**به نام خدا**

**نام موضوع:**

# رمزنگاری

**ارائه دهنده :**

**شروین ایران عقیده**

## ۱- مقدمه

الگوریتم‌های رمزنگاری در دنیای سایبر امروزی که همیشه خطر دسترسی غیرمجاز به همه نوع داده وجود دارد امری حیاتی محسوب می‌شوند. حساس‌ترین و آسیب‌پذیرترین داده، داده‌های سیستم مالی و پرداختی است که می‌تواند اطلاعات شناسایی شخصی یا مشخصات کارت پرداخت مشتریان و سایر اطلاعات شخصی را در معرض خطر قرار دهد. الگوریتم‌های رمزنگاری برای محافظت از این اطلاعات و کاهش خطراتی که مشاغل در انجام معاملات پرداخت با آن روبرو هستند، بسیار مهم است.

## ۲- رمزگذاری (رمزنگاری) چیست؟

رمزگذاری (رمزنگاری) روشی برای ویرایش اطلاعات به گونه‌ای است که فقط اشخاص مجاز می‌توانند اطلاعات را درک کنند. از نظر فنی، این فرآیند تبدیل متن ساده به متن رمز شده است. به عبارت ساده‌تر رمزگذاری داده‌های قابل خواندن را می‌گیرد و آن را تغییر می‌دهد تا غیر قابل فهم و تصادفی به نظر برسد. رمزگذاری برای انجام این عمل به استفاده از کلید رمزنگاری نیاز دارد. این کلید مجموعه‌ای از مقادیر ریاضی است که هم فرستنده و هم گیرنده پیام رمزگذاری شده آن را می‌شناسند.

اگرچه داده‌های رمزگذاری شده تصادفی به نظر می‌رسند، اما رمزگذاری به روشی منطقی و قابل پیش بینی انجام می‌شود، به گونه‌ای که طرفی که داده‌های رمزگذاری شده را در اختیار دارد و با اختیار داشتن کلید مورد استفاده برای رمزگذاری داده‌ها می‌تواند داده‌ها را رمزگشایی کند و آن را به متن ساده تبدیل کند. اما یک رمزگذاری امن می‌بایست به‌اندازه کافی پیچیده باشد تا شخص ثالث نتواند آن را حدس بزند و یا با استفاده از ابزارهای مختلف متن رمز شده را تبدیل به متن ساده کند.

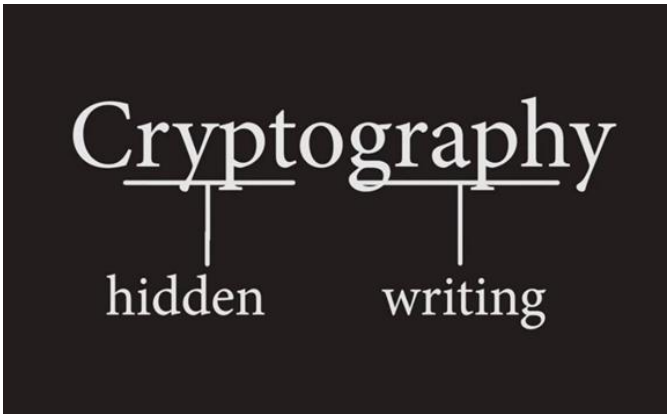
## ۳- تاریخچه رمزنگاری

رمزنگاری و استفاده از کدها و رمزها برای محافظت از اسرار در واقع هزاران سال پیش آغاز شد. این نوع از رمزنگاری را می‌توان رمزنگاری کلاسیک نامید، یعنی روش‌های رمزنگاری که در آن‌ها از قلم و کاغذ و یا شاید کمک‌های مکانیکی ساده استفاده می‌کنند. در اوایل قرن بیستم، اختراع ماشین‌های پیچیده مکانیکی و الکترومکانیکی، مانند دستگاه Enigma، ابزارهای پیچیده‌تر و کارآمدتری را برای رمزگذاری فراهم آورد که طرح‌های مختلف و پیچیده‌ای داشتند و هنوز هم برخی از آن‌ها بسیار پیچیده می‌باشند اما بزرگ‌ترین مشکل این نوع از رمزگذاری‌ها استفاده از کاغذ و قلم بود.

این نوع از رمزنگاری‌ها در جنگ جهانی اول مورد استفاده قرار می‌گرفت اما با تمام پیچیدگی‌ها معایبی را نیز به همراه داشت که سبب شکسته شدن آن می‌شد. تا اینکه الگوریتم‌های رمزنگاری جدید و امنی در این جنگ‌ها مورد استفاده قرار گرفت. تا دهه ۱۹۶۰ روش‌های این رمزنگاری امن در اختیار دولت‌ها بود و دولت‌ها قادر به رمزگشایی هر اطلاعاتی بودند. اما طولی نکشید تا با ارائه دو استاندارد رمزگذاری عمومی (DES) و اختراع رمزنگاری کلیدی عمومی الگوریتم‌های رمزنگاری عمومی شوند.

در این استانداردها هیچکس به جز افرادی که مجاز به دریافت پیام هستند قادر به شکستن اطلاعات کد شده نیستند. شاید از خودتان بپرسید که عمومی شدن الگوریتم‌های رمزنگاری چطور باعث شد تا دستیابی به اطلاعات برای اشخاص غیرمجاز غیرممکن شود، در صورتی که با دانستن الگوریتم دیگر هر شخصی می‌تواند به داده‌های اصلی دست یابد. اگر با یک الگوریتم کاملا شفاف و ساده روبرو بودیم عملا وجود این الگوریتم‌ها هیچ تاثیری نداشت اما مشخص نبودن بخش‌های حیاتی الگوریتم و همچنین پیچیدگی این الگوریتم‌ها آنقدر بالاست که عملا بدون داشتن اطلاعات مورد نیاز شکست آن غیر ممکن است.

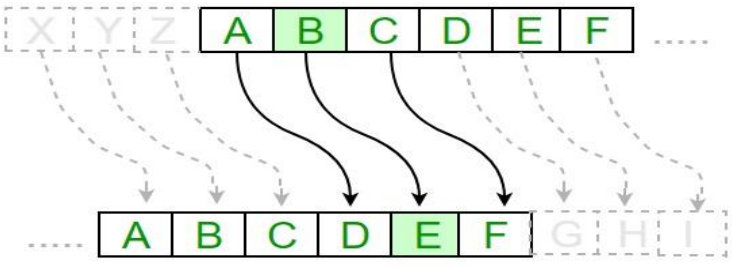
کلمه رمزنگاری از کلمات یونانی kryptos و graphein گرفته شده است که به معنای پنهان و نوشتن می‌باشند.




در رمزنگاری، **رمز سزار**[1] که با نام‌های کد سزار، شیفت سزار، یا رمز شیفت نیز شناخته می‌شود، یکی از ساده‌ترین و شناخته‌شده‌ترین تکنیک‌های رمزگذاری است. این رمز یک نوع رمز جانشینی است که در آن هر حرف در متن آشکار با حرف دیگری با فاصله ثابت در الفبا جایگزین می‌شود. برای مثال با مقدار انتقال ۳، D به جای A می‌نشیند، E به جای B، و الی آخر. نام این روش از ژولیوس سزار گرفته شده‌است که از آن برای ارتباطات محرمانه خود استفاده می‌کرد. الگوریتم رمز سزار دارای ویژگی‌های زیر است:

- تکنیک رمز سزار روش ساده و آسانی برای تکنیک رمزگذاری است.
- این رمز ساده است.
- هر حرف از متن ساده با حرفی تغییر می‌کند که دارای تعدادی موقعیت ثابت با حروف الفبا است.

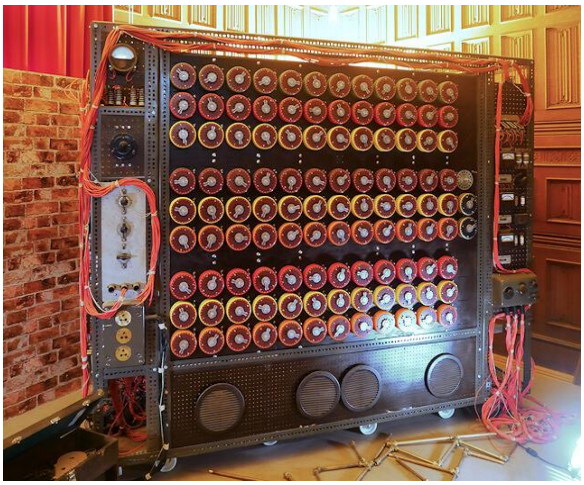
نمودار زیر نحوه اجرای الگوریتم رمز سزار را نشان می‌دهد:




[1] Caesar cipher

**ماشین انیگما (Enigma Machine):**

**ماشین بامب (Bombe Machine):**



## ۴- الگوریتم‌های رمزنگاری متقارن

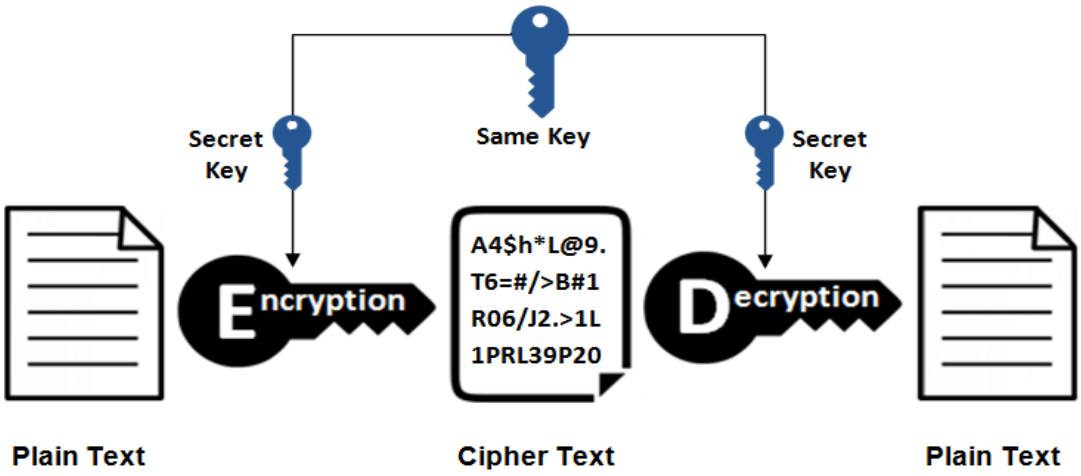
رمزگذاری متقارن یک روش رمزگذاری است که از یک کلید واحد برای رمزگذاری[۲] و رمزگشایی[۳] داده‌ها استفاده می‌کند. این قدیمی‌ترین و شناخته شده‌ترین تکنیک رمزگذاری است. کلید مخفی می‌تواند یک کلمه، یک شماره یا یک رشته از کارکترها یا اعداد باشد که توسط یک تولید کننده عدد تصادفی ایمن[۴] یا RNG تولید شده است. پیام، طبق قوانین الگوریتم رمزگذاری در کلید تغییر می‌کند. اشخاصی که از طریق رمزگذاری متقارن در حال برقراری ارتباط هستند باید کلید را مبادله کنند تا بتوانند اطلاعات را رمزگذاری و رمزگشایی کنند.

---

[۲] encryption

[۳] decryption

[۴] Random Number Generator




با استفاده از الگوریتم‌های رمزگذاری متقارن، داده‌ها به شکلی تبدیل می‌شوند که توسط کسی که کلید مخفی برای رمزگشایی را ندارد قابل درک نیست. هنگامی که گیرنده در نظر گرفته شده کلید مخفی پیام را داشته باشد، عکس عمل رمزگذاری جهت رمزگشایی انجام می‌دهد تا پیام به شکل اصلی و قابل فهم برگردد.

مؤلفه‌های اصلی در سیستم رمزنگاری متقارن وجود دارد که شامل موارد زیر می‌شوند:

- **متن ساده (Plaintext)**

واژه plaintext به پیام اصلی قابل فهم که نیاز به رمزگذاری دارد گفته می‌شود. متن ساده معمولا حاوی داده‌های حساس است که نباید توسط اشخاص غیرمجاز دیده شوند.

- **کلید (key)**

کلید به عنوان روش رمزگشایی شناسایی می‌شوند. بدون کلید متن رمزگذاری شده قابل رمزگشایی و خواندن نیست. کلید اطلاعات مربوط به همه سوئیچ‌ها و تعویض‌های ایجاد شده در متن ساده را در اختیار شما قرار می‌دهد. در رمزگذاری متقارن که نوعی از رمزگذاری است، کلید نیز می‌بایست بین طرفین به اشتراک گذاشته شود و روش

رمزگشایی جهانی نیست. امکان رمزگشایی به کلید بستگی دارد، زیرا در نهایت فرستنده و گیرنده کلید را به اشتراک می‌گذارند.

- **متن رمزنگاری شده (Cipher Text)**

متن رمزنگاری متنی است که رمزگذاری شده و آماده ارسال است. این متن ممکن است مانند یک حاوی مقادیر تصادفی از داده‌ها باشد و غیرقابل خواندن است.

## ۵- الگوریتم‌های رمزنگاری

یک الگوریتم رمزگذاری در حقیقت فرمول‌های ریاضی هستند که برای تبدیل داده (متن ساده) به متن رمزگذاری شده استفاده می‌شود. در برخی از رمزگذاری‌ها یک الگوریتم برای تغییر داده به روشی قابل پیش بینی از کلید استفاده می‌کند، به طوری که حتی اگر داده‌های رمزگذاری شده تصادفی به نظر برسند، اما می‌توان آن را با استفاده دوباره از کلید دوباره به متن ساده تبدیل کرد.

یک مثال ساده از الگوریتم رمزگذاری می‌توان در یک متن ساده همه حروف N را به عدد ۳ و یا تمام حروف Z را به ۱ تغییر دهد. این روال ممکن است چندین تغییر (یا جایگشت) را با متن ساده انجام دهد.

## ۶- الگوریتم رمزگشایی

در الگوریتم رمزگشایی، کلید مخفی (روش رمزگشایی) بر متن رمزنگاری اعمال شده و آن را به متن ساده تبدیل می‌کند. رمزگشایی معمولا رمزگذاری را به صورت معکوس انجام می‌دهد.

## ۷- برخی از نمونه‌های الگوریتم‌های رمزنگاری متقارن

رمزنگاری متقارن کاربردهای زیادی در تکنولوژی امروزه دارد. برخی از کارشناسان امنیت تنها الگوریتم‌های نامتقارن را پیشنهاد می‌کنند در صورتی که در بسیاری از موارد کاربردهای این الگوریتم با الگوریتم‌های نامتقارن متفاوت است. برخی از پرکاربردترین و محبوب‌ترین الگوریتم‌های رمزنگاری متقارن عبارتند از:

- AES
- DES
- IDEA
- Blowfish
- RC4
- RC5
- RC6

## ۸- مزایا و کاربرد الگوریتم‌های رمزنگاری متقارن

با این وجود که رمزگذاری متقارن روشی قدیمی برای رمزگذاری است، اما بسیار سریعتر و کارآمدتر از رمزگذاری نامتقارن است. رمزنگاری نامتقارن به دلیل مشکلات عملکردی و اندازه داده‌ها و استفاده از پردازنده‌های سنگین، شبکه‌ها را متضرر می‌کند. با توجه به عملکرد بهتر و سرعت سریعتر رمزگذاری متقارن (در مقایسه با نامتقارن)، رمزنگاری متقارن معمولاً در رمزگذاری فله (رمزگذاری مقادیر زیادی از داده‌ها) استفاده می‌شود.

به عنوان مثال برای رمزگذاری پایگاه داده، کلید مخفی فقط برای رمزگذاری یا رمزگشایی در دسترس پایگاه داده است. برخی از نمونه‌هایی که در آن از رمزنگاری متقارن استفاده می‌شود عبارتند از:

- برنامه‌های پرداخت، مانند معاملات کارت که در آن باید از اطلاعات شخصی و مالی محافظت شود تا از سرقت هویت یا اتهامات کلاهبرداری جلوگیری شود.

- استفاده برای تأیید اعتبار برای اینکه ثابت شود فرستنده پیام چه کسی است.
- RNG یا تولید کننده شماره به صورت رندم یا هشینگ.

**۹- معایب الگوریتم‌های رمزنگاری متقارن**

متأسفانه رمزگذاری متقارن با مشکلات خاص خود همراه است. یکی از مهم‌ترین نقطه ضعف‌های آن جنبه‌های مدیریت کلید در آن است که مشکلات زیر را در پی دارد:

- **فرسودگی کلید**

یکی از مشکلات بزرگ در رمزگذاری متقارن این است که در آن با هر بار استفاده از کلید، اطلاعاتی را فاش می‌کند که می‌تواند توسط مهاجمی برای بازسازی کلید استفاده شود. روش دفاعی در برابر این مشکل نیز استفاده از یک سلسله مراتب کلیدی برای اطمینان از عدم استفاده بیش از حد از کلیدهای رمزگذاری اصلی برای حجم بزرگی از رمزگذاری داده‌ها است.

- **تخصیص داده**

بر خلاف رمزگذاری‌های نامتقارن یا کلید عمومی، کلیدهای متقارن دارای فوق داده‌های جاسازی شده برای ثبت اطلاعاتی مانند تاریخ انقضا یا لیست کنترل دسترسی در استفاده از کلید ممکن نیستند.

- **مدیریت کلید در مقیاس بزرگ**

مدیریت چند کلید در یک طرح کوچک تا متوسط که زیر چندصد نقش دارند می‌توان از طریق دستی و از طریق فعالیت‌های انسانی انجام شود. اما در مقیاس‌های بزرگ، ردیابی انقضا و تنظیم چرخش کلیدها غیر عملی می‌شود. به عنوان مثال در کارت‌های پرداخت بانکی را در نظر بگیرید که میلیون‌ها کارت چاپ شده وجود دارد که به چندین کلید در هر کارت، و به یک سیستم اختصاصی و سیستم مدیریت کلید جامع نیاز دارند.

## ۱۰- الگوریتم هشینگ (Hashing)

به زبان ساده، هش کردن یا هشینگ (Hashing) به معنای دریافت یک رشته با یک طول دلخواه و تبدیل آن به یک خروجی با طول ثابت است. هشینگ یکی از مهم‌ترین فرآیندها در رمزنگاری بلاک چین است. هشینگ فرآیند ایجاد یک خروجی با اندازه ثابت از یک ورودی با اندازه متغیر است. به عبارتی داده وارد شده می‌تواند هر اندازه‌ای داشته باشد، اما بدون توجه به آن، داده به دست آمده همیشه اندازه ثابتی دارد. این کار به وسیله استفاده از یک سری فرمول‌های ریاضی به نام توابع هش انجام می‌شود. این تابع‌ها در قالب الگوریتم‌های Hashing پیاده می‌شوند. اگرچه در همه توابع هش، از رمزنگاری استفاده نمی‌شود، اما اصطلاحا توابع هش رمزنگاری، در مرکز رمز ارزها قرار داشته و جزوی جدانشدنی از آن‌ها به شمار می‌روند. به لطف این الگوریتم‌ها، بلاک چین‌ها و دیگر سیستم‌های توزیع شده به سطوح بالایی از یکپارچگی و امنیت داده دست پیدا می‌کنند. توابع هشینگ معمولی و رمزنگاری، قطعی هستند. قطعی بودن به این معنی است که تا زمان عدم تغییر ورودی، الگوریتم همیشه خروجی یکسانی تولید خواهد کرد. خروجی به دست آمده از تابعHashing ، یک هش (Hash) نام دارد. به طور معمول الگوریتم‌های هشینگ رمز ارزها به صورت توابع یک طرفه طراحی می‌شوند، به این معنی که امکان برگرداندن (رسیدن از خروجی به ورودی) بسیار دشوار بوده و در اغلب مواقع به سال‌ها پردازش نیاز دارد. این موضوع سبب می‌شود مشخص کردن ورودی بر اساس هش در دسترس، تقریبا غیر ممکن باشد. به راحتی می‌توان قطعه‌ای دیتا را وارد تابع و یک خروجی تولید کرد، اما بالعکس آن بسیار دشوار و در بیشتر مواقع کاملا غیر ممکن است. هر چه رسیدن به داده اولیه بر اساس هش دشوارتر باشد، الگوریتم هشینگ مورد استفاده ایمن‌تر محسوب می‌شود.