# Chapter 2

# Practical Work

## 2.1 Technical details

Our implementation of the algorithms has taken into account the need for an efficient and fast testing system. Testing has been done on only one byte of plaintext, in ECB mode, since this is generally the fastest mode of operation. We have used an OOP approach in order to provide a flexible, modular and easy to extend system. Our initial attempts to use other modes of operation seemed impractical at the time in terms of the duration of the tests.

**Table creation** For table creation, in order to avoid duplicate chains we always test for the uniqueness of the starting points. In order to do this effectively, we have used a binary search tree that before generating any chain verifies if the starting point has not already been used. We also establish an explicit bijection between the start and end points. After the chain generation is complete, before writing the table on the disk we efficiently sort the endpoints. Writing to the table now is only possible because we have a bijection between the start and end points.

**Comparison with the endpoints** Since our tables are now sorted according to the endpoints, we make the comparison between our intermediate key and the endpoints using binary and exponential search, to perform a maximum of $O(nlog(n))$ operations. Other small improvements are that instead of performing string comparison we compare the memory buffers. Also, we make extensive use of the heap memory to store our data since it removes the need for copying the same values repeatedly.

### 2.1.1 DES Exhaustive Key Search

This implementation of the DES exhaustive key search starts with a counter set to 0.The counter consists of 32 bits that are distributed amongst the last 5 bytes of the key currently tested.Therefore the running time of the algorithm depends on the position of the most significant bit that has the value 1. It takes about 200 minutes for the counter to discover the key with the first bit 1 and the others set to 0 (11941s). The fifth bit is reached within 774s - 12 min, the

sixth in 382s()  6min), the seventh in 192s ( 3min), therefore the time required to discover the key increases exponentially. Given that, by the time the first bit is reached, half of the search space has been verified, this gives us an average of 200 minutes to discover the key.
Example - for the key:
1111111011111110111111101110001110001100100101110101101111100101
the running time was 2309s - 38.5 minutes

## 2.2 Hellman TMTO

The implementation of the Hellman method will be provided as a C++ header file, attached to this report.

### Table values

*Success rate*

| t\m | 16384 | 32768 | 65536 | 131072 | 262144 | 524288 | 1048576 | 2097152 | 4194304 | 8388608 |
|---|---|---|---|---|---|---|---|---|---|---|
| 512 | - | - | 6 | 11 | 9 | 17 | 23 | 32 | 47 | 57 |
| 1024 | - | 7 | 5 | 5 | 13 | 13 | 20 | 30 | 47 | - |
| 2048 | 3 | 1 | 8 | 9 | 12 | 15 | 24 | 34 | - | - |

The success rate is relative to 1000 tests. it has remained consistently lower than 6 % .

We have also tested the hypothesis whether it might be easier or more difficult to find a key if it has already been processed a certain number of times. We have applied the step function on a randomly generated key many times in order to test the convergence of the chains in our tables. Contrary to our expectations, of a bigger success rate, we got less hits than for entirely random keys. This means, that at least when used to recover the key of the DES, the Hellman method is really deployable.

The other tables are provided at the end of this document.

**Reduction function** Our choice of a reduction function has taken into account its prospective speed as the first argument for its usage. For a given ciphertext, we take the 7 most significant bits out of its last 4 bytes. From the fifth to last byte we take its four less significant bits.

The function takes these bits from the five less significant bytes

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |

And for the fifth to last byte:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | ✓ | ✓ | ✓ | ✓ | |

We have chosen this function because it corresponds to a memory buffer copy, an operation modulo $2^5$ and an assignment. It is thus very fast to compute, much faster than the 32 less significant bits. Unfortunately, it behaved worse on almost every test we performed.

**Time to compute on Copacabana** We will consider the table t = 1024 and m = 4194304. Using the same line of reasoning as in exercise 6, and assuming that the key could be found using the input table, we obtain that the average cryptanalysis time is $\frac{nr.encryptions}{15*48*10^8} = \frac{185250}{15*48*10^8} = 0.00000257s$

## 2.3 Rivest TMTO

For our tables, we do not store the chain length along with the start and end points. Since the efficiency of our buffer comparison is so high, once we have an alarm, we perform a comparison of the ciphertexts at every step of the recreation of the chain. The choice of the parameter d should be according to the average chain length that we desire. We have tried to accommodate the values of t and d such that t is not much higher than $2^d$ and the other way around.

## 2.4 Comparison between methods

In our case, Rivest has had a slightly lower average rate, but it still performs quite well. The case where d = 7 has performed better. Perhaps the reason for this algorithm's not so stellar performance is the low convergence of the chains in the Hellman tables for our setting.

After performing our tests, it seems that we have erroneously measured the memory access time. More precisely, we have omitted to take into account the time needed to match a candidate to the endpoints. The time required to do so using the Rivest method is much lower than the time required by our implementation of Hellman, since we break our search immediately after we get the first set of alarms. We have also measured the success rate of our algorithms if we continue iterating even after we reach a distinguished point during the online phase, in the hope that even if we did not merge with an existing chain we might be able to do so in the future. Maybe surprisingly enough only negligible variations are to be observed in the recovery rate.

# Hellman TMTO results

*Success rate*

| t\m | 16384 | 32768 | 65536 | 131072 | 262144 | 524288 | 1048576 | 2097152 | 4194304 | 8388608 |
|---|---|---|---|---|---|---|---|---|---|---|
| 512 | - | - | 6 | 11 | 9 | 17 | 23 | 32 | 47 | 57 |
| 1024 | - | 7 | 5 | 5 | 13 | 13 | 20 | 30 | 47 | - |
| 2048 | 3 | 1 | 8 | 9 | 12 | 15 | 24 | 34 | - | - |

*Alarm rate*

| t\m | 16384 | 32768 | 65536 | 131072 | 262144 | 524288 | 1048576 | 2097152 | 4194304 | 8388608 |
|---|---|---|---|---|---|---|---|---|---|---|
| 512 | - | - | 1.88 | 4.398 | 8.198 | 16.625 | 31.227 | 66.231 | 124.314 | 253.625 |
| 1024 | - | 3.88 | 8.259 | 15.58 | 30.471 | 62.953 | 124.195 | 254.876 | 527.055 | - |
| 2048 | 7.64 | 15.158 | 30.552 | 63.326 | 125.499 | 257.383 | 524.175 | 1032.41 | - | - |

*Average number of encryptions*

| t\m | 16384 | 32768 | 65536 | 131072 | 262144 | 524288 | 1048576 | 2097152 | 4194304 | 8388608 |
|---|---|---|---|---|---|---|---|---|---|---|
| 512 | - | - | 826.57 | 1254.72 | 1955.39 | 3391.18 | 5917.86 | 11948.6 | 21434.4 | 43889.7 |
| 1024 | - | 2279.45 | 3898.95 | 6309.28 | 11357.8 | 22619 | 42880.8 | 86870.2 | 185250 | - |
| 2048 | 7169.69 | 12181.5 | 22796.2 | 45897.4 | 87367.4 | 178308 | 367626 | 702149 | - | - |

*Read table time*

| t\m | 16384 | 32768 | 65536 | 131072 | 262144 | 524288 | 1048576 | 2097152 | 4194304 | 8388608 |
|---|---|---|---|---|---|---|---|---|---|---|
| 512 | - | - | 0.027287 | 0.037347 | 0.077952 | 0.145621 | 0.284914 | 0.552151 | 1.14965 | 2.23479 |
| 1024 | - | 0.013632 | 0.026276 | 0.035899 | 0.07003 | 0.14807 | 0.29567 | 0.568602 | 1.12763 | - |
| 2048 | 0.005136 | 0.01075 | 0.035985 | 0.046705 | 0.079979 | 0.13963 | 0.283689 | 0.548732 | - | - |

*Encrypt time*

| t\m | 16384 | 32768 | 65536 | 131072 | 262144 | 524288 | 1048576 | 2097152 | 4194304 | 8388608 |
|---|---|---|---|---|---|---|---|---|---|---|
| 512 | - | - | 0.000416524 | 0.000608301 | 0.000934446 | 0.00159589 | 0.00274543 | 0.0057239 | 0.0100153 | 0.0204408 |
| 1024 | - | 0.00110904 | 0.00186312 | 0.002971 | 0.00533433 | 0.010543 | 0.0198321 | 0.0402688 | 0.0847293 | - |
| 2048 | 0.00343169 | 0.0058463 | 0.0109038 | 0.0218218 | 0.0412658 | 0.08078 | 0.167071 | 0.364431 | - | - |

*Test time*

| t\m | 16384 | 32768 | 65536 | 131072 | 262144 | 524288 | 1048576 | 2097152 | 4194304 | 8388608 |
|---|---|---|---|---|---|---|---|---|---|---|
| 512 | - | - | 0.00158009 | 0.00232139 | 0.00357258 | 0.00580993 | 0.0100439 | 0.0200469 | 0.0438221 | 0.0852149 |
| 1024 | - | 0.0028273 | 0.00448556 | 0.00683111 | 0.0116088 | 0.0213902 | 0.038945 | 0.0775758 | 0.166565 | - |
| 2048 | 0.00711442 | 0.0115494 | 0.0205248 | 0.0393977 | 0.0728286 | 0.13808 | 0.280968 | 0.60854 | - | - |

# Hellman with custom function

The alarm rates are similar but the success rate is lower.

*Success rate*

| t\m | 16384 | 32768 | 65536 | 131072 | 262144 | 524288 | 1048576 | 2097152 | 4194304 | 8388608 |
|---|---|---|---|---|---|---|---|---|---|---|
| 512 | - | - | 4 | 10 | 6 | 18 | 24 | 36 | 41 | - |
| 1024 | - | 4 | 13 | 9 | 12 | 17 | 24 | 30 | - | - |
| 2048 | 1 | 9 | 6 | 10 | 11 | 19 | 21 | - | - | - |

*Alarm rate*

| t\m | 16384 | 32768 | 65536 | 131072 | 262144 | 524288 | 1048576 | 2097152 | 4194304 | 8388608 |
|---|---|---|---|---|---|---|---|---|---|---|
| 512 | - | - | 1.942 | 3.96 | 8.219 | 16.034 | 31.731 | 65.416 | 129.448 | - |
| 1024 | - | 4.255 | 7.863 | 16.371 | 30.767 | 62.695 | 126.863 | 260.875 | - | - |
| 2048 | 8.415 | 16.058 | 32.561 | 63.553 | 133.385 | 254.319 | 519.574 | - | - | - |

*Average number of encryptions*

| t\m | 16384 | 32768 | 65536 | 131072 | 262144 | 524288 | 1048576 | 2097152 | 4194304 | 8388608 |
|---|---|---|---|---|---|---|---|---|---|---|
| 512 | - | - | 834.548 | 1171.64 | 1929.33 | 3260.41 | 6016.88 | 11558.9 | 22870.6 | - |
| 1024 | - | 2533.63 | 3762.86 | 6585.53 | 11632.3 | 22677.4 | 43651 | 93134.1 | - | - |
| 2048 | 7810.69 | 12746.3 | 24240 | 45706 | 94669.1 | 174358 | 354674 | - | - | - |

*Read table time*

| t\m | 16384 | 32768 | 65536 | 131072 | 262144 | 524288 | 1048576 | 2097152 | 4194304 | 8388608 |
|---|---|---|---|---|---|---|---|---|---|---|
| 512 | - | - | 0.032423 | 0.066682 | 0.132445 | 0.26367 | 0.522328 | 1.03105 | 2.09723 | - |
| 1024 | - | 0.015936 | 0.033856 | 0.064677 | 0.144506 | 0.247852 | 0.500674 | 0.994911 | - | - |
| 2048 | 0.004536 | 0.016726 | 0.018952 | 0.063087 | 0.12429 | 0.143207 | 0.50257 | - | - | - |

*Encrypt time*

| t\m | 16384 | 32768 | 65536 | 131072 | 262144 | 524288 | 1048576 | 2097152 | 4194304 | 8388608 |
|---|---|---|---|---|---|---|---|---|---|---|
| 512 | - | - | 0.000654366 | 0.000916259 | 0.00156048 | 0.00257075 | 0.00483137 | 0.0090826 | 0.0180006 | - |
| 1024 | - | 0.0019674 | 0.00295258 | 0.00531078 | 0.0068251 | 0.0121478 | 0.0222111 | 0.0502736 | - | - |
| 2048 | 0.00484498 | 0.00648876 | 0.012598 | 0.0237749 | 0.052015 | 0.090551 | 0.183524 | - | - | - |

*Test time*

| t\m | 16384 | 32768 | 65536 | 131072 | 262144 | 524288 | 1048576 | 2097152 | 4194304 | 8388608 |
|---|---|---|---|---|---|---|---|---|---|---|
| 512 | - | - | 0.00214478 | 0.00311558 | 0.00490363 | 0.00795825 | 0.0140707 | 0.0256465 | 0.0783554 | - |
| 1024 | - | 0.00462325 | 0.00666009 | 0.0112641 | 0.0139652 | 0.0232679 | 0.0410689 | 0.0895754 | - | - |
| 2048 | 0.00945544 | 0.0121429 | 0.0223817 | 0.0413127 | 0.0863203 | 0.146799 | 0.293194 | - | - | - |

# Table size

The table size is the same, regarding of the algorithm used.

*Table size*

| t\m | 16384 | 32768 | 65536 | 131072 | 262144 | 524288 | 1048576 | 2097152 | 4194304 | 8388608 |
|---|---|---|---|---|---|---|---|---|---|---|
| 512 | - | - | 1,0 MB | 2,1 MB | 4,2 MB | 8,4 MB | 16,8 MB | 33,6 MB | 67,1 MB | 134,2 MB |
| 1024 | - | 524,3 kB | 1,0 MB | 2,1 MB | 4,2 MB | 8,4 MB | 16,8 MB | 33,6 MB | 67,1 MB | - |
| 2048 | 262,1 kB | 524,3 kB | 1,0 MB | 2,1 MB | 4,2 MB | 8,4 MB | 16,8 MB | 33,6 MB | - | - |

# Using two tables

The comparative statistics for two tables of size 1024 x 2097152 with two functions versus one table of 1024 x 40194394 with one function. As it can be seen, the combination of two different tables with two different functions has reduced the number of collisions and lead to a higher number of successes.

| Measurement | 1024-2097152-LSB | 1024-2097152-CUSTOM | both tables | 1024-4194304-LSB |
|---|---|---|---|---|
| Number of successes: | 30 | 24 | 55 | 47 |
| Average number of alarms: | 254.876 | 267.932 | 500.005 | 527.055 |
| Average number of encryptions: | 86870.2 | 95054.1 | 173307 | 185250 |
| Table read time(seconds): | 0.568602 | 0.0516563 | 1.07059 | 1.12763 |
| Average encrypting time(seconds): | 0.0402688 | 0.0516563 | 0.0814892 | 0.0847293 |
| Average test time(seconds): | 0.0775758 | 0.0905885 | 0.153262 | 0.166565 |

## Rivest-LSB, d=7

Statistical data:

*Success rate*

| t\m | 16384 | 32768 | 65536 | 131072 | 262144 | 524288 | 1048576 | 2097152 | 4194304 | 8388608 |
|---|---|---|---|---|---|---|---|---|---|---|
| 512 | - | - | 3 | 3 | 4 | 6 | 14 | 27 | 28 | 39 |
| 1024 | - | 1 | 0 | 2 | 1 | 10 | 12 | 27 | 46 | - |
| 2048 | 1 | 0 | 0 | 1 | 8 | 10 | 11 | 25 | - | - |

*Alarm rate*

| t\m | 16384 | 32768 | 65536 | 131072 | 262144 | 524288 | 1048576 | 2097152 | 4194304 | 8388608 |
|---|---|---|---|---|---|---|---|---|---|---|
| 512 | - | - | 0.751 | 1.52 | 3.009 | 6.031 | 11.77 | 23.633 | 47.316 | 91.736 |
| 1024 | - | 0.87 | 1.77 | 3.607 | 7.153 | 13.56 | 28.684 | 54.824 | 110.432 | - |
| 2048 | 0.967 | 1.921 | 3.755 | 7.545 | 15.075 | 30.109 | 60.082 | 117.89 | - | - |

*Average number of encryptions*

| t\m | 16384 | 32768 | 65536 | 131072 | 262144 | 524288 | 1048576 | 2097152 | 4194304 | 8388608 |
|---|---|---|---|---|---|---|---|---|---|---|
| 512 | - | - | 607.356 | 697.026 | 890.538 | 1272.31 | 1981.72 | 3503.32 | 6305.36 | 11780.3 |
| 1024 | - | 1181.36 | 1354.43 | 1679.69 | 2373.78 | 3474.59 | 6350.99 | 11124.2 | 21161.8 | - |
| 2048 | 2268.67 | 2482.82 | 2874.19 | 3699.01 | 5380.43 | 8625.19 | 15193.8 | 27793.1 | - | - |

*Read table time*

| t\m | 16384 | 32768 | 65536 | 131072 | 262144 | 524288 | 1048576 | 2097152 | 4194304 | 8388608 |
|---|---|---|---|---|---|---|---|---|---|---|
| 512 | - | - | 0.021385 | 0.046203 | 0.081063 | 0.170726 | 0.37845 | 0.643534 | 2.55669 | 4.89161 |
| 1024 | - | 0.019859 | 0.042712 | 0.076657 | 0.159459 | 0.310815 | 0.416429 | 1.31612 | 2.62924 | - |
| 2048 | 0.009586 | 0.02289 | 0.045014 | 0.080103 | 0.156802 | 0.306007 | 0.593901 | 1.2024 | - | - |

*Encrypt time*

| t\m | 16384 | 32768 | 65536 | 131072 | 262144 | 524288 | 1048576 | 2097152 | 4194304 | 8388608 |
|---|---|---|---|---|---|---|---|---|---|---|
| 512 | - | - | 0.000358332 | 0.000400107 | 0.000516005 | 0.000795209 | 0.00114063 | 0.00208564 | 0.00520924 | 0.00836875 |
| 1024 | - | 0.000859513 | 0.00119848 | 0.00129112 | 0.00171893 | 0.00263655 | 0.00552649 | 0.0095004 | 0.0162402 | - |
| 2048 | 0.00188977 | 0.00208579 | 0.00231552 | 0.00294245 | 0.00439546 | 0.00737082 | 0.0129624 | 0.0234513 | - | - |

*Test time*

| t\m | 16384 | 32768 | 65536 | 131072 | 262144 | 524288 | 1048576 | 2097152 | 4194304 | 8388608 |
|---|---|---|---|---|---|---|---|---|---|---|
| 512 | - | - | 0.000869975 | 0.00128629 | 0.00207692 | 0.00375936 | 0.00637659 | 0.0128481 | 0.0615064 | 0.120275 |
| 1024 | - | 0.0018223 | 0.00238418 | 0.00360741 | 0.00555714 | 0.00931853 | 0.0198355 | 0.0371093 | 0.0995058 | - |
| 2048 | 0.00350362 | 0.0040866 | 0.00491755 | 0.00673086 | 0.0105012 | 0.0193791 | 0.0345446 | 0.0658356 | - | - |

## Rivest-LSB, d=11

Statistical data for Rivest d=11 using LSB

*Success rate*

| t\m | 16384 | 32768 | 65536 | 131072 | 262144 | 524288 | 1048576 | 2097152 | 4194304 | 8388608 |
|---|---|---|---|---|---|---|---|---|---|---|
| 512 | - | - | 5 | 7 | 5 | 10 | 11 | 12 | - | - |
| 1024 | - | 1 | 2 | 1 | 8 | 8 | 12 | - | - | - |
| 2048 | 0 | 0 | 3 | 6 | 9 | 8 | - | - | - | - |

*Alarm rate*

| t\m | 16384 | 32768 | 65536 | 131072 | 262144 | 524288 | 1048576 | 2097152 | 4194304 | 8388608 |
|---|---|---|---|---|---|---|---|---|---|---|
| 512 | - | - | 1.149 | 2.846 | 4.566 | 9.17 | 22.048 | 43.163 | - | - |
| 1024 | - | 2.644 | 5.672 | 9.696 | 18.298 | 44.922 | 75.446 | - | - | - |
| 2048 | 4.622 | 8.932 | 18.266 | 37.908 | 71.634 | 162.158 | - | - | - | - |

*Average number of encryptions*

| t\m | 16384 | 32768 | 65536 | 131072 | 262144 | 524288 | 1048576 | 2097152 | 4194304 | 8388608 |
|---|---|---|---|---|---|---|---|---|---|---|
| 512 | - | - | 689.707 | 889.302 | 1212.47 | 1933.81 | 3899.52 | 7031.62 | - | - |
| 1024 | - | 1823.01 | 2652.32 | 4039.83 | 6703.07 | 15464.8 | 24756.1 | - | - | - |
| 2048 | 4946.34 | 7382.6 | 13592.5 | 24558.3 | 46610.1 | 100205 | - | - | - | - |

*Read table time*

| t\m | 16384 | 32768 | 65536 | 131072 | 262144 | 524288 | 1048576 | 2097152 | 4194304 | 8388608 |
|---|---|---|---|---|---|---|---|---|---|---|
| 512 | - | - | 0.039504 | 0.079287 | 0.103238 | 0.221085 | 0.62203 | 1.21466 | - | - |
| 1024 | - | 0.019721 | 0.039589 | 0.086928 | 0.166311 | 0.334427 | 0.52224 | - | - | - |
| 2048 | 0.011264 | 0.019364 | 0.037705 | 0.069938 | 0.1286 | 0.283627 | - | - | - | - |

*Encrypt time*

| t\m | 16384 | 32768 | 65536 | 131072 | 262144 | 524288 | 1048576 | 2097152 | 4194304 | 8388608 |
|---|---|---|---|---|---|---|---|---|---|---|
| 512 | - | - | 0.000635409 | 0.000775561 | 0.000981688 | 0.00142652 | 0.00273554 | 0.00619449 | - | - |
| 1024 | - | 0.00164321 | 0.00242858 | 0.00377778 | 0.0062616 | 0.0133059 | 0.01878 | - | - | - |
| 2048 | 0.00452778 | 0.00682616 | 0.0109959 | 0.0197597 | 0.0349132 | 0.0762174 | - | - | - | - |

*Test time*

| t\m | 16384 | 32768 | 65536 | 131072 | 262144 | 524288 | 1048576 | 2097152 | 4194304 | 8388608 |
|---|---|---|---|---|---|---|---|---|---|---|
| 512 | - | - | 0.00139452 | 0.00198964 | 0.00295242 | 0.00479048 | 0.00919519 | 0.0211038 | - | - |
| 1024 | - | 0.00300194 | 0.00458419 | 0.00732528 | 0.0124014 | 0.0262952 | 0.0378502 | - | - | - |
| 2048 | 0.00779797 | 0.0118388 | 0.0192515 | 0.0338108 | 0.0601655 | 0.129387 | - | - | - | - |

## Rivest-custom function, d=11

Statistical data for Rivest d=11 using custom function R.

*Success rate*

| t\m | 16384 | 32768 | 65536 | 131072 | 262144 | 524288 | 1048576 | 2097152 | 4194304 | 8388608 |
|---|---|---|---|---|---|---|---|---|---|---|
| 512 | - | - | 4 | 3 | 5 | 8 | 9 | 12 | - | - |
| 1024 | - | 3 | 3 | 3 | 4 | 10 | 15 | 20 | - | - |
| 2048 | 1 | 3 | 5 | 6 | 4 | 9 | 17 | 27 | - | - |

*Alarm rate*

| t\m | 16384 | 32768 | 65536 | 131072 | 262144 | 524288 | 1048576 | 2097152 | 4194304 | 8388608 |
|---|---|---|---|---|---|---|---|---|---|---|
| 512 | - | - | 1.29 | 2.274 | 5.015 | 9.564 | 20.341 | 42.999 | - | - |
| 1024 | - | 2.363 | 4.891 | 9.05 | 20.651 | 43.926 | 75.373 | 170.549 | - | - |
| 2048 | 4.355 | 8.833 | 18.082 | 38.658 | 71.621 | 151.576 | 305.977 | 615.334 | - | - |

*Average number of encryptions*

| t\m | 16384 | 32768 | 65536 | 131072 | 262144 | 524288 | 1048576 | 2097152 | 4194304 | 8388608 |
|---|---|---|---|---|---|---|---|---|---|---|
| 512 | - | - | 720.755 | 885.371 | 1345.27 | 1993.48 | 3746.82 | 6459.9 | - | - |
| 1024 | - | 1780.7 | 2557.44 | 3834.57 | 7152.32 | 14437.5 | 24400.3 | 53862 | - | - |
| 2048 | 4702.44 | 7437.61 | 12820.2 | 26258.8 | 46215.7 | 90223.1 | 190830 | 371544 | - | - |

*Read table time*

| t\m | 16384 | 32768 | 65536 | 131072 | 262144 | 524288 | 1048576 | 2097152 | 4194304 | 8388608 |
|---|---|---|---|---|---|---|---|---|---|---|
| 512 | - | - | 0.020288 | 0.046788 | 0.072139 | 0.143997 | 0.30013 | 0.541175 | - | - |
| 1024 | - | 0.017461 | 0.025791 | 0.04855 | 0.1293 | 0.178755 | 0.550827 | 1.13346 | - | - |
| 2048 | 0.008522 | 0.020675 | 0.043082 | 0.08063 | 0.154513 | 0.258994 | 0.289505 | 0.536621 | - | - |

*Encrypt time*

| t\m | 16384 | 32768 | 65536 | 131072 | 262144 | 524288 | 1048576 | 2097152 | 4194304 | 8388608 |
|---|---|---|---|---|---|---|---|---|---|---|
| 512 | - | - | 0.000348295 | 0.000444406 | 0.000658977 | 0.000959509 | 0.00177931 | 0.00301587 | - | - |
| 1024 | - | 0.00110853 | 0.00153586 | 0.00230296 | 0.0039628 | 0.00767056 | 0.0163986 | 0.0400612 | - | - |
| 2048 | 0.00338252 | 0.00550068 | 0.00910765 | 0.01816 | 0.0319376 | 0.048589 | 0.085483 | 0.165066 | - | - |

*Test time*

| t\m | 16384 | 32768 | 65536 | 131072 | 262144 | 524288 | 1048576 | 2097152 | 4194304 | 8388608 |
|---|---|---|---|---|---|---|---|---|---|---|
| 512 | - | - | 0.000805207 | 0.00131677 | 0.00221073 | 0.00379083 | 0.00733685 | 0.0134469 | - | - |
| 1024 | - | 0.00213786 | 0.00314245 | 0.00504023 | 0.00882712 | 0.0166349 | 0.0352526 | 0.0831724 | - | - |
| 2048 | 0.00568151 | 0.00924226 | 0.0153296 | 0.0307549 | 0.0544799 | 0.0800006 | 0.13777 | 0.265578 | - | - |