ATTACK ON RSA PUBLIC KEY SYSTEM

By

Anumeha Shah
Veena Reddy Reguri
Neha Rajkumar

Types Of Attack Implemented

1. Common Modulus Attack
2. Chinese Remainder Theorem
3. Timing Attack

Brief Overview Of RSA

- ➔ Inventors, Rivest, Shamir, and Adleman
- ➔ Let p and q be two large prime numbers.
- ➔ N = pq be the modulus
- ➔ Choose exponent e such that e is relatively prime to (p-1)(q-1)
- ➔ Find d such that ed = 1 mod(p-1)(q-1)
- ➔ Public Key is (N, e)
- ➔ Private key is (N, d)
- ➔ $C = M^e \bmod N$
- ➔ $M = C^d \bmod N$
- ➔ Factoring the modulus can break RSA but factoring is hard and no body has found efficient solution

Problem Specification

- ➔ Alice wants to encrypt some messages for her party. Instead of using different N for each messages she decided on using the same N for all the messages.
- ➔ She encrypts the message using RSA public key system using (N, e1), (N, e2)
- ➔ calculates
- ➔ $C1 = M^{e1} \bmod N$,
- ➔ $C2 = M^{e2} \bmod N$
- ➔ Looks good and secure as Trudy(an attacker) does not know d1 and d2 to decrypt the ciphertexts.
- ➔ However it is not secure. Trudy can get M based on extended euclidean theorem,

Common Modulus Attack

- ➔ Extended Euclidean Theorem states that if gcd(m,n) = r then there exists integers a and b such that a*m + b*n = r.
- ➔ Similarly if we can choose e1 and e2 such that gcd(e1,e2) = 1 then there exists integer a and b such that a*e1 + b*e2 = 1
- ➔ that is we need have two no e1 and e2 and e1 and e2 should be relatively prime to each other only then we can find a and b such that e1*a + e2*b = 1
- ➔ Then Trudy can find the message M by calculating
- ➔ C^a mod N multiply C^b mod N
- ➔ which is equivalent to M^{e1a} mod N multiply M^{e2b} mod N
- ➔ $M^{e1a + e2b}$ mod N = M^1 mod N = M

Implementation

- We are using BigInteger as it is not possible with Integer.
- extended euclidean theorem.
- We are converting the plain text to base 64 encoded string and then converting the string to hexadecimal and from hexadecimal to decimal.
- We have also Implemented RSA.
- Java as our implementation language.

- N=402394248802762560784459411647796431108620322919897426002417858465984510150839043308712123310510922610690378085519407742502585978563438101321191019034005392771936629869360205383247721026151449660543966528254014636648532640397857580791648563954248342700568953634713286153354659774351731627683020456167612375777
- M = any value
- e1 and e2 such that gcd(e1, e2) = 1
- We should never use the same N for encrypting the message using RSA and even if we use same N. e1 and e2 should not be relatively prime.

ALICE BIRTHDAY PARTY-2

PROBLEM SPECIFICATION:

•Alice has learned from last year's mistake and no longer sends encrypted emails to recipients who use the same RSA modulus N.

•This year, she invites her friends Bob, Bertha, and Birte to her birthday party.

•Thus, Alice sends the same message to all three of her friends, encrypted with their respective public RSA keys.

•Is Trudy again able to decrypt the cipher texts?

INPUT PARAMETERS

- N1=
5147451670252223874341323771370567159547507298071514479298942896955872857938890999785369044944
55862473045694392353612260528582074521711173586408238050587426102676946559631584966824570308145
2047808798727647904141791488099702631575692170683102622471798376397440600292225038412176681344
166204027842724877162681931
- N2=
3324595527999155443560226416054481376170799213918322225578929498080609530284494223282814136299
1233505144074495545501085101230891829454976500548012106169771144708761532786078970824623515691
2421474047484838827777697938563515420810650393553528058831317409340577149233554235346445890238
642955390137465511286414033
- N3=
6657019121622430690596537816692308054734574277675143232627628917711223523287066954091037138643
8483343743864812021761599076522036574501373924602220359323478533817896380546364386939898611943
1772931646042972240277833431035018628949924813463553419243108837309078316455504749755062865258
063926243606206806549969161
- e = 3

CHINESE REMAINDER THEOREM

- Used to speed up modulo computations
- Working modulo a product of numbers

–eg. mod $M = m_1 m_2 .. m_k$

- Chinese Remainder theorem lets us work in each moduli m_i separately
- To compute (A mod M) can firstly compute all (a_i mod m_i) separately and then combine results to get answer using:

$$A \equiv \left(\sum_{i=1}^{k} a_i c_i \right) \bmod M \qquad c_i = M_i \times \left(M_i^{-1} \bmod m_i \right) \quad \text{for } 1 \le i \le k$$

CUBE ROOT ATTACK

- Use **$e = 3$** for all users (but not same N or d)

+Public key operations only require 2 multiplies

–Private key operations remain expensive

-If $M < N^{1/3}$ then $C = M^e = M^3$ and **cube root attack**

-For any M, if C_1, C_2, C_3 sent to 3 users, cube root attack works (uses Chinese Remainder Theorem)

Can prevent cube root attack by padding message with random bits

CRYPTANALYSIS APPROACH

- Cipher texts are evaluated as

C1 = M1^3 Mod N1

C2 = M2^3 Mod N2

C3 = M3^3 Mod N3

- On the other hand *M^3 = C1 Mod N1 ; M^3 = C2 Mod N2 ; M^3 = C3 Mod N3*
- Actual decryption:

M^3 = C1.(N2.N3).((N2N3)-1Mod N1) + C2.(N1.N3).((N1N3)-1Mod N2) + C3.(N1.N2).((N1N2)-1Mod N3) (Mod N1.N2.N3)

- Finally evaluate M = (M^1/3)

IMPLEMENTATION STEPS

- Inputting Message: Plaintext
- Convert string to decimal format
- Encrypt the message with different modulus values N1, N2, N3 generating C1, C2, C3
- Evaluate

  M^3 = C1.(N2.N3).((N2N3)-1Mod N1) + C2.(N1.N3).((N1N3)-1Mod N2) + C3.(N1.N2).((N1N2)-1Mod N3) (Mod N1.N2.N3)

- Calculate cube root value of M
- M in decimal format is obtained
- Convert decimal to string

RSA Timing Attack

- Timing attacks exploit the fact that some computations in RSA take longer than others.
- By carefully measuring the time that an operation takes we can determine the RSA private key.
- Repeated squaring is used for computing modular exponentiation.

Repeated Squaring Algorithm

```
// Compute y = x^d (mod N),
// where d = d_0 d_1 d_2 ... d_n in binary, with d_0 = 1
s = x
for i = 1 to n
    s = s^2 (mod N)
    if d_i == 1 then
        s = s · x (mod N)
    end if
next i
return(s)
```

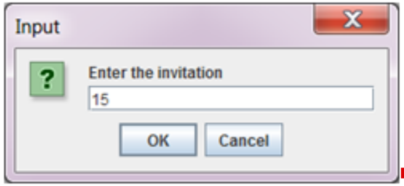
Kocher's Timing Attack

- Side channel attack.
- Choose random cipher texts Cj.
- Measure corresponding timing T(Cj), t~ denotes time at each iteration.

		Emulate 1010		Emulate 1001	
j	$T(C_j)$	$\tilde{t}_{0\ldots3}$	$T(C_j) - \tilde{t}_{0\ldots3}$	$\tilde{t}_{0\ldots3}$	$T(C_j) - \tilde{t}_{0\ldots3}$
0	12	5	7	7	5
1	11	5	6	4	7
2	12	6	6	7	5
3	13	8	5	6	7

Calculation

- Calculate the variance var(T(Cj) – t~0..3) for 1010 and 1001.
- Compare them.
- Smaller variance is the correct answer.



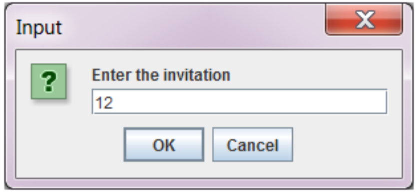
N=33

e=3

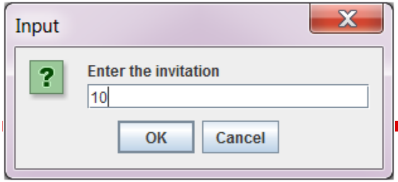
Trying to recover the value of d which is 7, 111(in binary)

Attacker choosing value of d as 4, 100

```
The user
Enter the value of N
33
Enter the value of e
3
the cipher text is9
Enter the value of d
7
The value of d in binary is:
1
1
1
The time required for decrypting the ciphertext: 9for the the user is: 373586

Attacker measuring the time
enter the cipher text
9
Enter the value of d
4
The value of d in binary is:
1
0
0
The time of iteration for the 0:th bit is: 26030
The time of iteration for the 1:th bit is: 13565
The time of iteration for the 2:th bit is: 21998
the time array[26030, 13565, 21998, 0, 0, 0, 0, 0, 0, 0, 0, 0]Which bit to check give 1 for bit **1** and 2 for bit **2**

1
[347556, 0, 0]
do u wish to continue??press 1 to continue1
```



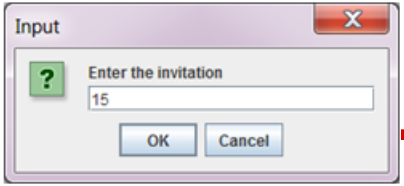

```
do u wish to continue??press 1 to continue1
The user
Enter the value of N
33
Enter the value of e
3
the cipher text is12
Enter the value of d
7
The value of d in binary is:
1
1
1
The time required for decrypting the ciphertext: 12for the the user is: 298429


Attacker measuring the time
enter the cipher text
12
Enter the value of d
4
The value of d in binary is:
1
0
0
The time of iteration for the 0:th bit is: 23464
The time of iteration for the 1:th bit is: 12832
The time of iteration for the 2:th bit is: 26030
the time array[23464, 12832, 26030, 0, 0, 0, 0, 0, 0, 0, 0, 0]Which bit to check give 1 for bit **1** and 2 for bit **2**

1
347556, 274965, 0]
do u wish to continue??press 1 to continue1
The user
```




```
do u wish to continue!!press 1 to continue1
The user
Enter the value of N
33
Enter the value of e
3
the cipher text is10
Enter the value of d
7
The value of d in binary is:
1
1
1
The time required for decrypting the ciphertext: 10for the the user is: 247102


Attacker measuring the time
enter the cipher text
10
Enter the value of d
4
The value of d in binary is:
1
0
0
The time of iteration for the 0:th bit is: 23464
The time of iteration for the 1:th bit is: 13931
The time of iteration for the 2:th bit is: 20164
the time array[23464, 13931, 20164, 0, 0, 0, 0, 0, 0, 0, 0, 0]Which bit to check give 1 for bit **1** and 2 for bit **2**

1
[347556, 274965, 223638]
do u wish to continue!!press 1 to continue3
Finished...checking for mean for the variables...
```

Displaying the Mean and Variance

```
1
[347556, 274965, 223638]
do u wish to continue??press 1 to continue3
Finished...checking for mean for the variables...
[347556, 274965, 223638]
the mean is 282053.0
the value of sum2 is 7.753194978E9
checking for variance for the variables...
[347556, 274965, 223638]
the variance is: 2.584398326E9
```



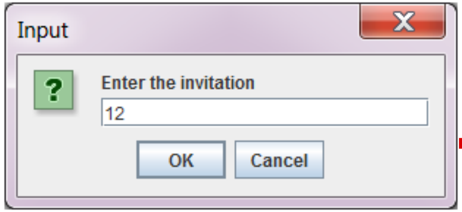
N=33

e=3

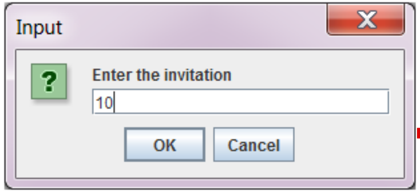
Trying to recover the value of d which is 7,111(in binary)

Attacker choosing value of d as 6, 110

```
The user
Enter the value of N
33
Enter the value of e
3
the cipher text is9
Enter the value of d
7
The value of d in binary is:
1
1
1
The time required for decrypting the ciphertext: 9for the the user is: 379085

Attacker measuring the time
enter the cipher text
9
Enter the value of d
6
The value of d in binary is:
1
1
0
The time of iteration for the 0:th bit is: 23097
The time of iteration for the 1:th bit is: 23097
The time of iteration for the 2:th bit is: 22364
the time array[23097, 23097, 22364, 0, 0, 0, 0, 0, 0, 0, 0, 0]Which bit to check give 1 for bit **1** and 2 for bit **2**
1
[355988, 0, 0]
do u wish to continue??press 1 to continue1
```




```
The user
Enter the value of N
33
Enter the value of e
3
the cipher text is12
Enter the value of d
7
The value of d in binary is:
1
1
1
The time required for decrypting the ciphertext: 12for the the user is: 298062

Attacker measuring the time
enter the cipher text
12
Enter the value of d
6
The value of d in binary is:
1
1
0
The time of iteration for the 0:th bit is: 23464
The time of iteration for the 1:th bit is: 22730
The time of iteration for the 2:th bit is: 22731
the time array[23464, 22730, 22731, 0, 0, 0, 0, 0, 0, 0, 0, 0]Which bit to check give 1 for bit **1** and 2 for bit **2**

1
[355988, 274598, 0]
do u wish to continue??press 1 to continue1
```




```
The user
Enter the value of N
33
Enter the value of e
3
the cipher text is10
Enter the value of d
7
The value of d in binary is:
1
1
1
The time required for decrypting the ciphertext: 10for the the user is: 276798

Attacker measuring the time
enter the cipher text
10
Enter the value of d
6
The value of d in binary is:
1
1
0
The time of iteration for the 0:th bit is: 28963
The time of iteration for the 1:th bit is: 21264
The time of iteration for the 2:th bit is: 22364
the time array[28963, 21264, 22364, 0, 0, 0, 0, 0, 0, 0, 0, 0]Which bit to check give 1 for bit **1** and 2 for bit **2**

1
[355988, 274598, 247835]
do u wish to continue??press 1 to continue3
```

Displaying the Mean and Variance

```
do u wish to continue??press 1 to continue3
Finished...checking for mean for the variables...
[355988, 274598, 247835]
the mean is 292807.0
the value of sum2 is 6.345887226E9
checking for variance for the variables...
[355988, 274598, 247835]
the variance is: 2.115295742E9
```

Comparison

- d=4, 100 in binary

  Variance : 2.584398326E9
- d=6, 110 in binary

  Variance : 2.115295742E9
- Variance for d=6 < Variance for d=4
- So 1st bit is 1
- d= 11_.

References

- Timing Attacks on RSA: Revealing Your Secrets through the Fourth Dimension.
- http://cs.sjsu.edu/~austin/cs265-spring14/PublicKeyCrypto.pdf
- http://crypto.stanford.edu/~dabo/papers/RSA-survey.pdf
- http://crypto.stackexchange.com/questions/3549/common-modulus-attack-on-rsa-when-the-2-public-exponents-differ-by-a-single-bit
- Chinese Remainder Theorem.
- http://en.wikipedia.org/wiki/Chinese_remainder_theorem
- http://www.math.tamu.edu/~jon.pitts/courses/2005c/470/supplements/chinese.pdf
- Cube Root Attack.
- www.cs.bilkent.edu.tr/~mustafa.battal/cs470/slides/cs470.RSA.**ppt**

Thank you !!!

Questions ???