EE720 Home Paper
Can we break RSA by finding the Local Inverse of Encryption/Decryption maps ?

Abheet Chaudhary (20D070002)
Deepankar Sehra (20D070024)
Sidharth Anand (20D070076)

25 November 2022

1 Brief description of the task

Let $F(x)$ be a mapping between finite fields. For an x it gives $y = F(x)$ in the same field. Consider a finite sequence

$$S(F, y) = \{y, F(y), F^{(2)}(y)..., F^{(2M-1)}(y)\}$$

If the sequence is sufficiently long then we can find the Local Inverse x of this finite sequence using Berlekamp-Massey Algorithm. This means if we only knew $S(F, y)$ then it is possible to find the x for which $F(x)$ maps to y, which is the first term of this sequence from which everything builds upon.

RSA cryptosystem is also based on two such equations which are a mapping within finite fields:

$$\text{Encryption map:} \quad y = E(x) \Rightarrow \quad c = x^e \mod n$$

$$\text{Decryption map:} \quad y = D(x) \Rightarrow \quad m = c^x \mod n$$

where $n = pq$ is a product of two primes p and q (both greater than 2) and e is a number which is invertible in mod $\phi(n)$ with d as its inverse.

The corresponding finite sequences are:

$$\begin{aligned} S(E, c) &\Rightarrow \{c, c^e \mod n, c^{e^2} \mod n, c^{e^3} \mod n...\} \\ S(D, m) &\Rightarrow \{m, c^m \mod n, c^{c^m \bmod n} \mod n...\} \end{aligned}$$

We can use these two sequences with multiple initial conditions i.e, different values of (p, q, e) and c and m, to try to find the local inverse.

If for certain (n, e) and c we find a local inverse x of $S(E, y)$ then this means we have found the plaintext x for which $E(x)$ outputs the ciphertext c.

If for certain (n, e) and m we found a local inverse x of $S(D, y)$ then this means we found d which is the inverse of e mod n

Our task is to use this sequence to break RSA under the constraints that p and q both have a binary length from 8-bit to 16-bit and the length of the finite sequence $S(F, y)$ is l^2 where l is the binary length of n.

2 Method

1. Start with coding the formula for F(for both funcctions E and D). These formulae accept an input y and create the corresponding finite sequence $S(F, y)$ as a list object in python. The size of these arrays is $M = l^2$.

2. Convert the integer values of that list into their binary form of length l. This step will transform $S(F, y)$ into a vector sequence of l-tuple vectors over $\mathbb{F}_2$(l is the binary length of n).

3. Using ith coordinate of each vector create i scalar sequences and use them to find and verify the minimal polynomial and Local Inverse. Berlekamp-Massey Algorithm was very helpful in this step.

4. Do this for 10,000-40,000 randomly chosen values of y. Correct Local Inverses will be found for some values of y. Find the frequency $\nu(F)$ of values of y for which Local Inverses are successfully found.

5. Do this for 5 different sets of parameters (p, q, e) for functions E and D. p, q are primes of minimum 8 bit length and upto 16-bit length.

After calculating n from p and q, in further steps only n and e are used. Other than this calculation we have not used p and q anywhere. This is done to simulate the challenges of finding factors of large n in real life.

To get a better understanding of how the chances of success diminish as we take larger and larger n we executed Step-5 multiple times and grouped together the same bit lengths of (p, q) such as 8, 10, 12 and 14. In these cases n will have binary length 16, 20, 24 and 28 respectively.

3 Results and Observations

3.1 Sample size of y is 10,000 i.e, $|S_y| = 10000$

Binary length of p and q is 8
i.e, p and q are both primes within the set $(2^7, 2^8)$

p	q	e	n	l	$\|S_y\|$	$\nu(E)$
163	197	5	32111	16	10000	0.0
149	251	3	37399	16	10000	0.0319
173	127	5	21971	16	10000	0.0
211	233	11	49163	16	10000	0.559
167	241	7	40247	16	10000	0.078

Table 1: For $S(E, y)$

As we can see from the results, for small values of p and q the frequency of finding the message from the ciphertext ranges from 0 to at most 50% . In subsequent tables for S(E,y) we will see that as we increase p and q the frequency of successfully retrieving the message get smaller and smaller.

p	q	e	n	l	$\|S_y\|$	$\nu(D)$
163	157	5	25591	16	10000	0.0
173	137	3	23701	16	10000	0.0
211	197	11	41567	16	10000	0.0
181	193	7	34933	16	10000	0.0
139	131	7	18209	16	10000	0.0

Table 2: For $S(D, y)$

The frequency of finding local inverse x (which is inverse of e mod(ϕ(n)) is consistently zero because a finite sequence of length $M = l^2$ is not long enough for us to calculate local inverse of S(D,y) correctly.

It took approximately 2.5 hours of computation to complete the above two tables.

Binary length of p and q is 10
i.e, p and q are both primes within the set $(2^9, 2^{10})$

p	q	e	n	l	$\lvert S_y \rvert$	$\nu(E)$
641	757	11	485237	20	10000	0.0
829	811	7	672319	20	10000	0.0912
937	761	7	713057	20	10000	0.4678
571	953	11	544163	20	10000	0.4798
887	683	3	605823	20	10000	0.0013

Table 3: For $S(E, y)$

As we chose five sets of two random values for p and q from the range $(2^9, 2^{10})$ we ended up with good values of $v(E)$ in this case coincidentally, and that might not be the case if we were to run the code again. Here, the frequency of getting correct local inverse x comes out to be in the range, zero to 50%, but the value never actually reaches 50%.

p	q	e	n	l	$\lvert S_y \rvert$	$\nu(D)$
887	769	5	682103	20	10000	0.0
563	907	5	510641	20	10000	0.0
967	709	5	685603	20	10000	0.0
571	743	11	424253	20	10000	0.0
673	643	5	432739	20	10000	0.0

Table 4: For $S(D, y)$

Here also as we try finding the local inverse for the decryption map S(D,y), the result remains the same as we never get the correct value of x due the length still not being long enough to give correct outputs. Hence $v(D)$ remains zero constantly.

It took approximately 4.5 hours of computation to complete the above two tables. Here, the time taken was due to the fact that we got good values for p and q, and local inverses existed for most of the sequences S(E,y). So we had to compute all of them and verify.

Binary length of p and q is 12
i.e, p and q are both primes within the set $(2^{11}, 2^{12})$

p	q	e	n	l	$\|S_y\|$	$\nu(E)$
3739	2909	5	10876751	24	10000	0.0005
2287	3359	5	7682033	24	10000	0.0
3347	3331	11	11148857	24	10000	0.0033
3137	3943	5	12369191	24	10000	0.0691
3251	2927	3	9515677	24	10000	0.0977

Table 5: For $S(E, y)$

Here, our inference from the theory shows up correctly as we can see that the value of $v(E)$ remains comparatively low throughout our sets. The range for $v(E)$ is 0 to close to 10% at maximum.

p	q	e	n	l	$\|S_y\|$	$\nu(D)$
2273	2953	5	6712169	24	10000	0.0
2281	3931	7	8966611	24	10000	0.0
3643	2801	11	10204043	24	10000	0.0
2797	3499	5	9786703	24	10000	0.0
4073	3617	3	14732041	24	10000	0.0

Table 6: For $S(D, y)$

The frequency of getting correct values for local inverse still remains zero.

It took approximately 3.5 hours of computation to complete the above two tables.

Binary length of p and q is 14
i.e, p and q are both primes within the set $(2^{13}, 2^{14})$

p	q	e	n	l	$\|S_y\|$	$\nu(E)$
11821	12689	7	149996669	28	10000	0.0026
9787	14669	5	143565503	28	10000	0.0
9733	15131	7	147270023	28	10000	0.0005
14549	14153	3	205911997	28	10000	0.0
12907	11807	5	152392949	28	10000	0.0

Table 7: For $S(E, y)$

Here the numbers p and q are taken from within the set $(2^{13}, 2^{14})$, are comparatively larger than for our previous computations and we can see that the frequency of getting correct local inverse goes to lower range than before. Reason behind this would be the large range for p and q because there will be many primes in the range, and the number of primes for which we can calculate local inverse decreases while the sample size is still kept at 10,000. Hence the value of $v(E)$ also decreases.

p	q	e	n	l	$\|S_y\|$	$\nu(D)$
12689	10529	3	133602481	28	10000	0.0
15787	14423	5	227695901	28	10000	0.0
14243	16111	7	229468973	28	10000	0.0
11239	9419	5	105860141	28	10000	0.0
10781	10639	13	114699059	28	10000	0.0

Table 8: For $S(D, y)$

The value for $v(D)$ still remains zero throughout our calculations.

It took approximately 3 hours of computation to complete the above two tables.

3.2 Sample size of y is 40,000 i.e, $|S_y| = 40000$

Binary length of p and q is 10
i.e, p and q are both primes within the set $(2^9, 2^{10})$

p	q	e	n	l	$\lvert S_y \rvert$	$\nu(E)$
599	577	5	345623	20	40000	0.0
937	787	5	737419	20	40000	0.01865
907	643	5	583201	20	40000	0.012025
617	727	5	448559	20	40000	0.0
631	1009	11	636679	20	40000	0.468975

Table 9: For $S(E, y)$

Taking p and q to be in the range $(2^9, 2^{10})$ we get similar results as per our previous calculations ($\nu(E)$ ranging from 0 to 50%) because when the above mentioned range is small we will have lesser number of primes and in turn lesser number of primes that don't give a local inverse x.

p	q	e	n	l	$\lvert S_y \rvert$	$\nu(D)$
523	691	7	361393	20	40000	0.0
877	823	5	721771	20	40000	0.0
727	739	5	537253	20	40000	0.0
769	977	5	751313	20	40000	0.0
661	809	7	534749	20	40000	0.0

Table 10: For $S(D, y)$

The frequency $v(D)$ remains the same yet again because the length ($M = l^2$) is still not long enough to give us correct local inverse.

It took approximately 5 hours of computation to complete the above two tables because of the large sample size.

3.3 Final Conclusions

To answer the question asked by the title of this assignment, 'Yes' we can sometimes break the RSA and find the message(local inverse x) by using the finite sequence corresponding to the encryption map but only for small primes p and q, with chances of success peaking at 50%(for 8 to 10 bit primes).

For decryption map D, based on our calculations we were never able to derive the local inverse, but if the length of the sequence was decently longer than **M** we could possibly find the local inverse correctly. Although it might take too much time and computation as the length of the sequence(**M**) increases.