TMTO attack on Light-Weight Cipher

Project Presentation for CSN-300 (Spring Semester 2020-2021)

Radhika (18114060)

Rishi Ranjan (18114066)

Shubhang Tripathi (18114074)

Supervised by **Prof. Sugata Gangopadhyay**

May 5, 2021

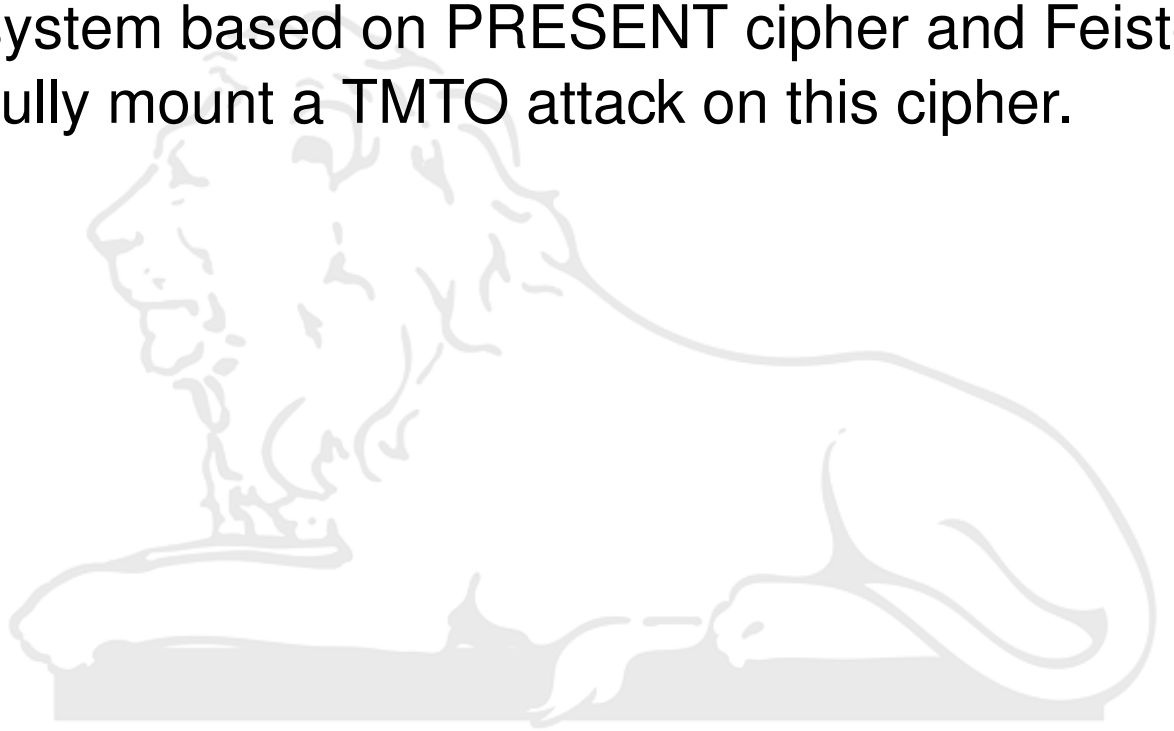
Table of Contents

Introduction

Many modern cryptographic systems today are based on the famous Feistel block cipher design and thus, cryptanalysis of this block cipher is of upmost importance for the researchers.
In this report, we present a refined version of the Time Memory Tradeoff attack (TMTO) as introduced by Hellman in [1].
The major limitation of Hellman's TMTO attack lies in the fact that it can only work for a cryptosystem with equal plaintext space, ciphertext space and keyspace. Further work and refinements have been done by Hellman to deal with smaller keyspaces. But the problem of dealing with larger keyspaces is still very open in the field of symmetric key cryptanalysis.

Introduction

In this report, we present a novel strategy to successfully carry out a TMTO attack on a cryptosystem with a larger keyspace than plaintext and ciphertext space. We then test our technique by implementing a novel cryptosystem based on PRESENT cipher and Feistel ciphers and successfully mount a TMTO attack on this cipher.



Literature Review

- Symmetric Key Cryptography - The cryptographic scheme in which the key used for encryption and decryption is the same. A good review can be found in [2]
- Block Ciphers - These are the major type of symmetric cryptographic systems, famous block ciphers include DES, AES etc. These ciphers have been surveyed well in [3]
- Substitution Permutation Network -Substitution Permutationnetworks are linked mathematical operations (to produce sub-stitutions) which are widely used in most of the block ciphers.[4]

Literature Review

- Pseudorandom Permutation Functions - These are permutation functions which are indistinguishible from random permutation (i.e permutation chose randomly from a set of permutations. The technique of creating pseudorandom permutation from pseudorandom function generators has been well studied in [5]
- TMTO Attack - Time Memory Tradeoff attack are probabilistic attacks which aim to cryptanlyze an N key system with $N^{2/3}$ time and $N^{2/3}$ memory. Thus this attack was a middle ground between space inefficient constant time attacks and time inefficient key search attacks. This was introduced by Hellman in [1]

Problem Statement

In this project we take up the problem statement of mouting a TMTO attack on a block cipher with larger keyspace than plaintext and ciphertext space. Thus the problem statement is divided into three major points.

- Implementation of a 3 round feistel network with an added substitution box from the PRESENT cipher.
- Mounting a time-memory-tradeoff attack on this feistel network cipher.

We also take care of another nuance while launching this attack. Since the keyspace is larger than ciphertext and plaintext space, we also take care of the collision which might occur between two keys $k1$ and $k2$. We finally introduce a novel attack strategy to deal with ciphers having larger keyspace than ciphertext/plaintext space.

Methodology

We begin with the implementation of a lightweight block cipher based on the feistel network. A 3 round feistel network cipher is used as the basis. The block size of the cryptosystem is 8. The key length is 12. If the key structure is of the form

$$k = k_1 k_2 ... k_{12}$$

We define the three round keys to be

$$k^1 = k_1 ... k_4 \oplus k_5 ... k_8$$

$$k^2 = k_5 ... k_8 \oplus k_9 ... k_{12}$$

$$k^3 = k_9 ... k_{12} \oplus k_1 ... k_4$$

Methodology

The standard encryption mechanism for a feistel network is used for each round.The substitution box used by our cipher is taken from the PRESENT cipher [6]
For each round, the plaintext block is divided into two halfs, left and right blocks. The left block x passes through a round key function

$$f(x, k_i) = x \oplus k_i$$

Finally both the halves are passed through the S-box and swapped with each other for the next round.

Attack Implementation

Attack implementation that we've chosen is a chosen plaintext TMTO attack on the above implemented cipher. Since, we're dealing with a larger keyspace, we deal with this by apppending 4 extra bits to the MSB of our ciphertext generated thus making our ciphertext and keyspace equal now.

$$C_m = r_1 r_2 r_3 r_4 C$$

This C_m acts as a key and entry into the TMTO matrix that we generate.
This attack has two phases, the offline phase (which generates these TMTO matrices) and the online phase, which searches for this key in the given matrices.

Precomputation Phase

We generate the TMTO lists for a semi exhaustive key search in this phase. We partition our plaintext into blocks of length 8 and compute lists for each.
For each list L_i the bits appended to the ciphertext are the binary representation i to help us in the search later.
These lists are generated using pseudo-random permutation functions which are defined as

$$f_p^j(X_{i0}) = X_{i0} \oplus (j * (2^4 + 1))$$

This generates random keys from computed keys for better regularity in our matrices.
Also, for every element in a row in the list, the next element is calculated as the ciphertext generated by encryption with the current element.

Precomputation Phase

Finally with each list calculated, in every row, drop the intermediate value and store only the initial and final list entries. Thus each list has $2 * m$ entries where m is TMTO parameter. Thus for t generated lists, we have total $2mt$ elements stored.

Online Phase

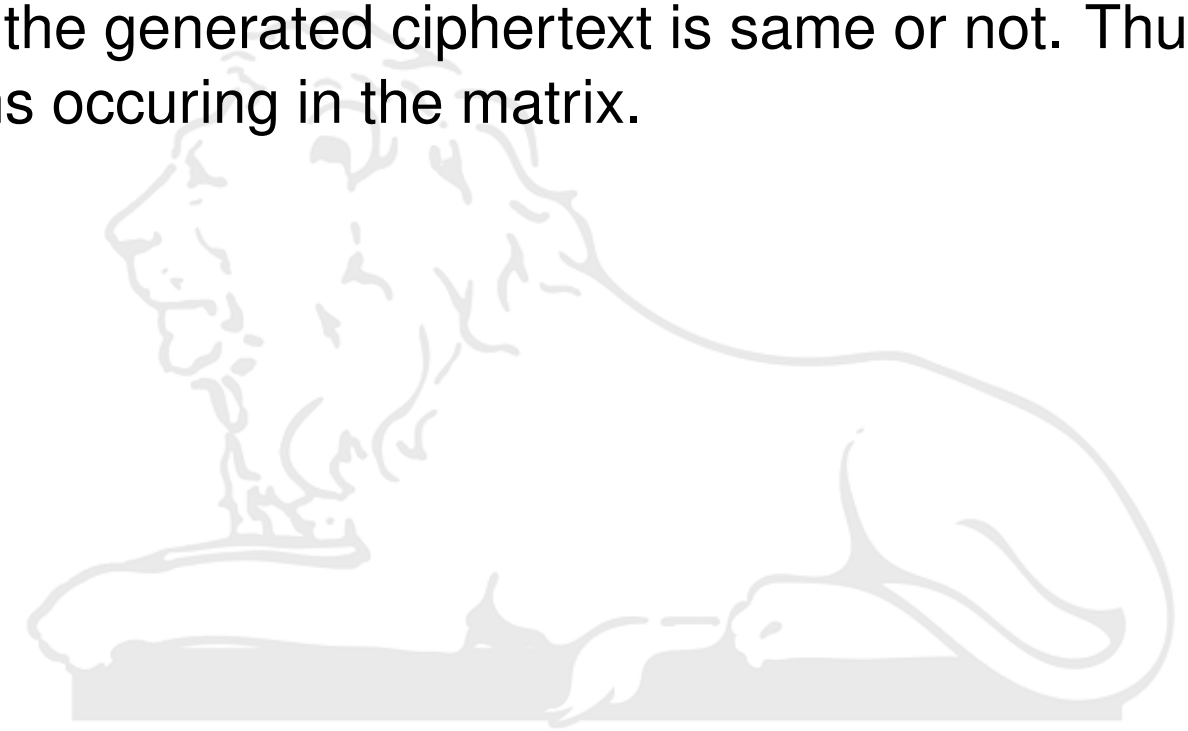
The online phase is the actual key search algorithm used. For given Ciphertext C, we chose a list L_i to search in, and append the appropriate bits to C. Then we encrypt the given plaintext with this ciphertext and search for it in list. If it isn't found, the ciphertext generated by it is used as the key and search continues.
When a match is found after x tries, we take the first element of that row and apply the function

$$X_{ij} = f(X_{i,j-1})$$

$t - x - 1$ times which gives us one of the discarded intermediate values and the key for this cryptosystem.

Online Phase

There is some overlap in the elements of a matrix. Thus, we check if the key found is a false positive and proceed accordingly. If a key is found, we keep encrypting the other blocks of plaintext with the key and check if the generated ciphertext is same or not. Thus, we deal with collisions occuring in the matrix.



Simulation Results

The results obtained through are methods and strategy are summarised below. $N = 2mt$ is the total space complexity that is consumed by the TMTO lists.

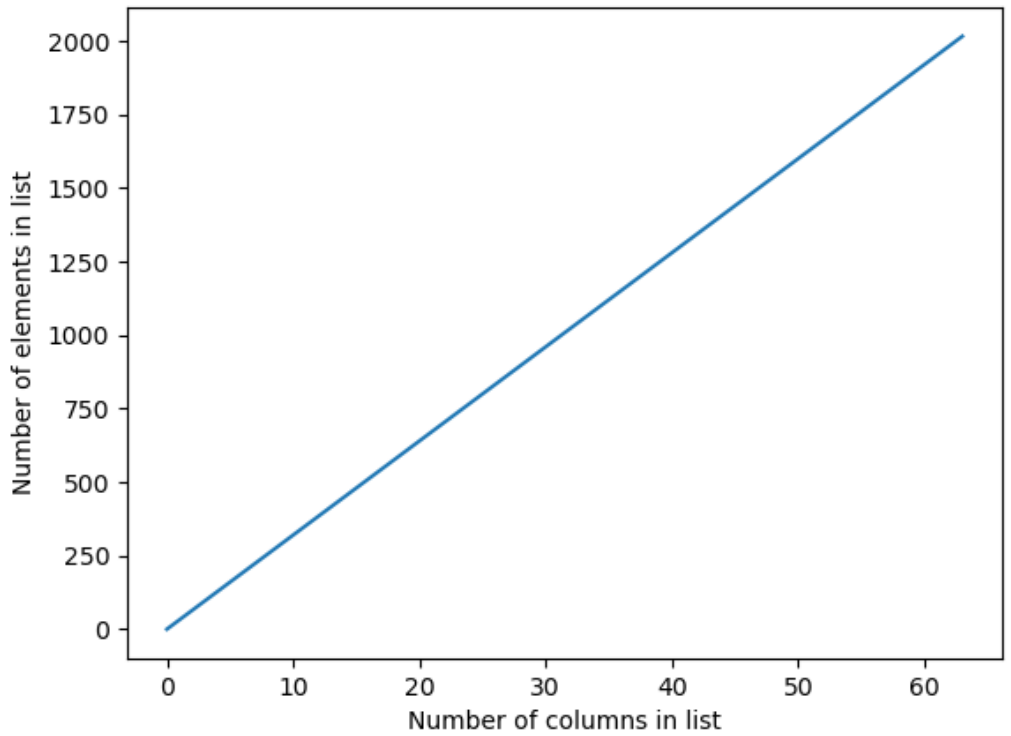


Figure: N vs t

Simulation Results

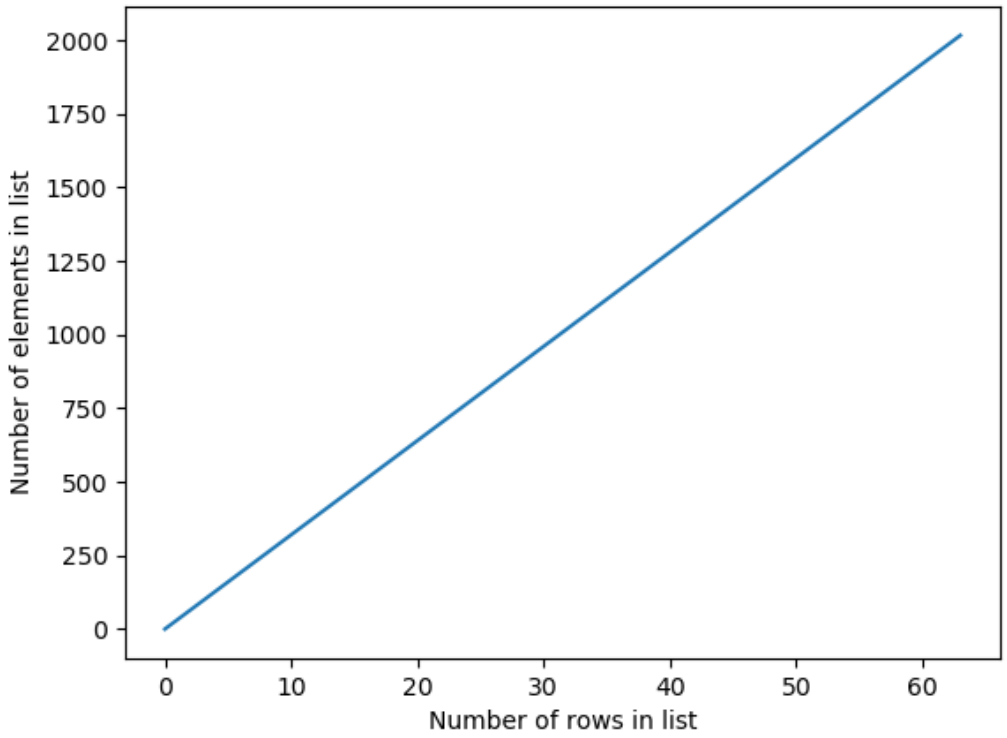


Figure: N vs m

Simulation Results

The precomputation time is calculated to be

$$P = t^2 m + m$$

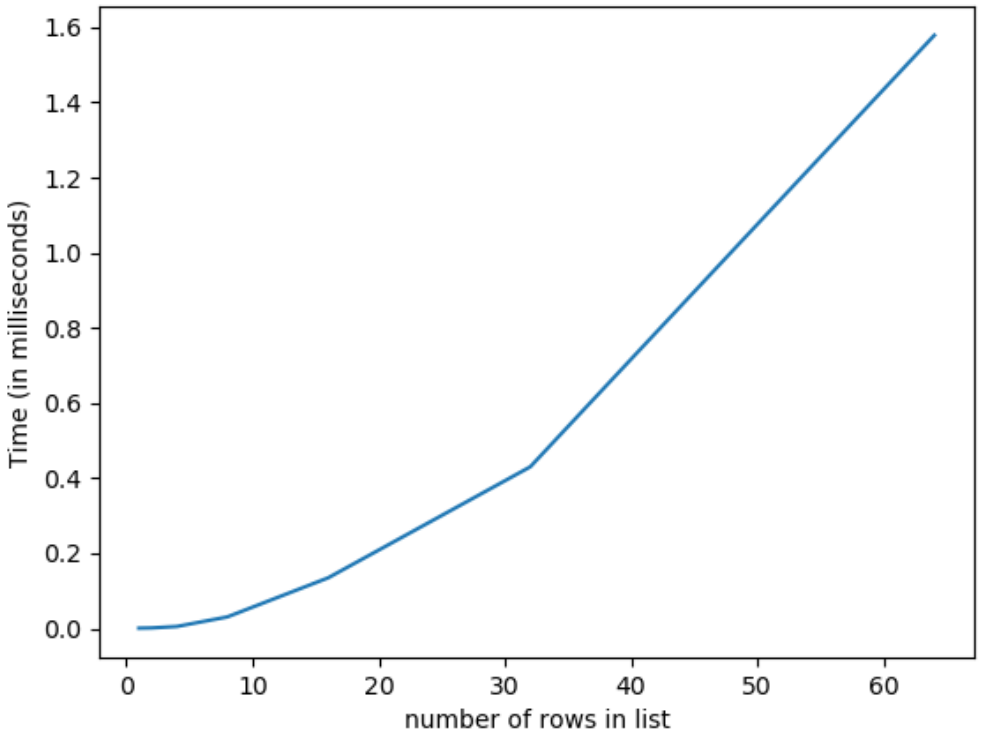


Figure: P vs t

Simulation Results

Finally the online/search phase of the algorithm has the best case complexity of $T = t^2$ and the worst case complexity being $T = nt^2$. The search is fastened by using a hash table which makes the search constant.

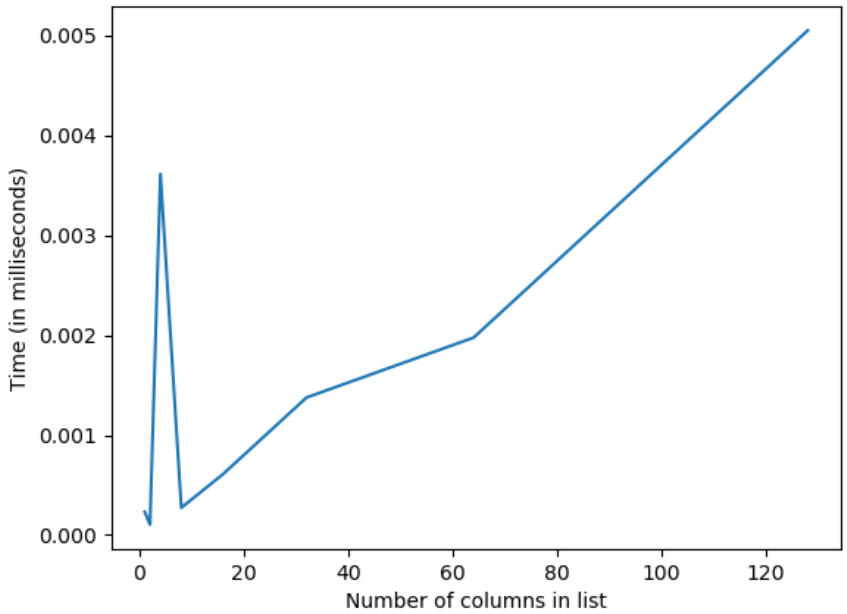


Figure: T vs t

Conclusions

In this project, we successfully mounted a Time Memory tradeoff attack on a well designed fiestel network and showed its eficiency. We implement a novel lightweight block cipher based on the feistelnetwork, by adding an extra layer of substitution box which helps ensure Shanon's propertyof confusion. We took up the intersting and open problem of mounting TMTO attacks on ciphers with keyspaces larger than plaintext and ciphertext spaces and introduced a novel strategy in order to tackle this. The final implementation presented is also spatially and computationally efficient.

References

- M. Hellman, “A cryptanalytic time-memory trade-off,” *IEEE Transactions on Information Theory*, vol. 26, no. 4, pp. 401–406, 1980.
- M. Ubaidullah and Q. Makki, “A review on symmetric key encryption techniques in cryptography,” *International Journal of Computer Applications*, vol. 147, pp. 43–48, 2016.
- S. Albermany and F. Radi, “Survey: Block cipher methods,” vol. 5, 11 2016.
- J. Katz and Y. Lindell, *Introduction to Modern Cryptography, Second Edition*. Chapman Hall/CRC, 2nd ed., 2014.
- M. Luby and C. Rackoff, “How to construct pseudorandom permutations from pseudorandom functions,” *SIAM Journal on Computing*, vol. 17, no. 2, pp. 373–386, 1988.
- A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and C. Vikkelsoe, “Present: An ultra-lightweight block cipher,” in *Cryptographic Hardware and Embedded Systems - CHES 2007* (P. Paillier and I. Verbauwhede, eds.), (Berlin, Heidelberg), pp. 450–466, Springer Berlin Heidelberg, 2007.

Thank you!