



دانشکده مهندسی کامپیوتر

امنیت داده و شبکه

تمرین ۱

دکتر میزانیان

آریا جلالی — ۹۸۱۰۵۶۶۵

۱۳ فروردین ۱۴۰۲

## فهرست مطالب

|    |   |  |
|----|---|--|
| ۴  | ۱ سوال ۱                                  |  |
| ۴  | ۱.۱ TCP Sequence Prediction Attack        |  |
| ۴  | ۲.۱ Reflector Attack                      |  |
| ۵  | ۳.۱ Masquerading Attack                   |  |
| ۵  | ۴.۱ Fault Injection Attack                |  |
| ۶  | ۵.۱ Man In The Middle Attack              |  |
| ۷  | ۲ سوال ۲                                  |  |
| ۷  | ۱.۲ ECB                                   |  |
| ۷  | ۲.۲ OFB                                   |  |
| ۸  | ۳.۲ PCBC                                  |  |
| ۹  | ۴.۲ CTR                                   |  |
| ۹  | ۳ سوال ۳                                  |  |
| ۱۱ | ۴ سوال ۴                                  |  |
| ۱۱ | ۵ سوال ۵                                  |  |
| ۱۱ | ۱.۵ خاصیت بهمنی اکید                      |  |
| ۱۱ | ۲.۵ خاصیت تمامیت                          |  |
| ۱۱ | ۳.۵ Random Cipher                         |  |
| ۱۲ | ۴.۵ بررسی خواص برای مدل داده شده          |  |
| ۱۲ | ۱.۴.۵ خاصیت بهمنی اکید                    |  |
| ۱۳ | ۲.۴.۵ خاصیت تمامیت                        |  |
| ۱۳ | ۳.۴.۵ Random Cipher                       |  |
| ۱۳ | ۶ سوال ۶                                  |  |
| ۱۴ | ۷ سوال ۷                                  |  |
| ۱۴ | ۱.۷ رمزگذاری با Caesar                    |  |
| ۱۵ | ۲.۷ رمزگذاری با رمز جایگزینی              |  |
| ۱۵ | ۳.۷ رمزگذاری با Vigenere                  |  |
| ۱۵ | ۱.۳.۷ کلید 'SUT'                          |  |
| ۱۶ | ۲.۳.۷ کلید 'SHARIFUNIVERSITYOFTECHNOLOGY' |  |
| ۱۶ | ۳.۳.۷ تاثیر طول کلید                      |  |
| ۱۷ | ۴.۷ رمزگذاری با Playfair                  |  |
| ۱۷ | ۵.۷ رمزگشایی جایگشتی                      |  |
| ۱۷ | ۱.۵.۷ تنظیمات                             |  |
| ۱۸ | ۲.۵.۷ رمزگشایی                            |  |
| ۱۸ | ۶.۷ رمزگشایی vigenere                     |  |

| سوال ۸ |                    | ۱۹ |
|--------|--------------------|----|
| ۱.۸    | پیدا کردن طول کلید | ۱۹ |
| ۲.۸    | بدست آوردن کلید    | ۲۰ |
| ۳.۸    | رمزگشایی           | ۲۰ |
| ۱.۳.۸  | متن با کلید 'دروغ' | ۲۱ |
| ۴.۸    | متن با کلید 'مف'   | ۲۱ |

## ۱ سوال ۱

## ۱.۱ TCP Sequence Prediction Attack

در این حمله مهاجم با گوش دادن به پیام‌های رد و بدل شده بین client و server سعی می‌کند شناسه پکت‌های TCP را حدس بزند. پس از درست حدس زدن این شناسه مهاجم می‌تواند بسته‌های ساختگی برای client بفرستد و آن‌ها را طوری طراحی کند که انگار از سمت Server آمده‌اند. مهاجم می‌تواند با گوش دادن به مکالمات رد و بدل شده شناسه را حدس بزند و پس از حدس شناسه یک مسابقه بین پکت‌های ارسال شده توسط سرور و مهاجم آغاز می‌شود. دقت کنید در اکثر مواقع، پکت‌های سرور زودتر از پکت‌های مهاجم می‌رسند. به همین دلیل اکثر اوقات یک حمله محروم‌سازی از سرویس (DOS) به صورت موازی نیز انجام می‌شود.

## روش‌های مقابله

۱. انتخاب اولین شناسه با استفاده از الگوریتم‌های تصادفی باعث می‌شود مهاجم نتواند شناسه را حدس بزند و حمله را آغاز کند.

۲. رمزنگاری کردن اطلاعات رد و بدل شده. در صورتی که پکت‌های در حال رد و بدل را رمزنگاری کنیم، این اجازه را به مهاجم نمی‌دهیم که بتواند شناسه پکت‌ها را بخواند و حتی در صورت درست حدس زدن، امکان تغییر و دستکاری را ندارد.

۳. استفاده از دیوار آتش این اجازه را به ما می‌دهد تا جلوی بسته‌هایی که از خارج شبکه با IP داخلی را بگیریم و نگذاریم بسته‌های جعلی از طریق مهاجم به کاربر برسد.

۴. استفاده از اطلاعات لایه‌های پروتکل زیرین. با استفاده از این اطلاعات مانند متادیتا فرستاده پکت، فاصله زمانی بین پکت‌ها می‌توانیم متوجه حقیقی یا جعلی بودن پکت فرستاده شده بشویم و آن‌ها را drop کنیم. البته دقت کنید در اکثر اوقات دسترسی به این اطلاعات ممکن نیست یا بسیار دشوار است.

در این حمله مهاجم می‌تواند پکت‌های ساختگی از طرف سرور برای کلاینت بفرستد. در این حمله اصل صحت (Integrity) و به طور خاص‌تر صحت منبع نقض شده است. دقت کنید می‌توان ادعا کرد اصل محرمانگی (Confidentiality) نقض شده است، اما دقت کنید در این نوع حملات مهاجم پیام‌های ارسال شده توسط کلاینت را دریافت نمی‌کند و تنها می‌تواند پیام‌های جعلی برای یک گیرنده بفرستد.

## ۲.۱ Reflector Attack

این حمله یک حمله محروم‌سازی از سرویس توزیع‌یافته (DDOS) است. در این نوع حمله با استفاده از تکنیک IP address spoofing مهاجم از طریق چندین دستگاه، IP جعلی تولید می‌کند و درخواستی به یک سرور آسیب‌پذیر مانند یک سرور DNS می‌فرستد. این سرورها معمولاً پاسخی به مراتب حجیم‌تر از درخواست فرستاده شده می‌فرستند. و به نحوی باعث تقویت سیگنال مزاحم مهاجم می‌شوند. با استفاده از این تکنیک مهاجم می‌تواند با فرستادن ترافیک‌های کوچک، حجم زیادی از داده را به سمت IP جعل شده هدایت کند و باعث مختل شدن آن شود.

## روش‌های مقابله

۱. استفاده از load balancing. در این روش پکت‌های ارسال شده توسط سرورها بین چندین سرور پخش می‌شود و مهاجم نمی‌تواند یک سرور به خصوص را مورد حمله قرار دهد.
  ۲. استفاده از دیوار آتش این اجازه را به ما می‌دهد تا ترافیک‌های دریافتی از سرورهای آسیب‌پذیر را محدود کنیم و از یک حدی به بعد این پکت‌ها را drop کنیم.
  ۳. با استفاده از فیلترهای Regex می‌توانیم پیام‌هایی که از یک فرم خاص پیروی می‌کنند (مانند پیام‌های DNS) تشخیص دهیم و آن‌ها را drop کنیم.
- همانطور که بالاتر به آن اشاره شد، این حمله یک نوع خاص حمله محروم‌سازی از سرویس است که همانطور که از اسم آن مشخص است اصل Availability را نقض می‌کند.

## ۳.۱ Masquerading Attack

در این حمله مهاجم با استفاده از یک هویت جعلی به سیستم‌ها یا شبکه‌های یک کاربر دسترسی پیدا می‌کند و با استفاده از این هویت به اعمالی می‌پردازد که محدود به آن کاربر است. در اکثر اوقات از این حمله برای انجام اعمال غیرقانونی با استفاده از نام و هویت فرد دیگری انجام می‌شود.

## روش‌های مقابله

۱. استفاده از روش‌های شناسایی پیشرفته‌تر. به عنوان مثال می‌توانیم از روش‌های au- multi-factor thentication استفاده کنیم که فرایند جعل هویت را به مراتب دشوارتر می‌کند.
  ۲. استفاده از امضاهای دیجیتال باعث جلوگیری از اجرای نرم‌افزارهایی می‌شود که دارای یک امضا از طرف یک منبع معتبر نیستند.
  ۳. استفاده از اصل privilege least باعث می‌شود هر کاربر تنها در حد نیاز دسترسی به اطلاعات حساس داشته باشد، اینکار باعث می‌شود مهاجم نتواند با جعل هویت هرکس بتوان به مقاصد شوم خود برسد و تنها افراد خاصی دسترسی‌های مورد نیاز را داشته باشند.
  ۴. بررسی فعالیت‌های کاربر می‌تواند باعث شناسایی رفتارهای مشکوک شود. به عنوان مثال در صورت لوگین کردن یک کاربر از یک مکان جدید با IP جدید ممکن است نشانه این نوع حمله باشد.
- در این حمله چون هویت یک فرد جعل می‌شود و دسترسی به اطلاعات کاربر به افراد غیرمجاز داده می‌شود هم اصل Integrity و هم اصل Confidentiality نقض می‌شود.

## ۴.۱ Fault Injection Attack

در حملات FIA مهاجم سعی می‌کند به صورت نرم‌افزاری یا سخت‌افزاری حالت خطا در سیستم ایجاد کند تا سیستم به صورت غیرمنتظره رفتار کند. در این بازه مهاجم می‌تواند با بررسی نوع رفتار سیستم ضعف‌هایی در آن پیدا کند و از آن‌ها سواستفاده کند.

در نوع فیزیکی FIA مهاجم می‌تواند با تغییر سیگنال کلاک یا تغییر ولتاژ پردازنده یا استفاده از تشعشعات متفاوت خطا وارد سیستم کند و سیستم را وادار به رفتارهای غیرمنتظره کند.

در نوع نرم‌افزاری این حمله مهاجم با استفاده از تکنیک‌های متفاوت مانند اختلال حافظه با Overflow کردن بافر سیستم با استفاده از فرستادن مقدار زیادی اطلاعات یا اجرای کد مخرب با گول زدن پردازنده، سیستم را وادار به رفتار غیرمنتظره می‌کند.

#### روش‌های مقابله

۱. استفاده از تصادفی سازی فضای حافظه یا (ASLR) در این روش فضای حافظه به صورت تصادفی چیده می‌شود و مهاجم نمی‌تواند به راحتی فضای ادرس را پیشبینی کند.
  ۲. محدود کردن دسترسی به سخت‌افزار سیستم به افراد معدود.
  ۳. استفاده از نرم‌افزارهای متفاوت برای اعمال Code review به صورت روتین برای تشخیص آسیب‌پذیری‌های ممکن.
- در این نوع حملات اصل Integrity به دلیل رفتار غیرمنتظره سیستم که ممکن است باعث تخریب داده شود و اصل Availability نقض می‌شوند.

### ۵.۱ Man In The Middle Attack

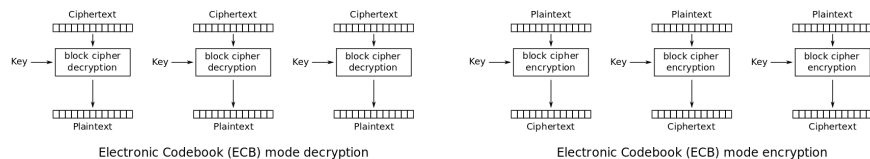
در این نوع حملات یک فرد سوم بین ارتباط دو نفر قرار می‌گیرد و آن‌ها را فریب می‌دهد که به طور مستقیم در حال ارتباط با یکدیگر هستند. یک نوع محبوب از این حمله eavesdropping است که در آن مهاجم دو ارتباط مستقل به ۲ طرف برقرار می‌کند. هر دو طرف ارتباط فکر می‌کنند به طور مستقیم با یکدیگر در حال ارتباط هستند، در صورتی که تمام این ارتباط توسط فرد سوم که مهاجم است کنترل می‌شود. مهاجم می‌تواند تنها بسته‌های ارسالی را شنود کند و یا آن‌ها را دستکاری کند و با جعل هویت به فرد دیگر بفرستد.

#### روش‌های مقابله

۱. استفاده از ارتباط رمزنگاری شده
  ۲. استفاده از گواهی دیجیتال. استفاده از این گواهی‌ها می‌تواند به تایید صحت هویت فرد مقابل کمک کند و نگذارد فرد سومی هویت او را جعل کند.
  ۳. استفاده از کلیدهای عمومی که توسط سازمان‌های معتبر صادر شده‌اند. این روش جلوی استفاده از کلیدهای تقلبی توسط افراد مهاجم را می‌گیرد.
- در این نوع حمله هم شاهد جعل هویت هستیم و هم شاهد تغییر و دستکاری داده که نشان می‌دهد اصول Confidentiality و Integrity در آن نقض می‌شوند.

## سوال ۲

## ۱.۲ ECB



شکل ۲: فرایند رمزگشایی

شکل ۱: فرایند رمزگذاری

الف) همانطور که از تصویر بالا مشخص است، خطا در یک بیت از ciphertext تنها تاثیر مستقیم بر روی plaintext بلوک دارای آن بیت خواهد داشت. و تاثیری روی بقیه بلوکها نخواهد گذاشت.

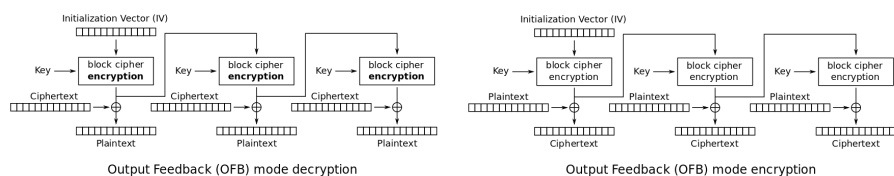
ب) از آنجایی که وابستگی ای بین بلوکها چه در زمان رمزگذاری و رمزگشایی وجود ندارد، از دست رفتن یک بلوک تاثیری بر روی بلوکهای دیگر ندارد.

پ) همانطور که پیشتر به آن اشاره شد به دلیل عدم وجود وابستگی بین بلاکها، امکان رمزگشایی و رمزگذاری موازی وجود دارد.

ت) به همان دلایلی که در بخش قبل به آنها اشاره شد (عدم وجود وابستگی بین بلوکها) امکان رمزگشایی موازی وجود دارد.

ث) چون وابستگی ای بین بلوکها وجود ندارد، امکان رمزگشایی و رمزگذاری یک بلوک دلخواه وجود دارد.

## ۲.۲ OFB



شکل ۴: فرایند رمزگشایی

شکل ۳: فرایند رمزگذاری

الف) ciphertext هر بلوک تنها تاثیر مستقیم بر روی plaintext همان بلوک دارد و خطا در انتقال یک بیت از متن رمز شده تنها متن رمزگشایی شده همان بلوک را تحت تاثیر قرار می دهد.

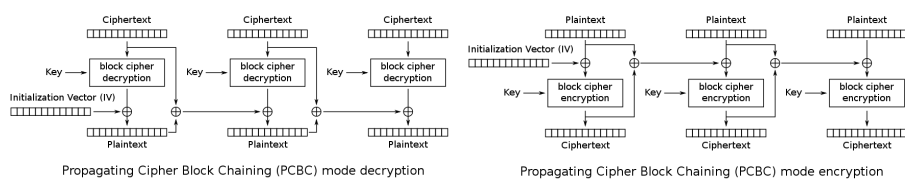
ب) همانطور که در تصویر مشخص است، یک بلوک متن رمز شده تنها در همان بلوک استفاده می شود و حذف آن تاثیری روی فرایند رمزگشایی بقیه بلوکها نمی گذارد.

پ) در صورتی که از قبل Initialization vector با استفاده از key به تعداد دفعات لازم encrypt شده باشد، می‌توانیم با آمدن متن ساده آن‌ها را به صورت موازی رمزگذاری کنیم.

ث) از آنجایی که فرایند رمزگشایی و رمزگذاری دقیقاً عین هم است، می‌توان با انجام عملیات رمزگذاری Initialization vector در ابتدا با استفاده از کلید فعالیت رمزگشایی را با آمدن متن‌های رمز به صورت موازی انجام داد.

ث) با توجه به توضیحات بالا، اگر از قبل Initialization vector با استفاده از کلید رمزگشایی شده باشد (به تعداد دفعات لازم) می‌توان بلوک دلخواه را رمزگشایی کرد.

## ۳.۲ PCBC



شکل ۵: فرایند رمزگذاری

شکل ۶: فرایند رمزگشایی

الف) همانطور که از شکل مشخص است خطا در بیت یک متن رمز به طور وضوح بر روی متن عادی همان بلاک تاثیر خواهد داشت، ولی چون از این متن رمز در رمزگشایی مرحله بعد نیز استفاده می‌شود، متن بلوک بعدی نیز تحت تاثیر قرار خواهد گرفت. با تحت تاثیر قرار گرفتن بلوک بعدی، به دلیل XOR شدن متن ساده با متن رمز و استفاده از نتیجه در بلوک بعد، این خطا تا انتها propagate می‌شود و تمامی بلوک‌های پس از بلوک دارای متن رمز خطا دار دچار مشکل می‌شوند.

ب) به دلیل وابستگی متن‌های رمزگشایی شده هر بلوک به متن‌های رمزدار بلوک‌های قبل، با از دست دادن یک بلوک فرایند رمزگشایی تمام بلوک‌ها دچار اختلال می‌شود.

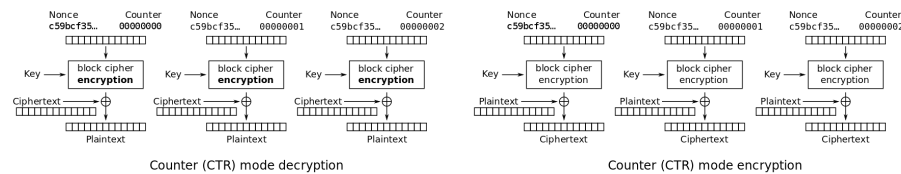
پ) همانطور که از تصویر مشخص است، هر بلاک برای شروع فرایند رمزگذاری نیاز به XOR متن بلوک قبل و نتیجه رمزگذاری آن دارد، این وابستگی اجازه رمزگذاری موازی را به ما نمی‌دهد.

ت) در فرایند رمزگشایی نیز برای بدست آوردن نتیجه نهایی نیاز به Plaintext بلوک قبل داریم و نمی‌توانیم عملیات را به صورت موازی انجام دهیم.

ث) به دلیل وابستگی نتیجه رمزگشایی یک بلوک دلخواه به نتیجه تمامی بلوک‌های قبل از خود، نمی‌توان یک بلوک دلخواه را بدون رمزگشایی بلوک‌های قبل از آن رمزگشایی کرد.



## ۴.۲ CTR



شکل ۸: فرایند رمزگشایی

شکل ۷: فرایند رمزگذاری

الف) از آنجایی که در فرایند رمزگشایی نتیجه بردار Initialization تنها در انتها با متن رمز XOR می‌شود، تنها بیت متناظر با بیت خطا دچار خطا می‌شود.

ب) به دلیل عدم وجود وابستگی بین بلوک‌ها چه در رمزگشایی و چه در رمزگذاری از دست دادن یک بلوک متن رمز شده تنها همان بلوک را تحت تاثیر قرار خواهد داد.

پ) به دلیل عدم وجود وابستگی بین Plaintext ها در فرایند رمزگذاری و دانستن مقدار offset(counter) و initialization (Nonce) می‌توانیم عملیات رمزگذاری را به صورت موازی انجام دهیم.

ت) همانند بخش قبل وابستگی‌ای بین متون رمز وجود ندارد و می‌توان عملیات رمزگشایی را به صورت موازی انجام داد.

ث) چون رمزگشایی هر بلوک وابستگی خارجی‌ای ندارد و تنها نیاز به دانستن Offset (Counter) و Nonce داریم، می‌توانیم بلوک دلخواه را رمزگشایی کنیم.

## ۳ سوال ۳

الف) این رمزنگاری در دسته رمزهای جانشینی چندالفبایی قرار می‌گیرد، زیرا چند حرف (در این الگوریتم ۲) با چند حرف دیگر (در این الگوریتم باز ۲) جایگزین می‌شوند.

ب) مزایا

۱. این الگوریتم به دلیل سادگی طراحی و رمزگذاری و رمزگشایی در مقاصد زیادی استفاده می‌شود.
۲. این الگوریتم در دسته رمزهای جانشینی چندالفبایی قرار می‌گیرد و هر دو حرف را جایگزین می‌کند که آن را نسبت به الگوریتم‌های جانشینی تک‌الفبایی امن‌تر می‌کند.

معایب

۱. در این الگوریتم نمی‌توان white space و یا نشانه‌های نگارشی را رمزنگاری کرد و خواندن متن رمزگشایی شده سخت است.

۲. هر حرف فقط می‌تواند با حروفی که در ستون یا سطر با آن اشتراک دارند جایگزین شود، که باعث می‌شود فضای حالت به اندازه کافی بزرگ نشود و بتوان با حملاتی مانند Brute Force یا تحلیل فرکانسی آن را رمزگشایی کرد.

(پ) در ابتدا یک حرف از حروف انگلیسی حذف می‌شود تا ۲۵ حرف باقی بماند. در ادامه یک ماتریس ۵ در ۵ ساخته می‌شود که در ابتدا حرف‌های یکتا کلید به ترتیب در آن وارد می‌شوند و در ادامه حروف استفاده نشده در کلید به ترتیب الفبایی نوشته می‌شوند. معمولا حرف J حذف می‌شود. در ادامه plaintext به صورت لیستی از جفت حرف‌ها تقسیم می‌شود.

۱. در صورتی که دو حرف یک جفت یکسان باشند یک حرف bogus که معمولا x است بین آن‌ها در plaintext اصلی قرار می‌دهیم.

۲. در صورتی که تعداد حروف plaintext فرد باشد یک حرف اضافه که معمولا z است در انتهای متن اصلی قرار می‌دهیم.

در ادامه هر دو حرف به صورت مقابل رمزگذاری می‌شوند

۱. در صورتی که دو حرف روی یک سطر باشند حرف راست هر حرف (چپ برای رمزگشایی) را جای هر حرف قرار می‌دهیم.

۲. در صورتی که دو حرف روی یک ستون باشند حرف پایین هر حرف (بالا برای رمزگشایی) را جای هر حرف قرار می‌دهیم.

۳. در صورتی که دو شرط بالا برقرار نباشد یک مستطیل محاط بین ۲ حرف تشکیل می‌دهیم و حرفی که روی راس مخالف افقی هر حرف قرار دارد را جای حرف اصلی قرار می‌دهیم.

(ت) فرایند رمزگشایی در عکس زیر قابل مشاهده است.

```

V I C T O
R Y A B D
E F G H K
L M N P Q
S U W X Z

DISYGLYOWLGRYDLYSYGRDTXICO
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040 1041 1042 1043 1044 1045 1046 1047 1048 1049 1050 1051 1052 1053 1054 1055 1056 1057 1058 1059 1060 1061 1062 1063 1064 1065 1066 1067 1068 1069 1070 1071 1072 1073 1074 1075 1076 1077 1078 1079 1080 1081 1082 1083 1084 1085 1086 1087 1088 1089 1090 1091 1092 1093 1094 1095 1096 1097 1098 1099 1100 1101 1102 1103 1104 1105 1106 1107 1108 1109 1110 1111 1112 1113 1114 1115 1116 1117 1118 1119 1120 1121 1122 1123 1124 1125 1126 1127 1128 1129 1130 1131 1132 1133 1134 1135 1136 1137 1138 1139 1140 1141 1142 1143 1144 1145 1146 1147 1148 1149 1150 1151 1152 1153 1154 1155 1156 1157 1158 1159 1160 1161 1162 1163 1164 1165 1166 1167 1168 1169 1170 1171 1172 1173 1174 1175 1176 1177 1178 1179 1180 1181 1182 1183 1184 1185 1186 1187 1188 1189 1190 1191 1192 1193 1194 1195 1196 1197 1198 1199 1200 1201 1202 1203 1204 1205 1206 1207 1208 1209 1210 1211 1212 1213 1214 1215 1216 1217 1218 1219 1220 1221 1222 1223 1224 1225 1226 1227 1228 1229 1230 1231 1232 1233 1234 1235 1236 1237 1238 1239 1240 1241 1242 1243 1244 1245 1246 1247 1248 1249 1250 1251 1252 1253 1254 1255 1256 1257 1258 1259 1260 1261 1262 1263 1264 1265 1266 1267 1268 1269 1270 1271 1272 1273 1274 1275 1276 1277 1278 1279 1280 1281 1282 1283 1284 1285 1286 1287 1288 1289 1290 1291 1292 1293 1294 1295 1296 1297 1298 1299 1300 1301 1302 1303 1304 1305 1306 1307 1308 1309 1310 1311 1312 1313 1314 1315 1316 1317 1318 1319 1320 1321 1322 1323 1324 1325 1326 1327 1328 1329 1330 1331 1332 1333 1334 1335 1336 1337 1338 1339 1340 1341 1342 1343 1344 1345 1346 1347 1348 1349 1350 1351 1352 1353 1354 1355 1356 1357 1358 1359 1360 1361 1362 1363 1364 1365 1366 1367 1368 1369 1370 1371 1372 1373 1374 1375 1376 1377 1378 1379 1380 1381 1382 1383 1384 1385 1386 1387 1388 1389 1390 1391 1392 1393 1394 1395 1396 1397 1398 1399 1400 1401 1402 1403 1404 1405 1406 1407 1408 1409 1410 1411 1412 1413 1414 1415 1416 1417 1418 1419 1420 1421 1422 1423 1424 1425 1426 1427 1428 1429 1430 1431 1432 1433 1434 1435 1436 1437 1438 1439 1440 1441 1442 1443 1444 1445 1446 1447 1448 1449 1450 1451 1452 1453 1454 1455 1456 1457 1458 1459 1460 1461 1462 1463 1464 1465 1466 1467 1468 1469 1470 1471 1472 1473 1474 1475 1476 1477 1478 1479 1480 1481 1482 1483 1484 1485 1486 1487 1488 1489 1490 1491 1492 1493 1494 1495 1496 1497 1498 1499 1500 1501 1502 1503 1504 1505 1506 1507 1508 1509 1510 1511 1512 1513 1514 1515 1516 1517 1518 1519 1520 1521 1522 1523 1524 1525 1526 1527 1528 1529 1530 1531 1532 1533 1534 1535 1536 1537 1538 1539 1540 1541 1542 1543 1544 1545 1546 1547 1548 1549 1550 1551 1552 1553 1554 1555 1556 1557 1558 1559 1560 1561 1562 1563 1564 1565 1566 1567 1568 1569 1570 1571 1572 1573 1574 1575 1576 1577 1578 1579 1580 1581 1582 1583 1584 1585 1586 1587 1588 1589 1590 1591 1592 1593 1594 1595 1596 1597 1598 1599 1600 1601 1602 1603 1604 1605 1606 1607 1608 1609 1610 1611 1612 1613 1614 1615 1616 1617 1618 1619 1620 1621 1622 1623 1624 1625 1626 1627 1628 1629 1630 1631 1632 1633 1634 1635 1636 1637 1638 1639 1640 1641 1642 1643 1644 1645 1646 1647 1648 1649 1650 1651 1652 1653 1654 1655 1656 1657 1658 1659 1660 1661 1662 1663 1664 1665 1666 1667 1668 1669 1670 1671 1672 1673 1674 1675 1676 1677 1678 1679 1680 1681 1682 1683 1684 1685 1686 1687 1688 1689 1690 1691 1692 1693 1694 1695 1696 1697 1698 1699 1700 1701 1702 1703 1704 1705 1706 1707 1708 1709 1710 1711 1712 1713 1714 1715 1716 1717 1718 1719 1720 1721 1722 1723 1724 1725 1726 1727 1728 1729 1730 1731 1732 1733 1734 1735 1736 1737 1738 1739 1740 1741 1742 1743 1744 1745 1746 1747 1748 1749 1750 1751 1752 1753 1754 1755 1756 1757 1758 1759 1760 1761 1762 1763 1764 1765 1766 1767 1768 1769 1770 1771 1772 1773 1774 1775 1776 1777 1778 1779 1780 1781 1782 1783 1784 1785 1786 1787 1788 1789 1790 1791 1792 1793 1794 1795 1796 1797 1798 1799 1800 1801 1802 1803 1804 1805 1806 1807 1808 1809 1810 1811 1812 1813 1814 1815 1816 1817 1818 1819 1820 1821 1822 1823 1824 1825 1826 1827 1828 1829 1830 1831 1832 1833 1834 1835 1836 1837 1838 1839 1840 1841 1842 1843 1844 1845 1846 1847 1848 1849 1850 1851 1852 1853 1854 1855 1856 1857 1858 1859 1860 1861 1862 1863 1864 1865 1866 1867 1868 1869 1870 1871 1872 1873 1874 1875 1876 1877 1878 1879 1880 1881 1882 1883 1884 1885 1886 1887 1888 1889 1890 1891 1892 1893 1894 1895 1896 1897 1898 1899 1900 1901 1902 1903 1904 1905 1906 1907 1908 1909 1910 1911 1912 1913 1914 1915 1916 1917 1918 1919 1920 1921 1922 1923 1924 1925 1926 1927 1928 1929 1930 1931 1932 1933 1934 1935 1936 1937 1938 1939 1940 1941 1942 1943 1944 1945 1946 1947 1948 1949 1950 1951 1952 1953 1954 1955 1956 1957 1958 1959 1960 1961 1962 1963 1964 1965 1966 1967 1968 1969 1970 1971 1972 1973 1974 1975 1976 1977 1978 1979 1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022 2023 2024 2025 2026 2027 2028 2029 2030 2031 2032 2033 2034 2035 2036 2037 2038 2039 2040 2041 2042 2043 2044 2045 2046 2047 2048 2049 2050 2051 2052 2053 2054 2055 2056 2057 2058 2059 2060 2061 2062 2063 2064 2065 2066 2067 2068 2069 2070 2071 2072 2073 2074 2075 2076 2077 2078 2079 2080 2081 2082 2083 2084 2085 2086 2087 2088 2089 2090 2091 2092 2093 2094 2095 2096 2097 2098 2099 2100 2101 2102 2103 2104 2105 2106 2107 2108 2109 2110 2111 2112 2113 2114 2115 2116 2117 2118 2119 2120 2121 2122 2123 2124 2125 2126 2127 2128 2129 2130 2131 2132 2133 2134 2135 2136 2137 2138 2139 2140 2141 2142 2143 2144 2145 2146 2147 2148 2149 2150 2151 2152 2153 2154 2155 2156 2157 2158 2159 2160 2161 2162 2163 2164 2165 2166 2167 2168 2169 2170 2171 2172 2173 2174 2175 2176 2177 2178 2179 2180 2181 2182 2183 2184 2185 2186 2187 2188 2189 2190 2191 2192 2193 2194 2195 2196 2197 2198 2199 2200 2201 2202 2203 2204 2205 2206 2207 2208 2209 2210 2211 2212 2213 2214 2215 2216 2217 2218 2219 2220 2221 2222 2223 2224 2225 2226 2227 2228 2229 2230 2231 2232 2233 2234 2235 2236 2237 2238 2239 2240 2241 2242 2243 2244 2245 2246 2247 2248 2249 2250 2251 2252 2253 2254 2255 2256 2257 2258 2259 2260 2261 2262 2263 2264 2265 2266 2267 2268 2269 2270 2271 2272 2273 2274 2275 2276 2277 2278 2279 2280 2281 2282 2283 2284 2285 2286 2287 2288 2289 2290 2291 2292 2293 2294 2295 2296 2297 2298 2299 2300 2301 2302 2303 2304 2305 2306 2307 2308 2309 2310 2311 2312 2313 2314 2315 2316 2317 2318 2319 2320 2321 2322 2323 2324 2325 2326 2327 2328 2329 2330 2331 2332 2333 2334 2335 2336 2337 2338 2339 2340 2341 2342 2343 2344 2345 2346 2347 2348 2349 2350 2351 2352 2353 2354 2355 2356 2357 2358 2359 2360 2361 2362 2363 2364 2365 2366 2367 2368 2369 2370 2371 2372 2373 2374 2375 2376 2377 2378 2379 2380 2381 2382 2383 2384 2385 2386 2387 2388 2389 2390 2391 2392 2393 2394 2395 2396 2397 2398 2399 2400 2401 2402 2403 2404 2405 2406 2407 2408 2409 2410 2411 2412 2413 2414 2415 2416 2417 2418 2419 2420 2421 2422 2423 2424 2425 2426 2427 2428 2429 2430 2431 2432 2433 2434 2435 2436 2437 2438 2439 2440 2441 2442 2443 2444 2445 2446 2447 2448 2449 2450 2451 2452 2453 2454 2455 2456 2457 2458 2459 2460 2461 2462 2463 2464 2465 2466 2467 2468 2469 2470 2471 2472 2473 2474 2475 2476 2477 2478 2479 2480 2481 2482 2483 2484 2485 2486 2487 2488 2489 2490 2491 2492 2493 2494 2495 2496 2497 2498 2499 2500 2501 2502 2503 2504 2505 2506 2507 2508 2509 2510 2511 2512 2513 2514 2515 2516 2517 2518 2519 2
```

Force در زمان نسبتاً کمی بدست آورد. دقت کنید! 25! کران بالا است و تعداد کلیدهای خاص بسیار کمتر است. از طرفی ویژگی‌های خاص این الگوریتم تحلیل فرکانسی را آسان‌تر می‌کند، به عنوان مثال اگر AB به LU تبدیل شود، BA به UL تبدیل می‌شود. ترکیب ویژگی‌های این الگوریتم با نمودار فرکانسی حروف دوتایی یک زبان شکستن آن را با تحلیل فرکانسی ممکن می‌سازد، اما به مراتب شکستن آن از شکستن الگوریتم‌های جانشینی تک‌الفبایی سخت‌تر است.

## ۴ سوال ۴

$$C' = K' \oplus P = \neg K \oplus P = \neg(K \oplus P) = \neg C$$

همانطور که رابطه بالا نشان می‌دهد پیام رمزنگاری شده با کلید نقیض، نقیض پیام را تولید می‌کند. همانطور که در درس به آن اشاره شد OTP دارای امنیت مطلق است و داشتن نقیض پیام رمز شده کمکی به ما برای رمزگشایی نمی‌کند.

## ۵ سوال ۵

### ۱.۵ خاصیت بهمنی اکید

خاصیت بهمنی به خاصیت دلخواه الگوریتم‌های رمزنگاری اشاره می‌کند که در آن یک تغییر کوچک در ورودی (تغییر یک یا چند بیت) باعث تغییر قابل توجه در خروجی می‌شود. خاصیت بهمنی اکید یا Strict avalanche criterion فرم فرمول‌سازی شده خاصیت بهمنی است. می‌گوییم یک الگوریتم دارای خاصیت بهمنی اکید است اگر تغییر یک بیت در ورودی باعث تغییر تمام بیت‌های خروجی با احتمال ۵۰ درصد شود.

### ۲.۵ خاصیت تمامیت

در رمزنگاری می‌گوییم یک تابع بولین کامل است اگر هر بیت خروجی به تمام بیت‌های ورودی وابسته باشد. در صورتی که به طور متوسط با تغییر هر بیت ورودی هر بیت خروجی با احتمال ۵۰ درصد تغییر پیدا کند استفاده از این تابع باعث وجود خاصیت بهمنی اکید در الگوریتم رمزنگاری می‌شود.

### ۳.۵ Random Cipher

مدل Random Cipher یک مدل تئوری است که در آن فرض می‌شود الگوریتم رمزگذاری یک تابع تصادفی است که پیام آشکار داده شده را با احتمال برابر به یک متن رمز نگاشت می‌کند. در صورتی که احتمال تمام متن رمزها برای یک متن آشکار یکسان باشد ارتباطی بین متن آشکار و متن رمز وجود نخواهد داشت و الگوریتم دارای امنیت مطلق خواهد بود.

## ۴.۵ بررسی خواص برای مدل داده شده

## ۱.۴.۵ خاصیت بهمنی اکید

دقت کنید پس از اجرای یک راند الگوریتم (همانطور که در تصویر نشان داده شده است) وابستگی بین بلوک‌های مختلف وجود ندارد و تنها بلوک  $y_3$  به بلوک  $x_1$  وابسته است. پس از ۴ بار تکرار این ساختار هر بلوک اولیه یک شیفت کامل می‌خورد و به بلوک سمت راست خود وابسته است. اگر خروجی‌های راند ۴ را با  $P_i$  نشان دهیم داریم

$$1. P_0 = g(X_1, X_0)$$

$$2. P_1 = g(X_2, X_1)$$

$$3. P_2 = g(X_3, X_2)$$

$$4. P_3 = g(X_0, X_3)$$

با انجام دادن این ساختار ۴ بار دیگر خواهیم داشت

$$1. P'_0 = g(g(X_1), P_1) = g'(X_1, X_2)$$

$$2. P'_1 = g(g(X_2), P_2) = g'(X_2, X_3)$$

$$3. P'_2 = g(g(X_3), P_3) = g'(X_3, X_0)$$

$$4. P'_3 = g(g(X_4), P_0) = g'(X_0, X_1)$$

و در نهایت پس از ۱۲ راند خواهیم داشت

$$1. P''_0 = g''(X_1, X_2, X_3)$$

$$2. P''_1 = g''(X_2, X_3, X_0)$$

$$3. P''_2 = g''(X_3, X_0, X_1)$$

$$4. P''_3 = g''(X_0, X_1, X_2)$$

تکرار ساختار داده شده به اندازه حداقل ۱۲ بار شرط لازم خاصیت تمامیت و خاصیت بهمنی اکید است. برای برقراری خاصیت بهمنی اکید و تمامیت اثبات می‌شود تابع  $F$  باید یک تابع bent باشد که در ادامه به تعریف آن خواهیم پرداخت:

**Bent Function**

تابع bent یک تابع بولین است که فاصله Hamming جدول صحت آن از تمامی تابع‌های خطی و affine بیشترین فاصله را دارد. تبدیل Walsh یک تابع بولین به صورت مقابل تعریف می‌شود.

$$f^a = \sum_{x \in \mathbb{Z}_2^n} (-1)^{f(x) + a \cdot x}$$

به طور دقیق‌تر تابع bent تابعی است که تبدیل Walsh آن یک تابع ثابت باشد.

## ۲.۴.۵ خاصیت تمامیت

همانطور که در بالا اشاره شد برای داشتن خاصیت تمامیت لازم است ساختار داده شده را حداقل ۱۲ بار تکرار کنیم و از تابع‌های خاصی برای  $F$  استفاده کنیم.

## ۳.۴.۵ Random Cipher

دقت کنید در مدل Random Cipher هر متن رمز برای یک متن آشکار مشخص احتمال برابری دارد و باید یک کلید مخصوص به خود داشته باشد. از آنجایی که CAST یک الگوریتم بلاکی است، طول کلید از قبل در آن فیکس می‌شود و سائز آن تغییر نمی‌کند، در صورتی که طول متن آشکار دست ما نیست. از آنجایی که فضای حالت متن آشکار بزرگتر از فضای حالت کلیدها است احتمال متن رمزها نمی‌تواند برای هر متن آشکار برابر باشد و این ساختار از مدل Random Cipher پیروی نمی‌کند.

## ۶ سوال ۶

بله. یکی از مشکلات اصلی رمزنگاری DES سائز کلید آن است. این کلید ۵۶ بیتی حمله آزمون جامع را ممکن می‌سازد. به همین دلیل یکی از ویژگی‌های اصلی کلید که سائز آن است تاثیرگذار است. عامل دیگری که در رمزگذاری DES حائز اهمیت است نوع کلید است. هر کلید ۵۶ بیتی در ۱۶ دور با استفاده از یک Key Scheduler به یک کلید ۴۸ بیتی تبدیل می‌شود. در رمزنگاری هر الگوریتم ممکن است تعدادی کلید ضعیف داشته باشد که باعث می‌شود الگوریتم به درستی عمل نکند. در DES از بین  $2^{64}$  چهار کلید ضعیف داریم. در صورتی که این کلیدها در بدترین حالت انتخاب شوند در تمام ۱۶ مرحله یک کلید تولید می‌شود.

## کلیدهای ضعیف الگوریتم DES

1. 0x0101010101010101
2. 0xFEFEFEFEFEFEFEFE
3. 0xE0E0E0E0F1F1F1F1
4. 0x1F1F1F1F0E0E0E0E

دقت کنید کلیدهای بالا پس از Permutation اولیه به کلیدهای ۵۶ بیتی مقابل تبدیل می‌شوند

1. 0x00000000 00000000
2. 0x00000000 FFFFFFFF
3. 0xFFFFFFFF 00000000
4. 0xFFFFFFFF FFFFFFFF

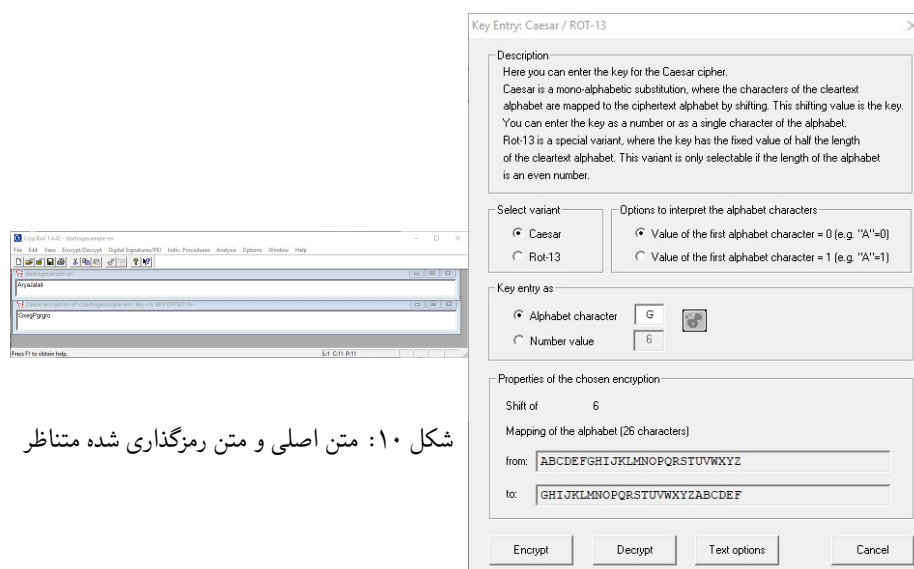
در صورتی که کلیدهای بالا به عنوان کلید ۶۴ بیتی انتخاب شوند، مهاجم اکنون تنها به دنبال یک کلید ۴۸ بیتی است و ۸ بیت از طول کلید حذف شده است. از طرفی دقت کنید الگوریتم DES به صورت عمومی در دسترس است و نحوه decipher کردن یک متن در الگوریتم DES به صورت وارون cipher کردن آن است، اگر ترتیب کلیدها را برعکس کنیم. حال در حالتی

که تمام کلیدها یکسان باشند عملیات decipher با cipher یکسان خواهد بود و مهاجم می‌تواند با یکبار رمزنگاری کردن پیام رمز شده به پیام اصلی دست پیدا کند، یعنی

$$E_{WK}(E_{WK}(P)) = P$$

## ۷ سوال ۷

### ۱.۷ رمزگذاری با Caesar

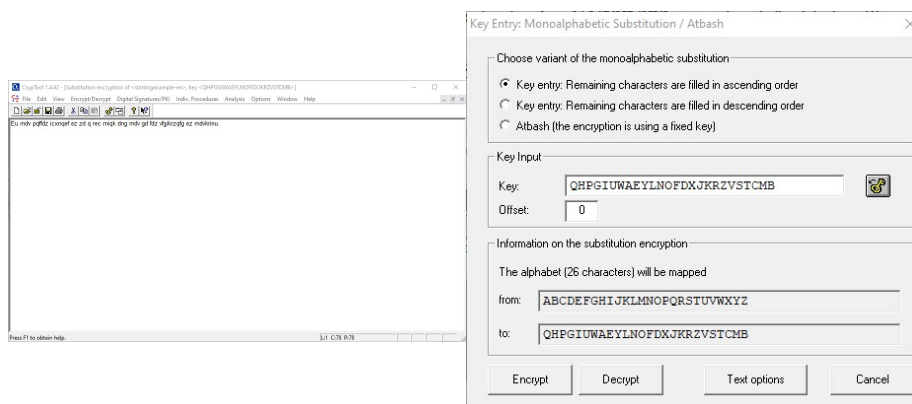


شکل ۱۰: متن اصلی و متن رمزگذاری شده متناظر

شکل ۹: فرایند رمزگذاری

متن رمز شده‌ی 'AryaJalali' با استفاده از روش سزار برابر است با 'GxegPgrgro'.

## ۲.۷ رمزگذاری با رمز جایگزینی

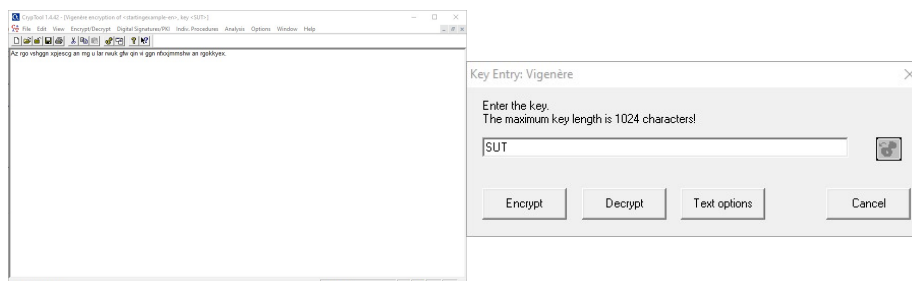


شکل ۱۲: متن رمزگذاری شده

شکل ۱۱: فرایند رمزگذاری

## ۳.۷ رمزگذاری با Vigenere

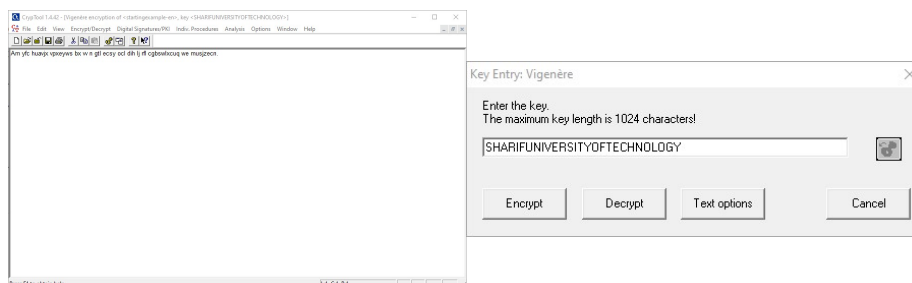
۱.۳.۷ کلید 'SUT'



شکل ۱۳: فرایند رمزگذاری

شکل ۱۴: متن رمزگذاری شده

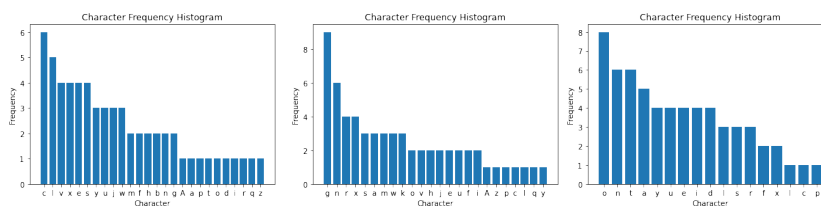
## ۲.۳.۷ کلید 'SHARIFUNIVERSITYOFTECHNOLOGY'



شکل ۱۵: فرایند رمزگذاری

شکل ۱۶: متن رمزگذاری شده

## ۳.۳.۷ تاثیر طول کلید



شکل ۱۹: کلید 'SHARIF'

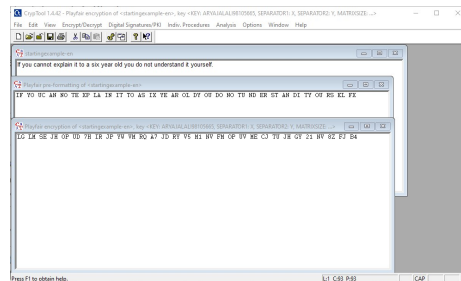
شکل ۱۸: کلید 'SUT'

شکل ۱۷: متن اصلی

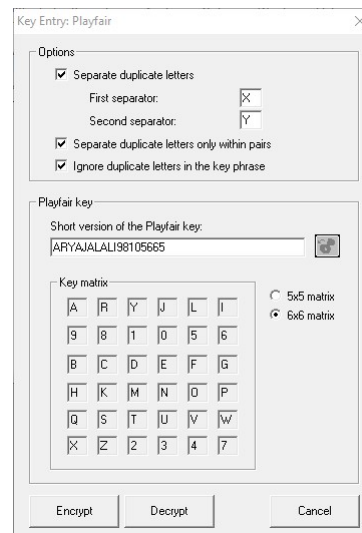
همانطور که از نمودارهای فرکانسی بالا مشخص است، استفاده از کلیدهای طولانی‌تر یکتایی کلمات را کاهش می‌دهد و احتمال اینکه حروف یکسان با یک حرف رمز شوند کاهش پیدا می‌کند و بررسی جایگشت حروف در نمودار فرکانسی به مراتب دشوارتر است به دلیل تعداد حالات بیشتر.



## ۴.۷ رمزگذاری با Playfair



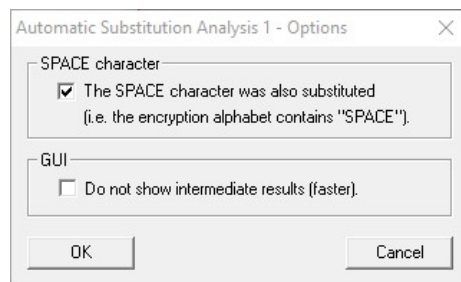
شکل ۲۱: متن رمزگذاری شده



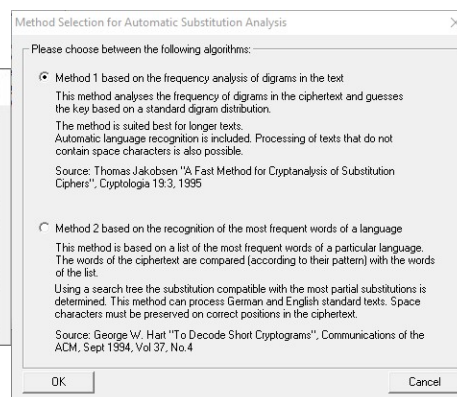
شکل ۲۰: فرایند رمزگذاری

## ۵.۷ رمزگشایی جایگشتی

## ۱.۵.۷ تنظیمات



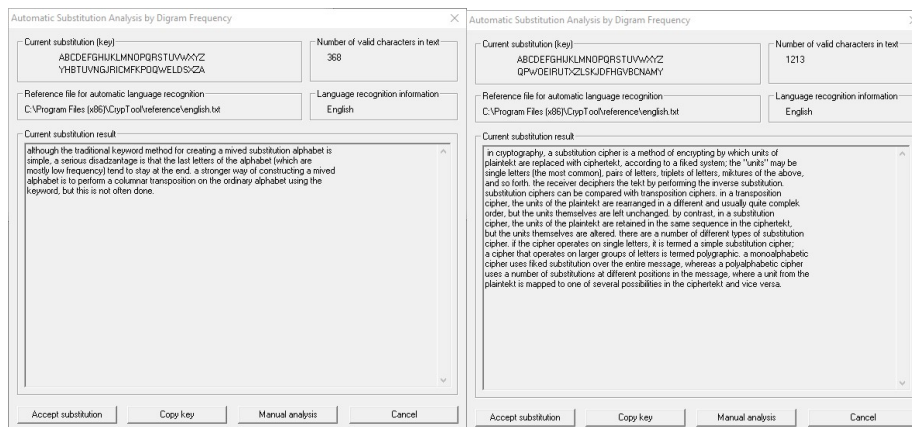
شکل ۲۳: تنظیمات رمزگشایی



شکل ۲۲: انتخاب روش رمزگشایی

دو روش رمزگشایی پیشنهاد شد که برای این سوال از روش اول که روش default است استفاده کردیم و در ادامه space را به عنوان حروف الفبا متن در نظر گرفتیم (در صورت عدم این فرض، خروجی معنی دار نمی شود)

## ۲.۵.۷ رمزگشایی

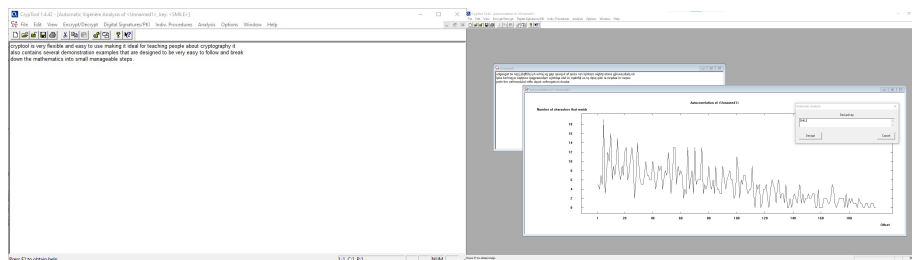


شکل ۲۴: متن اول

شکل ۲۵: متن دوم

در متن اول حرف X به اشتباه به A مپ شده است، در متن دوم این اتفاق برای حروف X و V رخ داده است. یکی از دلایلی که می‌تواند بانی این اتفاق باشد طول بیشتر متن اول است. در صورتی که طول متنی بزرگتر باشد، تحلیل فرکانسی رمز جایگشتی آن با احتمال بیشتری به فرکانس زبان انگلیسی نزدیک‌تر خواهد بود و رمزگشایی آن آسان‌تر خواهد بود. به همین دلیل متن ۱ به زبان طبیعی نزدیک‌تر است.

## ۶.۷ رمزگشایی vigenere



شکل ۲۶: کلید و نمودار Autocorrelation

شکل ۲۷: رمزگشایی شده متن رمز

علاوه بر روش Kasiski دیگری که برای بدست آوردن طول کلید استفاده می‌شود روش Autocorrelation است. در این روش متن رمز را مقداری شیفت می‌دهیم و تعداد کاراکترهایی که عین هم هستند را بدست می‌آوریم، به احتمال بالا زمانی که به اندازه طول کلید شیفت می‌دهیم این مقدار ماکسیمم خود را می‌گیرد که در نمودار بالا برابر با ۵ است که طول کلید ما یعنی 'SMILE' است. ایده پشت این روش این است که پس از شیفت

دادن به اندازه طول کلید، بقیه متن با یک الگو رمزگذاری شده‌اند و احتمال برابری زیاد است، در صورتی که کمتر از طول کلید شیفت دهیم حروف یکسان به احتمال بالا با حروف مختلف کلید رمزگذاری شده‌اند و تعداد کاراکترهای یکسان کمتر از حالت گفته شده است. منطقی هست که هرچه بیشتر شیفت دهیم به دلیل اشتراک کمتر بین دو متن تعداد کاراکترها به صورت متوسط کاهش پیدا می‌کند.

$$c_i - c_j = p_i - p_j + k_{i \bmod \mathcal{L}} - k_{j \bmod \mathcal{L}}$$

برای 0 شدن عبارت بالا که به معنای برابری کاراکترهای  $c_i$  و  $c_j$  در ciphertext است، نیاز داریم عبارت سمت راست 0 شود. اگر فرض کنیم که  $i \equiv j \pmod{\mathcal{L}}$  رابطه بالا به صورت مقابل در خواهد آمد

$$c_i - c_j = p_i - p_j$$

به دلیل نامنظم بودن توزیع آماری متن اصلی، انتظار داریم عبارت سمت راست با احتمال بالاتری نسبت به بقیه حالات 0 شود که نتیجه می‌دهد اگر عبارت را به اندازه طول کلید شیفت دهیم احتمال دیدن حروف برابر بیشتر است.

## ۸ سوال ۸

### ۱.۸ پیدا کردن طول کلید

```
import math

words = CFG.encrypted.split(" ")
repeated_words = {}
gcd = 0

current_length = 0
for word in words:
    if len(word) < 3:
        current_length += len(word)
        continue
    if word not in repeated_words:
        repeated_words[word] = [current_length]
    else:
        for length in repeated_words[word]:
            if gcd == 0:
                gcd = current_length - length
            else:
                gcd = math.gcd(gcd, current_length - length)
        repeated_words[word].append(current_length)
        current_length += len(word)

possible_key_lengths = [i for i in range(2, gcd+1) if gcd % i == 0]
```

با استفاده از یک دیکشنری کلمات را ذخیره می‌کنیم و در صورتی که طول آن‌ها بزرگتر از ۳ باشد و قبلاً در متن با آن‌ها برخورد کرده باشیم، ب.م.م فاصله‌های کلمه‌های یکسان را محاسبه می‌کنیم و این فرایند را تا پایان متن ادامه می‌دهیم و در نهایت مقسوم‌علیه‌های ب.م.م را به عنوان طول‌های ممکن کلید در لیست possible\_key\_lengths ذخیره می‌کنیم.

## ۲.۸ بدست آوردن کلید

```
import numpy as np

def get_distribution(text):
    freq = [text.count(char) for char in CFG.alphabet_frequencies.keys()]
    freq = [f / sum(freq) for f in freq]
    return freq

def get_current_char(distribution):
    word_freq = np.array(list(CFG.alphabet_frequencies.values())) / 100
    mse_list = [np.mean((np.roll(word_freq, shift) - distribution) ** 2) for shift \
in range(1, len(distribution) + 1)]
    best_shift = np.argmin(mse_list)
    return list(CFG.alphabet_frequencies.keys())[ (best_shift + 1) % len(distribution)]

clean_text = CFG.encrypted.replace(" ", "")

keys = []
for key_length in possible_key_lengths:
    cur_key = ""
    for i in range(key_length):
        distribution = get_distribution(clean_text[i::key_length])
        cur_key += get_current_char(distribution)
    keys.append(cur_key)
```

در این بخش برای هر طول کلید ممکن حروفی که توسط یک حرف کلید رمزگذاری شده‌اند را جدا می‌کنیم و توزیع آماری تکرار حروف را بدست می‌آوریم و با معیار MSE مقدار شیفی را پیدا می‌کنیم تا این معیار کمینه شود و با توجه به مقدار شیف، حرف متناظر را به عنوان حرف  $i$  ام کلید برمی‌گردانیم. در نهایت کلیدها را در آرایه keys ذخیره می‌کنیم.

\*کلیدهای بدست آمده 'دروغ' و 'مف' هستند.

## ۳.۸ رمزگشایی

```
def decode_words(text, key):
    spaces = [i for i, letter in enumerate(text) if letter == " "]
    text = text.replace(" ", "")
    letters = list(CFG.alphabet_frequencies.keys())
```

```

decoded = ""
for i, letter in enumerate(text):
    decoded += letters[(letters.index(letter) - \
        letters.index(key[i % len(key)])) % len(letters)]

for index in spaces:
    decoded = decoded[:index] + ' ' + decoded[index:]
return decoded

for key in keys:
    print(f"Key = {key}\n", decode_words(CFG.encrypted, key))

```

در کد بالا ابتدا پس از ذخیره کردن مکان white space ها، حروف کلید را به ترتیب از حروف متن رمز کم می‌کنیم تا به متن اصلی برسیم و در نهایت با استفاده از لیست spaces فاصله‌ها را در مکان خود قرار می‌دهیم.

#### ۱.۳.۸ متن با کلید 'دروغ'

لقمان حکیم پسر را گفت امروز طعام مخور و روزه دار و هرچه بر زبان راندی بنویس شبانگاه همه آنچه را که نوشتی بر من بخوان آنگاه روزها را بگشا و طعام ختیج شبانگاه پسر هر چه نوشته بود خواند دیروقت شد و طعام نتوانست خورد روز دوم نیز چنین شد و پسر هیچ طعام نخورد روز سوم باز هرچه گفته بود نوشت و تا نوشته را بر خواند افتاب روز چهارم طلوع کرد و او هیچ طعام نخورد روز چهارم هیچ نگفت شب پدر از او خواست که کاغذها بیاتجد و نوشته‌ها بخواند پسر گفت امروز هیچ نگفتم تا برخوانم لقمان گفت پس بیا و از این نان که بر سفره است بخور و بدان که روز قیامت آنان که کم گفته‌اند چنان حال خوشی دارند که اکنون تو داری

#### ۴.۸ متن با کلید 'مف'

خزویذ نلهد قصذغف مغض غوذرب عطسص وحرای ذریا خسا ی وگما اگ بتیذ اېمقع تمرعص سشغھکسط الز غهجز اب قز ضیسضع تذ دض تحرغه یذسپو گطسوسک ژی شسضی ر حفید وجث وفیمحفا بنا اذ عظ هنوکا اره ذنسضر خژایفض ثرن ادپل ذکیذتج حرار ذرب رند ضبر عضبم وه ی بنا اھع حفید ضدنگه ژنل تیل شغس وگما کثکا اره هنوک ی پس ضیسضط ژی شا ذنسضر یشکپا گطس جزغژل اشیط چار ن سط اھع حفید ضدنگه ژنل مایگص اھع ضمغض ثت بقا پر سط ذنسج قز ژپعکظ اژغجیق ط هنوکاوس فذنسضر بنا مغض غوذرب اھع ضمغضطیل ضغ تذفطیمد شگلضض مغض قص اژغ ی یل غبم ذغه قز فژ ژئاا ینک تحرای اقغه قز ایر جعپلض غھیز ژا قد سکیزغھخ عضیم غغن حرثب خساھخ چظ پقذطه پر هپژ

همانطور که واضح است، کلید واقعی 'دروغ' با طول ۴ است.