NP-Hardness of Key Recovery / Frequency Analysis in the Chimera Cipher

Amartya Mathew

March 23, 2025

Abstract

This paper describes a formally proof that recovering the secret mapping (or performing effective "frequency analysis") in the Chimera symmetric algorithm, that assigns multiple substitutions per character and randomly inserts dummy symbols is NP-hard. This proof is via a polynomial-time reduction from the well-known NP-complete problem *Exact Cover*.

1 Introduction

Consider the encryption scheme $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ over the ASCII alphabet $\Sigma = \{0, 1, \ldots, 127\}$. Each character $c \in \Sigma$ is mapped to a set of numeric substitutions $\mathbf{K}(c) \subset \{128, \ldots, N\}$, where $\left|\mathbf{K}(c)\right| = m$. Additionally, a dummy set $\mathbf{D} \subset \{128, \ldots, N\}$ of disjoint symbols can be inserted randomly into the ciphertext. The goal is to show that *recovering the plaintext* (or the mapping from ciphertext symbols to plaintext characters) is NP-hard when performed via "frequency analysis" or any combinatorial method that tries to label each ciphertext symbol as "dummy" or "character-substitution".

2 Preliminaries

2.1 Exact Cover

We recall the *Exact Cover* problem, which is a well-known NP-complete problem (Karp, 1972).

Definition 1 (Exact Cover)**.** *An* Exact Cover *instance is given by:*

$$U = \{u_1, u_2, \ldots, u_n\} \quad (\textit{universe}) \quad \textit{and} \quad \mathcal{S} = \{S_1, S_2, \ldots, S_m\},$$

where each $S_j \subseteq U$. The decision question is:

Does there exist a subfamily $\mathcal{S}' \subseteq \mathcal{S}$ such that $\bigcup_{S_j \in \mathcal{S}'} S_j = U$ and $S_j \cap S_{j'} = \varnothing$ for all $j \neq j'$?

Equivalently, can we cover each element of U exactly once by subsets in $\mathcal{S}'$?

Exact Cover is NP-complete because one can reduce other NP-complete problems, like 3D-Matching or Subset Sum, to it in polynomial time.

2.2 Frequency Analysis Problem (FREQ-ANAL)

We model the frequency-analysis or mapping-recovery problem for the multi-substitution cipher as follows:

Definition 2 (FREQ-ANAL Problem)**.** *An instance of the* FREQ-ANAL *problem consists of:*

- *A ciphertext* $C = (x_1, x_2, \ldots, x_L)$*, each* $x_i \in \{128, \ldots, N\}$*.*
- *The knowledge that each character* $c \in \Sigma$ could *map to one of* m *possible integers from* $\{128, \ldots, N\}$*, but the exact sets* $\mathbf{K}(c)$ *are* unknown *(or at least the mapping of ciphertext symbols to those sets is unknown).*
- *A dummy set* $\mathbf{D} \subset \{128, \ldots, N\}$ *which may be interspersed arbitrarily among real ciphertext symbols, but the exact positions and membership are also unknown.*

Decision Version: *"Is there a consistent assignment of each symbol in* C *to either (a) some character's substitution set or (b) the dummy set* $\mathbf{D}$*" such that the resulting plaintext is valid under the scheme's rules (i.e., each real symbol must come from exactly one character's substitution set and no two characters share a substitution)?*

Informally, the *FREQ-ANAL* problem asks us to figure out which symbols are real vs. dummy, and which real symbol belongs to which character—an attacker's essential goal when performing a combinatorial frequency analysis.

3 NP-Hardness Proof

We show a polynomial-time reduction from *Exact Cover* to *FREQ-ANAL*. Hence, solving FREQ-ANAL in polynomial time would imply a polynomial-time solution to Exact Cover, contradicting NP-completeness (unless $P = NP$).

Theorem 1. *FREQ-ANAL (Definition 2) is NP-hard.*

Proof. We give a many-to-one reduction from Exact Cover to FREQ-ANAL.
Exact Cover Instance: Let $\langle U, \mathcal{S} \rangle$ be an instance of Exact Cover, where

$$U = \{u_1, u_2, \ldots, u_n\}, \quad \mathcal{S} = \{S_1, S_2, \ldots, S_m\}.$$

We wish to decide if there is a subfamily $\mathcal{S}' \subseteq \mathcal{S}$ that covers U exactly once.
Construct FREQ-ANAL Instance: We build a ciphertext C and specify a set of possible substitution blocks so that any valid labeling of C corresponds to picking an exact cover.

(i) *Ciphertext Layout*: We create a ciphertext C with length at least n. For each universe element $u_i \in U$, we include a "slot" in C to represent u_i. These slots become symbols drawn from $\{128, \ldots, N\}$.

(ii) *Subset Characters*: For each subset $S_j \in \mathcal{S}$, define a *character* C_j. The set $\mathbf{K}(C_j)$ includes exactly one symbol for each $u_i \in S_j$. Possibly we add some filler symbols to make the set size up to m. Thus, if we pick "character" C_j for a slot u_i, it implies $u_i \in S_j$.

(iii) *Dummy Symbols*: We allow $\mathbf{D} \subset \{128, \ldots, N\}$ for random insertion. These dummy symbols can appear in the ciphertext but do not correspond to any universe element.

(iv) *Valid Labeling*: A solution to FREQ-ANAL is an assignment of each symbol $x_k \in C$ to either:

- *dummy* (ignore it), or
- *character* C_j (meaning x_k is one of the symbols in $\mathbf{K}(C_j)$).

For it to yield a consistent plaintext, each universe position u_i must be covered exactly once by some subset S_j with $u_i \in S_j$. This exactly matches the notion of an *exact cover*.

Correctness of the Reduction:

- If there exists an *exact cover* $\mathcal{S}' \subseteq \mathcal{S}$, we can label each slot u_i in the ciphertext by the unique subset $S_j \in \mathcal{S}'$ such that $u_i \in S_j$. This corresponds to choosing the character C_j whose substitution set covers u_i. Thus, the labeling explains all symbols without overlap. Any leftover symbols can be marked dummy.
- If there is a valid labeling in the FREQ-ANAL instance, it means we can cover each slot u_i exactly once by picking a subset-character C_j. This subfamily of subsets $\mathcal{S}'$ is by construction an *exact cover* of U.

Complexity: The construction is polynomial-time in $\langle U, \mathcal{S} \rangle$. Hence, if we could solve FREQ-ANAL in polynomial time, we could solve Exact Cover in polynomial time by transforming $\langle U, \mathcal{S} \rangle$ to an instance of FREQ-ANAL, then solving it.

Since Exact Cover is NP-complete, FREQ-ANAL must be NP-hard. This completes the proof of Theorem 1. □

4 Conclusion

By Theorem 1, the problem of *frequency analysis* or mapping recovery in the multi-substitution, dummy-insertion scheme is NP-hard. Intuitively, an attacker trying to label ciphertext symbols to recover the original plaintext is solving a puzzle at least as difficult as Exact Cover. Therefore, under widely believed assumptions ($P \neq NP$), no efficient (polynomial-time) algorithm can, in general, break the scheme by purely combinatorial means, implying *computational security*.