گزارش پروژه رمزنگاری کلاسیک

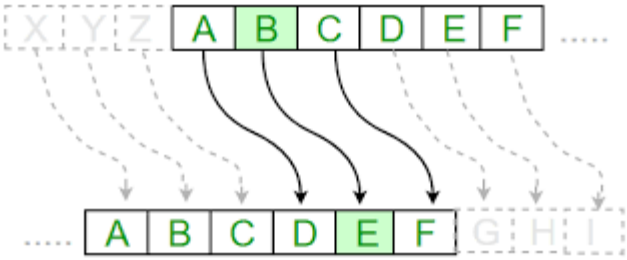
پیاده سازی و تحلیل رمز سزار
(Caser cipher)

نام و نام خانوادگی دانشجو:محمد امین نامور آقچای شماره دانشجویی:40423743

تاریخچه رمز سزار:

گایوس ژولیوس سزار، ژنرال و سیاستمدار معروف رومی، در دوران روم باستان رمز سزار را اختراع کرد. رمز سزار نوعی ساده اما موثر از رمزنگاری است که با جابه‌جایی هر حرف از الفبا به تعداد معینی به سمت راست کار می‌کند. این کار پیام را برای هر کسی که مقدار جابه‌جایی یا کلید را نمی‌داند، غیرقابل خواندن می‌کند.

سزار از رمز سزار برای محافظت از ارتباطات خود در طول نبردهای نظامی استفاده می‌کرد. او از رمز برای ارسال پیام‌های مخفی به ژنرال‌ها و فرماندهان خود استفاده می‌کرد و آن‌ها از همان رمز برای ارسال پاسخ‌های خود به او استفاده می‌کردند. این امر به اطمینان از این‌که دشمن نمی‌تواند پیام‌های او را رهگیری و بخواند، کمک کرد.



مباحث ریاضی رمز سزار:

در هر زبان با توجه به تعداد حروف مد های مختلفی خواهیم داشت مثلا برای زبان انگلیسی مقدار مد برابر 26 و در زبان فارسی این مقدار برابر 32 خواهد بود

اگر حروف الفبای انگلیسی را با اعداد 0 تا 25 نشان دهیم:

$A = 0, B = 1, C = 2, ..., Z = 25$

برای رمزنگاری این فرمول استفاده میشود:

$E(x) = (x + k) \bmod 26$

برای رمزگشایی نیز از این فرمول استفاده میکنیم:

$D(y) = (y - k) \bmod 26$

توابع رمز گشایی و رمز نگاری در کد:

با استفاده از فرمول های ریاضی که برای رمز گشایی و رمز نگاری گفتیم توابع رمز نگاری و رمزگشایی در کد را پیاده سازی کردیم
به اینصورت که این توابع یک متن و یک شیفت مشخص را دریافت کرده و با استفاده از فرمول های گفته شده متن را رمزنگاری کرده یا
با استفاده از شیفت مشخص شده متن را رمز نگاری میکند

Brute-Force:

با توجه به این تعداد شیفت ها مثلا در زبان انگلیسی 26 حالت دارد پیچیدگی زمانی برابر:

$O(26 \times n)$

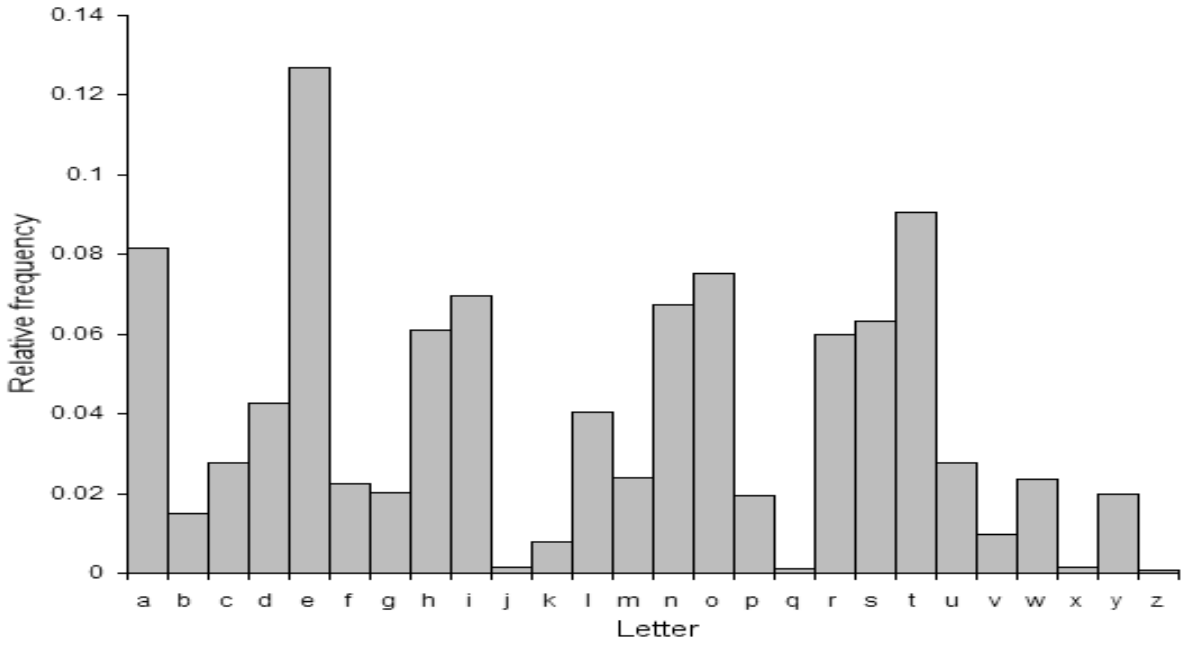
که با توجه به کوچک بودن عدد 26 این مقدار ناچیز است و میتوان گفت که :

Brute Force عملی است.

برای تابع آن نیز در کد با استفاده از یک حلقه شیفت های مختلف را رو متن انجام داده و آنها را در یک آرایه ذخیره میکنیم

تابع امتیازدهی به جملات در کد:

با استفاده از جدول فراوانی فراوانی هر حرف انگلیسی در جملات را پیدا کرده و با توجه به فراوانی ها به هر جمله یک امتیاز میدهیم



تابع رمزگشایی خودکار یا رمزگشایی بدون اطلاع از شیفت در کد:

در این تابع برای یک رمز شیفت های مختلف را چک میکنیم و با استفاده از تابع امتیازدهی به جملات جمله ای که بیشترین امتیاز ممکن را داشته باشد جمله رمز گشایی شده خواهد بود.

تابع برای بدست آوردن درصد موفقیت رمزگشایی خودکار در کد:

در این تابع جمله هایی رندوم با طول مشخص ایجاد کرده و آنهارا با یک شیفت مشخص رمزنگاری میکنیم
از طرفی در تابع امتیاز دهی به جملات بهترین شیقت را پیدا میکنیم

اگر شیفت ها با هم برابر بود تعداد موفقیت هارا زیاد میکنیم

و در نهایت تعداد کل موفقیت ها به کل را محاسبه میکنیم

و آنهارا برای نمایش بصری داده های آماری در داخل نمودار نشان میدهیم

نتیجه گیری:

در این پروژه، رمز سزار به عنوان یکی از ساده‌ترین روش‌های رمزنگاری تاریخی، پیاده‌سازی و مورد بررسی قرار گرفت.
نتایج به دست آمده نشان می‌دهد که این روش رمزنگاری به دلیل فضای کلید بسیار کوچک در برابر حملات به شدت آسیب پذیر است.
با استفاده از روش تحلیل فراوانی حروف و امتیازدهی به متون رمزگشایی شده، توانستیم بدون داشتن کلید متن اصلی را با دقت بالایی بازیابی کنیم و در نمودار ها مشاهده شد که هرچقدر طول متن بیشتر باشد دقت رمز شکنی بیشتر خواهد شد.
این پروژه به خوبی نشان می‌دهد که چرا رمزهای جانشینی ساده مانند رمز سزار برای استفاده در دنیای امروز امن نیستند و چگونه ترکیب روش های آماری و محاسباتی میتواند آنهارا شکست دهد.