МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
ФАКУЛЬТЕТ ЭЛЕКТРОННО-ИНФОРМАЦИОННЫХ СИСТЕМ
Кафедра интеллектуальных информационных технологий

Отчет
по дисциплине
«Современные методы защиты компьютерных систем»
по лабораторной работе № 3
«Атака на алгоритм шифрования RSA»

Выполнила:
студентка 4
курса группы
ИИ-22
Сокол С.М.
Проверила:
Хацкевич А.С.

Брест 2024

Цель: изучить атаку на алгоритм шифрования RSA посредством Китайской теоремы об остатках.

Постановка задачи: Разработать приложение для декодирования зашифрованного текста, используя Китайскую теорему об остатках.

4	89318473363897	2227661	3403106899606 2674690010117 7 67769260919924 77873792354218 15782947730235 15100267747684 28877721728826 62898555111378 4989704651236 55293402838380 4108112294245 849226996417 2

Ход работы: Была написана программа на языке Python, для реализации необходимых требований

Код:

```
import math
    def if_perfect_square(n):
        sqrt_n = int(math.isqrt(n))

        if sqrt_n * sqrt_n == n:
            return sqrt_n
        else:
            return None
    if __name__ == "__main__":
        #Мои данные(спасибо P_P) doc - 4, pdf - 4:
                                                                                                N =
126950797675950874303096925604851105399222699041017772196733549462723607342661210782274795904266336243185243286021793217252014033077747061132255750621710014623309161858309621921048730559780051605895782074804738598488339822530135260363201631576483938149741164655458142155729683330861142337726276708972349891552220378399667486626167723038633616467511230661439976946056099092810425616590416424337756433550640944307059384349454291846980118447544161276995570521814479141157494662581395163154846523155171242347940506510433968793183846538396643855412382565195358343435348930837099777286383013660611770082127619961852206006609
        e = 11560063
                                                                                                C =
137392820091395484131751509524492762658849652040496520404181628872684883659355096555706221177575869353617006630824661552058696088366783392304010608554722685080817108242428286970396703238196767845633080618950857003814381082703712455256339604615736559324314034765137260179847605959970047132239235834244051982294070117016593070674370842827048602775265470105491633573282258752226346513698852739958442599617061379529941351943179948951100065392325867129254816175603925826025583812438751150942951195169396436236798150460471839142877134173029078454188944958423354228755042496642269810410143209189969359456088679379822236472 84

        #1 - вычисляй n=sqrt(N)
        n = int(math.isqrt(N))
        print(f"n = {n}\n")

        #2 - t1 ^ 2, t1 = n + 1
        t1 = n + 1#метод факторизации

        #4 - мозг считай такие t1 и w1, пока w1 не будет квадратом целого числа
        while True:#и тут
            t1_pow = pow(t1, 2)

            #3 - w1 = (t1 ^ 2) - N
            w1 = t1_pow - N
```

```
    w1 = if_perfect_square(w1)
    if w1 == None:
        t1 += 1
    else:
        break

print(f"Финальные значения t1 и w1:\n t1 = {t1}\n w1 = {w1}\n")

w1_sqrt = int(math.isqrt(w1))
print(f"корень w1 = {w1_sqrt}\n")

#5 - вычисляй p = t1 + w1_sqrt, q = t1 - w1_sqrt
p = t1 + w1_sqrt
print(f"p = {p}\n")

q = t1 - w1_sqrt
print(f"q = {q}\n")

#6 - высчитывай Phi(N) = (p - 1)(q - 1)
phi = (p - 1) * (q - 1)
print(f"Phi(N) = {phi}\n")

#7 - высчитывай d как обратный экспоненте e
d = pow(e, -1, phi)
print(f"d = {d}\n")

#8 - дешифруй сообщение D = C ^ d mod N
D = pow(C, d, N)
print(f"\"Дешифрованное\" сообщение D = {D}\n")
```

```
n = 3563015544113593595691704489446221191559323597881124659134228133500856180421276301655505091740393497991554741105152952648730572507884356043280777223247271307992501470098106707238546375357962580273613715295196994022493579239854216025941925153247286865700884406999695059428385067642881732919161617115347423752475247
Финальные значения t1 и w1:
 t1 = 3563015544113593595691704489446221191559323597881124659134228133500856180421276301655505091740393497991554741105152952648730572507884356043280777223247271307992501470098106707238546375357962580273613715295196994022493579239854216025941925153247286865700884406999695059428385067642881732919161617115347423752475247
 w1 = 3906519585718230671525974851079451329363016234935668752130981459076955688221598734164837242260089913930213342971115218610894368869722930302373050048408362
0

корень w1 = 19764917368201240905298101704818114810483671087044236355234049451954015227233
4

p = 3563015544113593595691704489446221191559323597881124659134228133500856180421276301655505091740393497991554741105152952648730572507884356043280777223247271307992501470098106707238546375357962580273613715295196994022493579239854216027918416890067410956230694577481506540476752176347305368442566562310748946475810
q = 3563015544113593595691704489446221191559323597881124659134228133500856180421276301655505091740393497991554741105152952648730572507884356043280777223247271307992501470098106707238546375357962580273613715295196994022493579239854216029654334164271627751710742365178835783800179589384580973957566719199459010291
3

Phi(N) = 1269507976759508743030969256048511053992226990410177721967335494627236073426612107822747959042663362431852432860217932172520140330777470611322557506217100146233091618583096219210487305597800516058957820748047385984883398225301352603632016315764839381497411646554581421557296833308611423377262767089723498915530338648182047815665537521888888304509213196476070435584209363308266737484670918086068772957304598255205981797588396499114608664945820949922691828776544961130198291258613883130769751666169170439945252376908986974182482483612781922414291529357703193235682477000779736822365382361175903731019099067696150171752960

d = 9096400079317958194636035531073952815933159857688737651019834194333728159763448753552220981616777563643089615203791172991332514725613344678443691689915124236751365004982538621926460162158625428928793136247883337773167211711342504697132405255535602817687508138745187069135892639160408220512843582405736142096890014678899432621803390532621576673626134879085381099474331159540510605013050255119756502533935714985719008530602142293875830322435812395402140971190005921702180128843665158170265560639446487349344375099849374997782644944962749326419521548190597212036559739053357805339148153864888002724383973515809394190759674
"Дешифрованное" сообщение D = 2635885400891894778705326554966042327130873808923744643199398719850385567935194182237907623663956559990122241955245688208606103545627697242591741273739930804839276957801284990284333409643430323775766802318312862483740059726193090446524813710102927237005232660931941623998595209940343489622907121357789137347434661552392841990198065634812801989589711864843586674585152339357399948758773592921149588938309175982993742832831044856043673815042787004327600394271926415900441449101758527229707851359414176329475679594102262187637691079776900818384291267543529160486671777299530541836420518650955826494917294518326037439381
```

Вывод: изучили и реализовали атаку на алгоритм шифрования RSA посредством Китайской теоремы об остатках.