1)$	d=6
5	Получил	d=6		
6	Вычисление и отправка D	$D = 7 \cdot 61^6 \bmod 4841 = = 341$	Получил	D=341

Продолжение таблицы 7.5

7			Вычисление T' и сравнение с T	$T' = 341^5 \times$ $\times 4246^6 \bmod 4841 =$ $= 2284$

3. Ответ: поскольку сравнение $T \equiv T' \bmod n \Rightarrow 2284 \equiv 2284 \bmod 4841$ выполняется, то абонент А прошел идентификацию успешно.

Задание №3

Пример схемы разделения секрета для сети из 4-х абонентов. Пусть характеристика конечного поля p=5. Поле Fp состоит из элементов {0, 1, 2, 3, 4}. Пусть многочлен f(x) имеет вид:

$f(x)=4x^2+3x+2$.

Для формирования многочлена f(x) были выбраны три элемента поля: a0=2, a1=3, a2=4.

В данном случае m=2.

7.3.1. Решение

Для восстановления секрета в будущем потребуется объединение трех любых абонентов сети. Секрет S=a0=2. Вычисляем части этого секрета:
s1=f(r1)=f(1)=(2+3+4)mod 5=4,
s2=f(r2)=f(2)=(2+6+16)mod 5=4,
s3=f(r3)=f(3)=(2+9+36)mod 5=2,
s4=f(r4)=f(4)=(2+12+64)mod 5=4.

Каждый абонент получает пару чисел, элемент поля ri и часть секрета si, i=[1,n]:

1-ый абонент – (1,4);

2-ой абонент – (2,4);

3-ий абонент – (3,2);

4-ый абонент – (4,3).

Пусть для восстановления секрета объединяются 3-ий, 4-ый и 1-ый абоненты. Тогда

$$S = \sum_{j=0}^{m} s_j \prod_{j \neq k} \frac{r_j}{r_j - r_k} = \sum_{j=0}^{2} s_j \prod_{j \neq k} \frac{r_j}{r_j - r_k}$$

В данном случае имеем:

r0=3, s0=2;
r1=4, s1=3;
r2=1, s2=4.

Вычисляем значение секрета:

$$S = s_0 \frac{r_1 r_2}{(r_0 - r_1)(r_0 - r_2)} + s_1 \frac{r_0 r_2}{(r_1 - r_0)(r_1 - r_2)} + s_2 \frac{r_0 r_1}{(r_2 - r_0)(r_2 - r_1)} =$$

$$= (2 \cdot \frac{4 \cdot 1}{(3-4)(3-1)} + 23 \cdot \frac{3 \cdot 1}{(4-3)(4-1)} + 4 \cdot \frac{3 \cdot 4}{(1-3)(1-4)}) \bmod 5 = (-4 + 3 + 8) \bmod 5 = 2$$

Для объединения 3-го, 4-го и 1-го абонентов значение секрета S=2 было восстановлено правильно. Аналогичным образом секрет может восстановить любая группа из трех абонентов.
Меняться будут только rj и sj.

Задание №4

Используя схему электронной цифровой подписи Эль-Гамаля подписать и верифицировать сообщение M.

q=947, p=2083, x=64, m=h(M)=1031,k=1009.

7.4.1. Решение

1. Генерация ключей. x - секретный ключ. Вычисляем открытый ключ y:

$$y = q^x = 947^{64} \bmod 2083 = 1449$$

2. Подпись сообщения. Для подписи сообщения необходимо вычислить два числа a и b:

$$a = q^k = 947^{1009} \bmod 2083 = 1954;$$

$$b = k^{-1}(m - a * x) \bmod (p-1).$$

Для вычисления b необходимо знать $k^{-1} \bmod (p-1)$. Это число можно найти с помощью расширенного алгоритма Евклида. Из соотношения $k \cdot k^{-1} \equiv 1 \bmod (p-1)$ находим k^{-1}=553 (k и p известны). Тогда

b=553(1031-1954*64)mod(2083-1)=1501.

Пара чисел (a, b) является значением цифровой подписи для сообщения.

3. Верификация. Необходимо вычислить V= $y^a a^b \bmod p$ и W= $q^m \bmod p$, а потом сравнить полученные значения. Если V=W, то подпись признается подлинной.

$$V = 1449^{1954} 1954^{1501} \bmod 2083 = 1890$$

$$W = 947^{1031} \bmod 2083 = 1890$$

Поскольку V=W, то подпись подлинная.

ЛИТЕРАТУРА

1. Використання алгоритмів криптосистем з відкритим ключем. Методичні вказівки до розрахунково-графічної роботи з дисципліни “Захист інформації в комп”ютерних системах та мережах” для студентів напряму підготовки 0915 - “Комп’ютерна інженерія” / Укладачі Соломаха В.В., Павловський В.І., Верьовко О.В. – Чернігів: ЧДТУ, 2010. - 41 с. Рос. мовою.
2. Ю. В. Романец, П.А.Тимофеев, В.Ф.Шаньгин. Защита информации в компьютерных системах и сетях. – М.: Радио и связь, 1999.
3. В. Столингс. Криптография и защита сетей. – М., СПб, К.: Вильямс, 2001.
4. Д. П. Зегжда, А. М. Ивашко. Основы безопасности информационных систем. – М.: Горячая линия – Телеком, 2000.
5. А. В. Домашев и др. Программирование алгоритмов защиты информации. – М.: Нолидж, 2000.
6. В. В. Ященко. Введение в криптографию. – СПб.: Питер, 2001.
7. В. В. Мельников. Защита информации в компьютерных системах. – М.: Финансы и статистика, 1997.
8. К. П. Исагулиев. Справочник по криптологии. – М.: Новое знание, 2004.
9. В. О. Осипян, К.В. Осипян. Криптография в упражнениях и задачах. – М.: Гелиос АРВ, 2004.