Appendix

Reviewer: 1

I do appreciate the revised manuscript and the explanations given in the cover letter. For the newly added complexity of exploring mappings, I'd suggest to support the mathematical derivation through references. (I also understand about space and references limitations w/ ESL. If needed, please consider "outsourcing" references to all the prior art you studied, mentioned as >100 papers in the cover letter, to some webpage of yours.)

Response: We would like to thank the honourable reviewer for their valuable remark. We have assembled all the papers that we studied in a separate GitHub webpage- `https://github.com/nivi1501/ESL/`.

The mathematical derivation of address mapping complexity is a novel concept. Due to constraints on space, the manuscript does not detail all intermediate steps of the derivation. We present two detailed proves to support our argument:

Lemma 1: *An adversary needs to try out $n^{O(loglogn)}$ candidate mappings to find out the correct X - Y mapping.*

Proof: Algorithms such as AES have an interesting property, for accessing T-tables or S-boxes we need to XOR plain text bytes with key bytes. A XOR operation preserves the Hamming distance (for the same key).

Section 3.2 of Reference [1] emphasizes this point. We have exploited this dependency in our idea. The mathematical derivations are straightforward and are described in detail below. We can represent T-table entries as a hypercube. A hypercube with n vertices has $n(logn)!$ labellings. We apply sterling approximation which is:

$$n! = \sqrt{2\pi n}(\frac{n}{e})^n \tag{1}$$

Using this we derive logn! as:

$$(logn)! = \sqrt{2\pi logn}(\frac{logn}{e})^{logn} \tag{2}$$

Additionally:

$$(logn)^{logn} = n^{loglogn} \tag{3}$$

In the equation above, adding log on both sides:

$$logn.log(logn) = loglogn.logn \tag{4}$$

Additionally,

$$e^{log_2 n} = e^{lnn.log_e 2} = n^{ln2} \tag{5}$$

Putting Equations 5 and 3 in Equation 2, we get:

$$(logn)! = \sqrt{2\pi logn}\frac{n^{loglogn}}{n^{ln2}} \tag{6}$$

As $\sqrt{logn} < n$ and emitting constants, we conclude that the (logn)! grows with a complexity of $O(n^{loglogn})$, which is a slow growing function.

Now, we add another lemma to further back up our argument. The results are not included in the main manuscript due to space constraint.

Lemma 2: $I(K;Y|T)$ establishes a lower bound on $I(X;Y|T)$.

Proof: Given the plaintext T, key K, we have a Markov chain $\langle key(K), T\rangle \rightarrow X \rightarrow Y$ [2]. This basically means that the MI between X and Y determines the amount of information that can be extracted out of Y about the key. The two assumptions are that this is a Markov process and the noise distribution N is independent of the key, K.

In the generic case, we can use the data processing inequality in information theory and write $I(K;Y|T) \leq I(X;Y|T)$ (derived from Eqn.8 in [2]). Below is a description of the steps in the derivation:

Using Equation 10 in [2]:

$$I((X,T);(Y,T)) = I(X;Y|T) + H(T) \tag{7}$$

$$I((K,T);(Y,T)) = I(K;Y|T) + H(T) \tag{8}$$

Now, using Equation 8 in [2]:

$$I((K,T);(Y,T)) \leq I((X;T),(Y;T)) \tag{9}$$

Putting Equation 1 and 2 in Equation 3, we get:

$$I(K;Y|T) \leq I(X;Y|T) \tag{10}$$

This means that $I(K;Y|T)$ establishes a lower bound on $I(X;Y|T)$ and it suits us best if we minimize $I(K;Y|T)$, which can be achieved by not using the key K to transform X to Y. We need to use a *separate process* to minimize $I(X;Y|T)$ that could possibly rely on a different key K'. Now, given that both X and Y have the same domain, the mapping of X to Y ($g(X) = Y$) needs to be a bijection, specifically a 1-way permutation, where it is hard to compute g^{-1} [3] without any side information. g needs to be a bijection because there has to be a one-to-one correspondence between X and Y, which are memory addresses. A compute and storage-efficient version of this is precisely defined to be a *block cipher* !!!

References

[1] R. Spreitzer and T. Plos, "Cache-access pattern attack on disaligned aes t-tables," in *International Workshop on Constructive Side-Channel Analysis and Secure Design*. Springer, 2013, pp. 200–214.

[2] E. de Chérisey *et al.*, "Best information is most successful," *CHES*, 2019.

[3] L. Batina *et al.*, "Mutual information analysis: a comprehensive study," *J. of Cryptology*, 2011.