Pro dešifrování šifry ji musíme provést zpětné inženýrství na zdrojovém kódu generátoru. Proto nejdříve musíme začít tím, jak zašifrovaný text převést na bajty, které pak můžeme dešifrovat.

V kódu generátoru si můžeme všimnout, že zašifrovaný bajt je převeden na slabiku, která má stejný index jako hodnota bajtu. Protože žádná ze slabik v zašifrovaném textu není prefixem jiné slabiky, můžeme zašifrovaný řetězec převést na bajty tak, že vezmeme podřetězec, vyzkoušíme, jestli odpovídá nějaké slabice, a buď se posune, nebo prodlouží zkoušený řetězec (pro implementaci viz funkci `decode_string()`). Zároveň musíme ignorovat všechny znaky kromě znaků abecedy. V mé implementaci, kde využívám hashmapu, má tato část složitost $\mathcal{O}(n)$, protože nejvíce to provede $\mathcal{O}(2n)$ operací.

Když už máme bajty, s kterými pracovat, musíme zjistit, jak je generátor získal. V kódu generátoru můžeme vidět, že ze tří bajtů vytvoří čtyři bajty, které pak jsou indexy slabik ve výsledném textu. Mezi těmito bajty jsou pak bity původních bajtů pravidelně "promíchané", proto nám stačí akorát ty bity vrátit na své místo. Ze zdrojového kódu zjistíme, že pro původní bajty `X, Y, Z` platí:

```
X = ((B & 0b110000) >> 4) + ((A & 0b111111) << 2)
Y = ((C & 0b111100) >> 2) + ((B & 0b1111) << 4)
Z = D + ((C & 0b11) << 6)
```

kde `A, B, C, D` je označení čtveřice zašifrovaných bajtů.

V kódu též řeší případ, kdy zbývá méně než čtyři bajty. Protože tohle je vyřešeno tak, že jen nepřičítá část s bajtem, který chybí, stačí nám jenom kontrolovat počet nezpracovaných bajtů, aby nedošlo k přetečení.

Tahle část provede lineárně operací, proto časová složitost bude $\mathcal{O}(n)$.

Ještě ale nemáme zcela dešifrováno. V kódu generátoru vidíme, že všechny bajty kromě posledního jsou předtím zašifrovány operátorem XOR. Protože však byl poslední bajt zachován, použijeme ho jako klíč a odzadu budeme provádět XOR. Pak už nám stačí převést bajty na řetězec znaků a to vrátit.

Tahle část má též časovou složitost $\mathcal{O}(n)$, proto celý algoritmus na dešifrování zprávy má časovou složitost $\mathcal{O}(n)$. Zároveň prostorová složitost je též $\mathcal{O}(n)$.

Výsledný dešifrovaný text zprávy je:

```
24.4.1719 Plavba zdala se byti nekonecnou. Nase obavy byly ale zazehnany kdyz
    jsme dnes spatrili zemi!
25.4.1719 Z cista jasna nas prekvapila boure a nahnala lod na skaliska u ostrova
    . Lod je poskozena a dalsi plavba vyzaduje opravy trupu. Nekolik dni zde
    budeme nuceni zustat.
27.4.1719 Ostrov je plny divych tvoru. Neboji se cloveka a utoci s ohromnou
    silou. Jen stezi jsem vyvazl z naseho setkani s nimi. Jeden z namorniku to
    stesti vsak nemel.
28.4.1719 Odhalil jsem duvod agresivity ostrovnich selem. Chrani sva cerstve
    narozena mladata!
30.4.1719  Po nekolika pokusech se nasi posadce podarilo spratelit se s touto
    divou zveri. Stacilo je nalakat na maso a bobule rostliny zvane fiksus.
1.5.1719  Opravy lodi jsou skoro hotovy. Jeste par dni a plavba bude moci
    pokracovat.
3.5.1719  V noci nas prekvapili nejaci lide. Netusime kdo to byl, nevideli jsme
    je. Zabili straze a vypalili tabor. My co jsme prezili se schovavame v
    jeskyni a doufame, ze nas nenajdou.
```