CS 641 : ASSIGNMENT : PART 6 : SCHUTZSAS

A PREPRINT

Srinjay Kumar
170722

Abhyuday Pandey
170039

Kumar Shivam
170354

May 17, 2020

1 Path to the Cipher Text

The value for C and N were provided in the mail for the assignment.

2 Available Text

The values that were provided are :

- $N = 84364443735725034864402554533826279174703893439763343343863260342756678609216895093779263028809246505955647572176682669445270008816481771701417554768871285020442403001649254405058303439906229201909599348669565697534331652019516409514800265887388539283381053937433496994442146419682027649079704982600857517093$
- $C = 58851190819355714547275899558441715663746139847246075619270745338657007055698378740637742775361768899700888580870506626143183054430644488980265035567576103429384907413616436962850518672602785678969919273519645573749776196447636332298966685117524322225281592140131733198556453516193938714334555505817416432 99$
- The value of the exponent for the given question is 5.

3 RSA Algorithm

The algorithms for decryption and encryption are :

- The two different primes be p and q
- **Encryption** : $C = M^e mod\ N$
  - C : Ciphertext
  - M : Message to be encrypted
  - e : public key exponent such that:
    - $1 < e < \phi(N)$
    - $gcd(e, \phi(N)) = 1$
  - N : product of p and q
- **Decryption** : $M = C^d mod\ N$
  - d : private key exponent such that:
    - $de = 1 mod\ N$
- **Public Key** : N, e
- **Private Key** : p, q, d

4 Decryption

- We have the values of N, C and e.
- The values of N and C are quite large so it will be computationally difficult to determine the values of p and q.
- Without determining p and q, we cannot determine the value of d and therefore, we cannot decrypt the message.
- So, we need to use some other algorithm to decrypt the message.
- One of the observations is that the value of the exponent is quite small and it can be decrypted by using the Copperfield theorem.

5 Copperfield Theorem

Let N be an integer and f be a polynomial of degree δ. Given N and f , one can recover in polynomial time all x_0 such that $f(x_0)$ = 0 $mod\ N$ and $x_0 < N^{\frac{1}{\delta}}$.

6 Modeling the Problem

- We model our problem as follows $f(x)$ = $(a + x)^e\ mod\ N$. If $(a + x) < N^{\frac{1}{e}}$, we will find the required password as the root to the polynomial $(a + x)^e = 0$ in ring $\mathbb{Z}$.
- a is the padding added to x and a should be known beforehand.

7 Finding the value of x

7.1 Algorithm

[As Discussed in Class]

- The expression $(a + x)^5$ can be expanded as $a^5 + 5a^4x + 10a^3x^2 + 10a^2x^3 + 5ax^4 + x^5$
- We now need to find the value of x from the equation $a^5 - C + 5a^4x + 10a^3x^2 + 10a^2x^3 + 5ax^4 + x^5$ $\mod N$ = $P(x)$ = 0 $\mod N$
- We need an α such that $P(\alpha)$ = 0 $\mod N$, also we know that the value of α is less than the padding a.
- If we can show that the value of $||P(x)||$ does not exceed N, then we can simplify the problem to $P(x)$ = 0 in $\mathbb{Z}$. For proving this statement, we can use the concept of shortest vector as done in class for the exponent 3.
- The problem $||P(x)||$ = 0 can be solved by any numerical method. We later used binary search. This will be discussed in detail.
- The only task is to gather significant number of polynomials P_i such that $P_i(x) = 0 \mod N^j$ for all these polynomials.

7.2 Designing polynomials

Consider the following transformation. We will use a parameter α which we call the scale factor, $P(x) = 0 \mod N$ is surely the first polynomial we have.

We will scale $x \to \alpha x$. Consider the polynomials $P(\alpha x)^j N^{2-j}$ $j \in \{0, 1\}$ which is clearly 0 $\mod N^2$. So, we now have 2 polynomials and the dimension is still 5 as observed. However we should also keep in mind that volume of lattice does not increases much otherwise all our efforts will be futile.

We carefully select the polynomials $(\alpha x)^i N^{2-j} P(\alpha x)^j$ $j \in \{0, 1\}$ $i \in [0, 4]$ (as discussed in class) which are 0 $\mod N^2$. This will increase the dimension to 10. Let us see the volume of lattice formed.

The volume of the lattice can be visualized by the given matrix. The matrix is an upper triangular matrix, and therefore the volume can be given by the product of the diagonal elements.

$$\begin{bmatrix}
\alpha^9 N & . & . & . & . & . & . & . & . & . \\
0 & \alpha^8 f_1(a)N & . & . & . & . & . & . & . & . \\
0 & 0 & \alpha^7 f_2(a)N & . & . & . & . & . & . & . \\
0 & 0 & 0 & \alpha^6 f_3(a)N & . & . & . & . & . & . \\
0 & 0 & 0 & 0 & \alpha^5 f_4(a)N & . & . & . & . & . \\
0 & 0 & 0 & 0 & 0 & \alpha^4 N^2 & . & . & . & . \\
0 & 0 & 0 & 0 & 0 & 0 & \alpha^3 N^2 & . & . & . \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha^2 N^2 & . & . \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha^1 N^2 & . \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha^0 N^2
\end{bmatrix}$$

Ignoring the padding term as they have abysmal contribution $vol(\mathcal{L}) \sim N^{15}\alpha^{45}$.

7.2.1 Shortest Vector

- **Theorem** : Minkowski (1890) : $\lambda(\mathcal{L}) < \sqrt{n}\, vol(\mathcal{L})^{\frac{1}{n}}$ where n is the dimension of lattice.
- **Theorem** : Aytai (1992) : Finding shortest vector is NP-hard
- **Theorem** : Finding shortest vector within $\sqrt{2}$ is NP hard
- **Theorem** : Finding short vector within $2^{\frac{n}{2}}$ factor is in P

We know that shortest possible vector to get in this lattice is $\lambda(\mathcal{L}) \leq 2^5\sqrt{10}N^{1.5}\alpha^{4.5}$. It is easy to see that $\alpha \leq N^{0.11}$ is still possible (important to note that x is scaled and all polynomials are $0 \mod N^2$).

Accordingly, we set $\alpha = N^{0.1}$ (i.e around 100 digits) and expect to land safely. If this approach fails we would reconstruct the lattice with more dimensions.

We constructed the lattice and proceeded.

7.3 L^3 reduced bias algorithm

Since, we had quite some time at hand, we thought of implementing L^3 algorithm from scratch. The algorithm implemented following,

Algorithm 1: Reduced bias Algorithm

```
Compute GS-orthogonalisation of $b_1, b_2, ...., b_m$;
for $i = 2$ to $m$ do
    for $j = i - 1$ to $1$ do
        $b_i = b_i - \lfloor \frac{<b_i, b_j^*>}{||b_j||^2} \rfloor . b_j$
    end
end
If $\exists\, i$, $||b_i||^2 > \frac{4}{3}.||b^*_{i+1} + \mu_{i+1,i}.b^*_i||^2$, then swap $\{b_i, b_{i+1}\}$ and GOTO(1);
Output $\{b_1, b_2, ....b_m\}$;
```

However the algorithm failed to converge on our gigantic lattice. Even oplll an open-source library failed to converge.

We observed an efficient implementation of LLL in Sage (actually in fpylll which Sage uses) and use sage script to run LLL function.

7.4 Finding roots of obtained polynomial

After LLL algorithm, the polynomial obtained need to be searched for roots. We exploit a very beautiful property of odd degree polynomial (there is always a real root that can be searched using binary search on a reasonable interval).

We obtained the roots of polynomial by applying binary search on domain. Fortunately, we found a root of our interest.

8 Random Padding (a)

We tried several padding for the value of a. We also varied length of x from 0 to 96 in multiples of 8. Some of the random paddings were :

- "This door has RSA encryption with exponent 5 and the password is "
- "This door has RSA encryption"
- "This door has RSA encryption;"
- "This door has RSA encryption:"
- "This door has RSA encryption,"
- "This door has RSA encryption."
- "This door has Rivest Shamir Adleman encryption"
- "CS 641A"
- "Modern Cryptology"
- "Manindra sir"
- "This door has RSA encryption with exponent 5 and the password is :"
- "This door has RSA encryption with exponent 5 and the password is"
- "This door has RSA encryption with exponent 5 and the password is;"
- "This door has RSA encryption with exponent 5 and the password is,"
- "This door has RSA encryption with exponent 5 and the password is."
- "This door has RSA encryption with exponent 5 and the password is "

9 Solution

- "This door has RSA encryption with exponent 5 and the password is" and length 80 gave us a root which made less sense as it couldn't be converted to ASCII.
- The correct padding used is "This door has RSA encryption with exponent 5 and the password is ".
- The size of x comes out to be 72.
- Using this padding, we get the password to be "tkigrdrei".

10 References

- J.-S. Coron. Universite du Luxembourg Cryptography, Lecture Slides: Attacks against RSA, 2010.
- J. Dyer. Lattice reduction on low-exponent rsa, 2002.