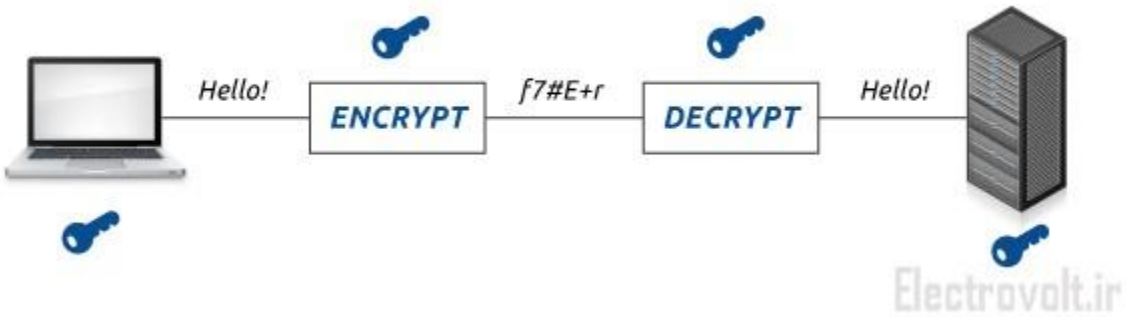
آشنایی با علم رمزنگاری (Cryptography) و پنهان نگاری (Steganography)

مقدمه

در گذشته از رمزنگاری برای ایجاد امنیت در ارتباطات نظامی و دولتی استفاده می شده است به طوری که اولین استفاده عملی از رمزنگاری به شیوه مدرن در جنگ جهانی دوم بود. اما امروزه با پیشرفت سریع فناوری اطلاعات و انجام الکترونیکی بسیاری از داد و ستد ها ، علوم رمزنگاری و پنهان نگاری نیز فراگیر شده است. به طوری که افراد زیادی با آن ها درگیر می شوند و باید از نحوه عملکرد این علوم و کاربردهای آن ها مطلع باشند. در این مقاله که توسط وبسایت تخصصی الکترو ولت تهیه و منتشر شده است ، با اصلی ترین مفاهیم این حوزه آشنا خواهید شد.

رمزنگاری چیست ؟

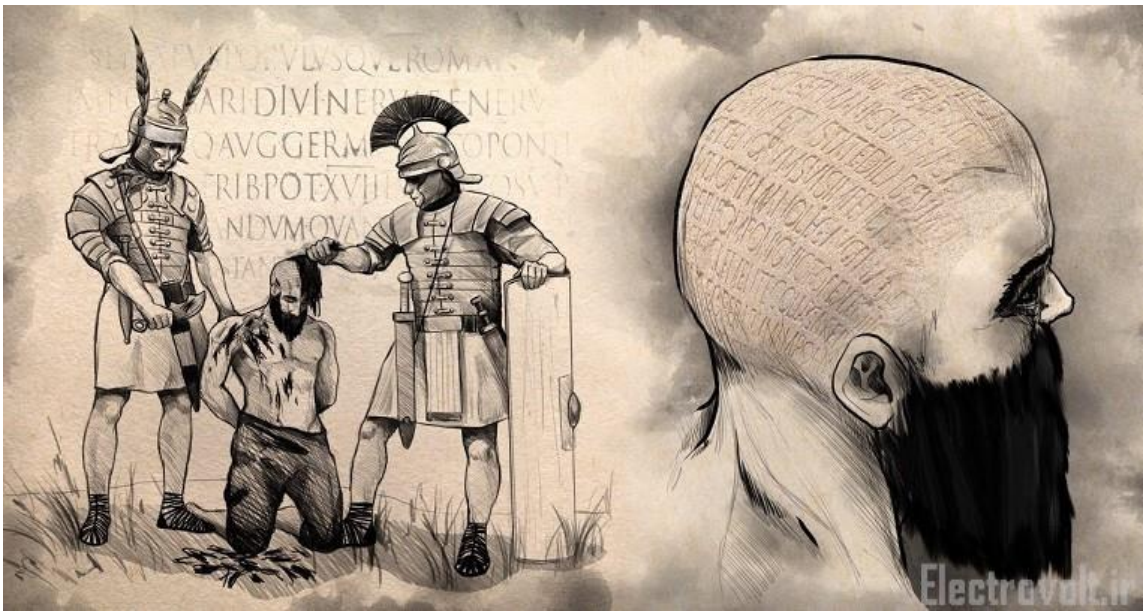
عبارت Cryptography برگرفته از لغات یونانی kryptos به معنای «محرمانه» و graphien به معنای «نوشتن» است. بنابراین کل کلمه به معنای "محرمانه نوشتن" می باشد. در رمزنگاری هدف این است که با استفاده از اصول و روابط ریاضی اطلاعات را به گونه‌ای ایمن رمز کنیم که با خیال آسوده بتوان از جایی به جای دیگر انتقال داد. رمزنگاری پیشینهٔ طولانی و درخشان دارد که به هزاران سال قبل برمی گردد. از این شیوه رمزنگاری در گذشته دور و به خصوص در یونان و فرانسه در مکاتبات و نامه های دولتی استفاده می شد. یکی از ساده ترین الگوریتم های رمزنگاری جایگزینی کاراکترهای یک کلمه با کاراکترهای دیگر می باشد که منجر به تولید یک کلمه جدید و نامفهوم می گردد. به شکل زیر توجه کنید.



پنهان نگاری چیست ؟

واژهSteganography ، برگرفته از دو کلمه یونانی stego به معنای پوشیده و graphien به معنای نوشتن است. بنابراین کل کلمه به معنای "پوشیده نویسی" می‌باشد. هدف steganography این است که پیغامی را در یک پیغام دیگر به روشی ذخیره کند که دشمن پی به وجود پیغام اولی در پیغام دوم نبرد. در ابتدا یونانیان باستان از این روش استفاده می‌کردند. به این‌صورت که پیغام در سنگ لوح بطور مستقیم حکاکی می‌شد؛ سپس نوعی واکس از بالای پیغام ریخته می‌شد که باعث پوشیده شدن پیام می شد.

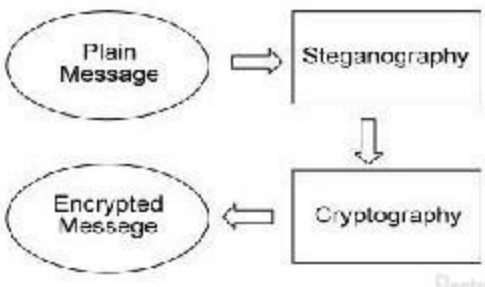
همچنین آمده است که در آن زمان مردی یونانی به نام هیستایاکاس می خواست پیغامی را به صورت محرمانه برای شخص دیگری بفرستد. وی برای فرستادن پیغام مورد استفاده از این روش استفاده کرد: او برده ای را برای این کار انتخاب کرد و موهای سر برده را تراشید و پیغام محرمانه را بر روی پوست سر برده خالکوبی کرد و سپس مدتی صبر کرد تا موهای فرد رشد کرده و به حالت اول برگشت و بعد او را به سمت مقصد (گیرنده) روانه کرد در مقصد، گیرنده ی پیغام دوباره موهای برده را تراشید و پیغام را بر روی پوست سر او مشاهده کرد.



تفاوت رمز نگاری و پنهان نگاری

در رمزنگاری ارسال به هیچ عنوان مخفی نیست و علاوه بر ارسال کننده و دریافت کننده دیگران نیز می‌توانند به محتوا دسترسی داشته باشند، فقط پیام برای همه به جز ارسال کننده و دریافت کننده غیر قابل فهم است. اما در پنهان نگاری نه تنها پیام مخفی است بلکه باید هر گونه نشانه‌ای از وجود پیام نیز مخفی باشد تا هیچ کس جز ارسال و دریافت کننده نتواند از وجود آن با خبر شود.

نکته : برای داشتن امنیت بیشتر میتوان این دو را با هم ترکیب کرد به طوری که ابتدا یک پیام رمزگذاری شود و سپس نهان نگاری و ارسال شود.



انواع نهان نگاری

استگانوگرافی علاوه بر حمل اطلاعات مخفی کاربردهای دیگری نیز دارد. یکی از کاربردهای عمومی آن می‌تواند این باشد که برای مثال صاحب حقوقی یک عکس، یک سری پیام درون تصویر به طور مخفی جاسازی کند. هر گاه چنین تصویری دزدیده شود و در یک وب سایت قرار داده شود، مالک قانونی آن می‌تواند این پیام محرمانه و سری را برای اثبات مالکیت به دادگاه عرضه کند. به این نوع استگانوگرافی اصطلاحاً نشانه گذاری یا watermarking گفته می‌شود. اما گاهی واترمارکینگ برای جلوگیری از کپی کردن آشکار بوده و نهان نگاری نمی شود که به آن نوع Visible گفته می شود (مانند شکل زیر)

فرمول کلی برای تابع Steganography این چنین است:

فایل مورد نظر که قرار است اطلاعات در آن نگهداری شود + اطلاعاتی که باید مخفی شوند + الگوریتم نهان نگاری مورد نظر = شی مورد نظر که اطلاعات در آن مخفی شده اند.

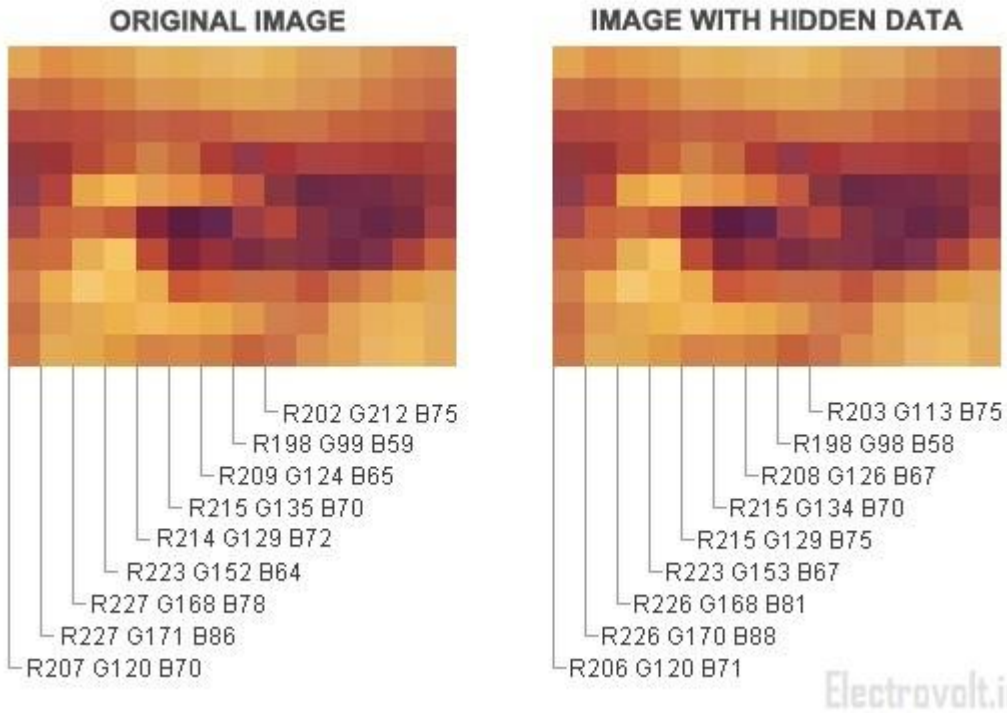
فایلی که برای مخفی کردن اطلاعات به کار می رود، می تواند یک تصویر، فایل صوتی و یا یک فایل ویدئویی باشد. همچنین دو روش معروف برای الگوریتم نهان نگاری وجود دارد که عبارتند از Injection و LSB

LSB: وقتی فایلی ساخته می شود، معمولاً بعضی از بایت های آن یا قابل استفاده نیستند و یا کم اهمیت هستند. این بایت ها می توانند تغییرداده شوند، بدون اینکه لطمه قابل توجهی به فایل وارد شود. این خاصیت کمک می کند تا بتوان اطلاعاتی را در این بایت ها قرار داد، بدون اینکه کسی متوجه این موضوع گردد. روش LSB بر روی فایل های تصویری که دارای رزولوشن وتعداد رنگ های بالایی است و بر روی فایل های صوتی که دارای تعداد زیادی صدای مختلف است، به خوبی کارمی کند. ضمناً این روش حجم فایل را افزایش نمی دهد.

Injection: روشی ساده است که برمبنای آن، اطلاعاتی که قراراست مخفی شوند را در یک فایل تزریق می کنند. مهمترین مسأله در این روش، افزایش حجم فایل است.

نهان نگاری در تصاویر

وقتی از یک تصویر برای مخفی نمودن یک متن (نوشته) استفاده می شود، معمولاً ازروش LSB استفاده می شود. ضمناً اگر در درون یک تصویر اطلاعاتی درج شده باشد وسپس این تصویربه فرمت دیگری تبدیل شود، به احتمال بسیار زیاد، بخش اعظمی ازاطلاعات مخفی شده از بین می رود وبخش باقی مانده نیزشاید با سختی فراوان قابل بازیابی باشد. در پنهان‌نگاری به جای تصویر می‌توان از فایل‌های صوتی و یا تصویری و حتی متنی برای مخفی سازی اطلاعات استفاده کرد. در فایل‌های متنی معمولا از tab ها و space های آخر سطرها که در اکثر ویرایشگرها توسط انسان قابل تشخیص نیستند، استفاده می‌شود. اطلاعات مخفی شده نیز لزوما متن نیستند بلکه می‌توانند هر نوع فایلی باشند. مثلا می‌توان یک تصویر را نیز در داخل تصویر دیگر جاسازی کرد. همچنین روش‌های پنهان نگاری، محدود به روش‌های مطرح شده‌ی موجود نیستند بلکه هر شخص می‌تواند از روش دلخواه خود برای پنهان نگاری استفاده کند.



در روش LSB دیتای ارسالی ابتدا به صورت 0 و 1 در می آید و سپس در بیت کم ارزش تصاویر جاسازی می گردد. همانطور که مشاهده می کنید بیت آخر در برخی از پیکسل های تصویر تغییر کرده و در برخی دیگر تغییر نکرده است. بنابراین در شکل کلی تصویر تغییر محسوسی مشاهده نمی شود و دیتا نهان نگاری می گردد.

نهان نگاری در صوت

برای این منظور نیز از روشی مشابه روش LSB استفاده می کنند. البته مشکل استفاده از بیت های کم ارزش در یک فایل صوتی، این است که تغییرات در این بیت ها نیز برای گوش انسان قابل تشخیص است.

در حقیقت Spread Spectrum روش دیگری برای مخفی نمودن اطلاعات در یک فایل صوتی است. دراین روش، یک نویز به طور تصادفی در سراسر فایل پخش می شود و اطلاعات در کنار این نویزها قرارداده می شوند Echo data hiding .نیز روش دیگری برای مخفی نمودن اطلاعات در یک فایل صوتی است. این روش از اکو (پژواک) در فایل استفاده می کند تا بتواند اطلاعات را مخفی نماید. دراین وضعیت با اضافه کردن صداهای اضافی به بخش های اکو، می توان اطلاعات را در این قسمت ها مخفی نمود.

نهان نگاری در ویدئو

برای این کار ، معمولاً از روش DCT استفاده می شود. این تکنیک شبیه تکنیک LSB است. یک فایل ویدئویی از تعدادی تصاویر پشت سرهم تشکیل شده است که این تصاویر به نام فریم شناخته می شوند. بنابراین کافی است که اطلاعات خود را درهر فریم یک فایل ویدئویی ، به روش LSB مخفی نماییم.

رمزگشایی و نهان کاوی

به طور کلی باز کردن رمز و فهمیدن محتوای پیام را رمزگشایی و نیز فهمیدن پیام مخفی شده را نهان کاوی می گویند که به ترتیب عکس رمزگذاری و پنهان نگاری است.

رمزگشایی یا Cryptanalysis به اصولی گفته می‌شود که بر اساس روابط ریاضی سعی در شکستن رمز بدون در اختیار داشتن کلید رمز دارد. اصولا دانش رمزگشایی به منظور اطمینان از غیر قابل شکست بودن رمزها پدید آمده است ولی گاهی برای شکستن رمزها و نفوذ به سیستم‌های امنیتی نیز مورد استفاده قرار می‌گیرد.

همچنین Steganalysis یا نهان کاوی نیز برای فهمیدن و کشف کردن یک پیام مخفی شده مورد استفاده قرار می گیرد.

انواع سیستم های رمز نگاری

سیستم های رمز نگاری به صورت زیر تقسیم بندی می گردند.

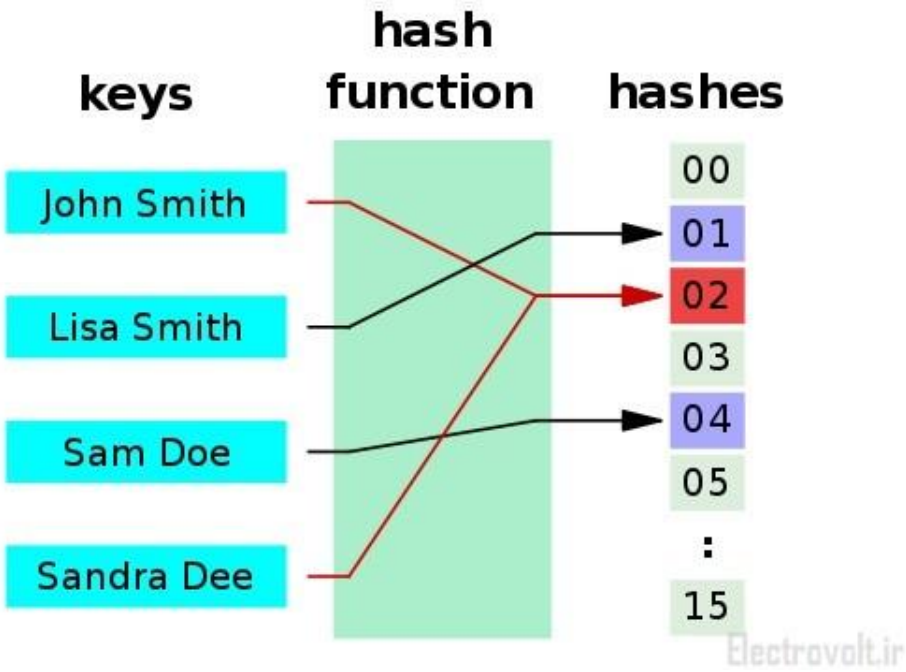
رمزنگاری برگشت ناپذیر که تحت عنوان توابع درهم سازی یا Hash Function مطرح می شود. توابعی یک طرفه و برگشت ناپذیر هستند که روی متونی اعمال می شود که نیاز به دانستن اصل آن متن نباشد. برای مثال در ذخیره سازی رمزعبور (Password) که شما در سایت های مختلف از آن ها استفاده می کنید ، چنین تابعی استفاده می شود. بنابراین شما با خیال راحت میتوانید پسورد خود را وارد کنید و مطمئن باشید که هیچ کسی جز خودتان نمیتواند از پسورد شما مطلع شود ، یا آن را بازیابی کند و خودش را جای شما قرار دهد.

اما رمزنگاری برگشت پذیر برای زمانی مورد استفاده قرار می گیرد که شخص می خواهد پیامی محرمانه را برای شخصی دیگر ارسال کند ، به طوری که فقط فرستنده و گیرنده از متن اصلی مطلع شوند و هیچ کس دیگری نتواند از محتوای پیام آگاهی یابد.

درهم سازی یا هش چیست ؟

توابع درهم سازی یاHash ، از جمله توابع و الگوریتم های پرکاربرد در امنیت سیستم های کامپیوتری و بررسی محتوای دیجیتال هستند. کارکرد این توابع، مشابه با اثر انگشت در انسان هاست و هدف از اعمال توابع Hash به یک بخش از داده، رسیدن به کدی تقریباً یکتا برای آن محتوا است که در محتواهای دیگر، تکرار نمی شود. یعنی احتمال تکرار آن، در کمترین مقدار ممکن است. تعداد قابل توجهی از این نوع الگوریتم ها طراحی شده اند و بسته به میزان پیچیدگی الگوریتم، احتمال تکراری بودن کد تولید شده، به مرور افزایش یافته است. از طرفی، این الگوریتم ها، به نوعی کارکرد فشرده سازی (البته با اتلاف)

را دارند و در بسیاری از سیستم های نرم افزاری و وب سایت ها، از این توابع برای تشخیص محتوای تکراری (یا به بیان دقیق تر محتوای احتمالاً تکراری) استفاده می شود که از میان آن ها، می توان به سایت گیت (Git) اشاره کرد.

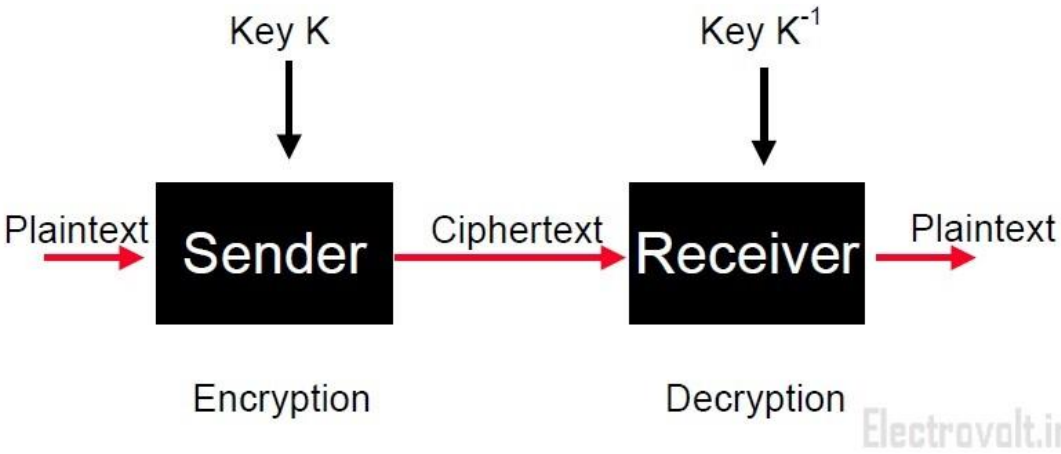


توابع هش، از نظر ریاضی توابع یک به یک نیستند و به دلیل کوچک تر بودن فضای خروجی (برد تابع) در مقایسه با فضای ورودی (دامنه تابع) و طبق اصل لانه کبوتری، قطعاً ورودی هایی قابل یافتن هستند که عیناً یک کد Hash را داشته باشند. اما یافتن این موارد، که به تصادم یا Collision معروف هستند، با پیچیده شدن الگوریتم ها سخت تر می شود. برخی افراد ، hashingرا به عنوان یک مدل رمزنگاری تلقی می نمایند Hashing .یک مدل رمزنگاری نمی باشد چراکه Hash نمی تواند رمزگشایی گردد و مقدار ورودی آن با استناد و آنالیز از روی مقدار خروجی قابل تشخیص نیست.

یکی از الگوریتم های معروف و پرکاربرد در میان توابع هش، الگوریتم SHA-1 (یا نسخه ۱ الگوریتم Security Hash Algorithm) است که در سال ۱۹۹۵ معرفی شده است. اما در سال ۲۰۰۵، به صورت نظری ثابت شد که احتمالاً یافتن تصادم در این الگوریتم، با کامپیوترهای فعلی، یک موضوع عادی خواهد بود. این موضوع، باعث شد که برخی از شرکت های فعال در حوزه فناوری اطلاعات، استفاده از این الگوریتم را متوقف نمایند. اما این موضوع و خطر، چندان جدی گرفته نمی شد، تا این که در ۲۳ فوریه سال ۲۰۱۷ میلادی (۵ اسفند ۱۳۹۵)، تیمی مشترک بین گوگل و گروه CWI Amesterdam توانستند دو فایل PDFمتفاوت بسازند که دارای هش SHA1 کاملاً برابر با هم هستند. بعد از آن الگوریتم های بهبود یافته دیگری نظیرSHA-2 ، SHA-3 و SHA-256 ارائه شدند.

از الگوریتم های hashing پرکاربرد دیگر میتوان به دسته Message-Digest Algorithm نیز اشاره کرد. هم اکنون از نسخه های 4 و 5 این الگوریتم که به ترتیب به نام های MD4 و MD5 هستند استفاده می شود. این سری از هش ها در دانشگاه MITو توسط پروفسور Ronald L. Rivest طراحی شده است.

رمزنگاری کلید متقارن



در رمزنگاری متقارن پیامی وجود دارد که فرستنده می خواهد آن را برای گیرنده ارسال نماید به طوری که هیچ کسی نتواند به محتویات آن پی ببرد. برای انجام این کار از یک الگوریتم و یک کلید رمزنگاری مشترک بین دو شخص فرستنده و گیرنده استفاده می شود. (الگوریتم و کلید باید مخفی باشد و فقط شخص فرستنده و گیرنده از آن مطلع باشند)

Encryption: به عملیات رمزنگاری اطلاعات گفته می شود.

Decryption: به عملیات رمزگشایی اطلاعات گفته می شود.

key: به معنای کلید است که در رمزگذاری و نیز رمزگشایی اطلاعات از آن استفاده می شود.

Encryption Algorithm: به الگوریتم رمزگذاری اطلاعات گفته می شود.

Decryption Algorithms: به الگوریتم رمزگشایی اطلاعات گفته می شود.

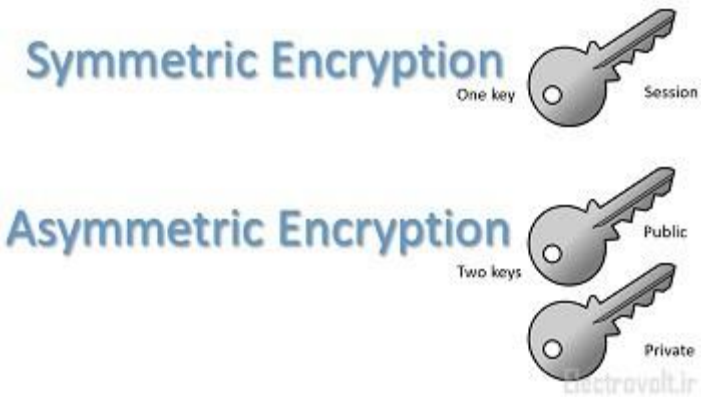
Plain text: به متن اصلی یا پیامی که قرار است رمزگذاری شود گفته می شود.

Cipher text: به متن رمز شده (تغییر یافته پس از رمز گذاری) گفته می شود.

نکته : امنیت یک سیستم رمزنگاری ، به دو چیز وابسته است : 1- کلید رمزنگاری 2- الگوریتم رمزنگاری و در صورت فهمیدن آن دو میتوان سیستم رمزنگاری را شکسته و به پیام اصلی دسترسی داشت.

رمزنگاری کلید نامتقارن

در سال 1976 دسته ای جدید از سیستم های رمزنگاری مبتنی بر دو کلید توسط دو دانشمند به نام‌های دیفی هلمن و مارتین هلمن ارائه شدند. به این سیستم ها نامتقارن یا Assymetric گفته می شود که در برابر سیستم های قدیمی تر متقارن یا Symetricقرار می گیرند. رمزنگاری نامتقارن برای حل مشکل انتقال کلید یکسان از گیرنده به فرستنده ابداع شد.

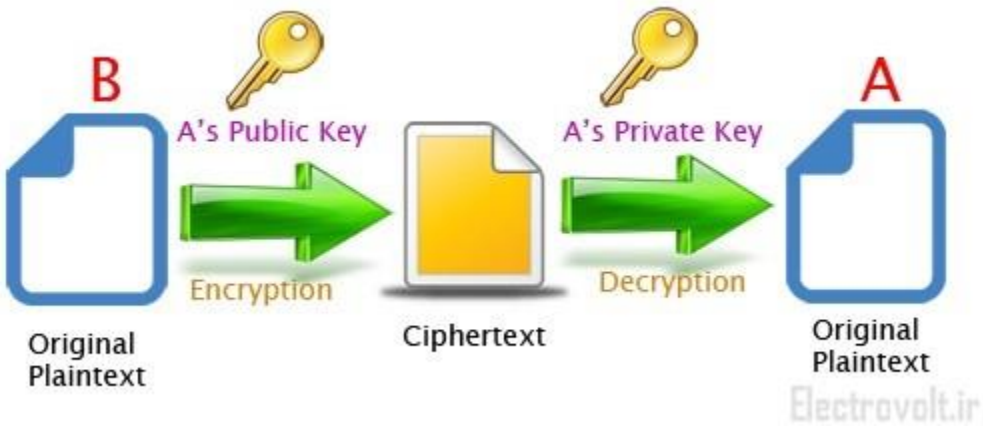


در سیستم های متقارن برای رمزنگاری و رمزگشایی از یک کلید استفاده می شود. اما در سیستم های نامتقارن برای کلید رمزنگاری ویژگی ارائه شد که افراد مختلف بتوانند با یکدیگر به وسیله کلید عمومی پیام های رمز شده مبادله کنند و در عین حال هیچ کسی از کلید خصوصی یکدیگر اطلاعی نداشته باشد.

Public Key: به کلید رمزنگاری عمومی گفته می شود که در دسترس افراد دیگر (ارسال کنندگان پیام) قرار می گیرد.

Private Key: به کلید رمزنگاری خصوصی گفته می شود که تنها در دسترس فرد اصلی (گیرنده پیام) قرار می گیرد.

برای درک بیشتر موضوع به شکل زیر نگاه کنید:



در این شکل فرد B می خواهد برای فرد A پیام محرمانه ای ارسال نماید. چون سیستم نامتقارن است ابتدا فرد B کلید عمومی فرد A را از او دریافت می کند. سپس پیامی که میخواهد ارسال کند را با کلید عمومی فرد A رمزنگاری کرده و برای او ارسال می کند. متن رمز شده برای هیچ کسی قابل رمزگشایی نیست و تنها بوسیله کلید خصوصی فرد A قابل رمزگشایی می باشد. این سیستم قابلیتی ایجاد میکند که افراد مختلف با یکدیگر پیام های محرمانه ارسال کنند و به افراد مجموعه اجازه می دهد که هر کسی کلید مخصوص به خود را داشته باشد که هیچ کسی از آن اطلاعی ندارد.

انواع الگوریتم های رمزنگاری

الگوریتم رمزنگاری، به هر الگوریتم یا تابع ریاضی گفته می‌شود که به علت دارا بودن خواص مورد نیاز در رمزنگاری، در پروتکل‌های رمزنگاری مورد استفاده قرار گیرد. اصطلاح الگوریتم رمزنگاری یک مفهوم جامع است و لازم نیست هر الگوریتم از این دسته، به طور مستقیم برای رمزگذاری اطلاعات مورد استفاده قرار گیرد، بلکه صرفاً وجود کاربرد مربوط به رمزنگاری مد نظر است. در گذشته سازمان‌ها و شرکت‌هایی که نیاز به رمزگذاری یا سرویس‌های دیگر رمزنگاری داشتند، الگوریتم رمزنگاری منحصربه‌فردی را طراحی می‌نمودند. به مرور زمان مشخص گردید که گاهی ضعف‌های امنیتی بزرگی در این الگوریتم‌ها وجود دارد که موجب سهولت شکسته شدن رمز می‌شود. به همین دلیل امروزه رمزنگاری مبتنی بر پنهان نگاه داشتن الگوریتم رمزنگاری منسوخ شده‌است و در روش‌های جدید رمزنگاری، فرض بر این است که اطلاعات کامل الگوریتم رمزنگاری منتشر شده‌است و آنچه پنهان است فقط کلید رمز است. به الگوریتم های رمزی که تا قبل از سال های 1950 میلادی (یعنی قبل از ارائه تئوری شنون) ارائه شده بودند ، رمزهای کلاسیک و به الگوریتم های رمزی که بعد از آن منتشر شده اند رمز های مدرن گفته می شود.

الگوریتم های رمزنگاری کلاسیک

در دوره کلاسیک ، ابتدایی ترین تلاش‌ها برای یافتن اصول رمزنگاری توسط آگوست کرشهف صورت پذیرفت. کرشهف در سال 1884 مقالاتی برای دانش رمزنگاری نظامی نوشت که حاوی شش اصل و قانون برای رمزنگاری داده ها است. این شش اصل عبارتند از:

- سیستم رمزنگاری باید در عمل غیر قابل شکستن باشد.
- سیستم رمزنگاری نباید پنهان باشد، بلکه این کلید رمز است که پنهان است.
- کلید رمز باید قابل به خاطر سپردن و تعویض باشد.
- متون رمزنگاری باید توسط تلگراف قابل انتقال باشند.
- دستگاه رمزنگاری و اسناد رمز شده باید توسط یک فرد قابل حمل باشند.
- راه اندازی سیستم رمزنگاری باید آسان باشد.

نکته : از اصول شش گانه کرشهف تنها اصل دوم اکنون نیز مورد تایید است.

سوال: چرا اصل دوم رمز نگاری قانون کرشهف هنوز نیز مورد تایید دانشمندان است؟

جواب سوال فوق ساده است، اگر یک کلید رمز افشا شود می‌توان به سرعت آن را با یک کلید رمز دیگر جایگزین کرد اما اگر یک الگوریتم و سیستم رمزنگاری لو برود می‌توان هر پیامی را با هر کلیدی رمزگشایی کرد. البته دلایل دیگری نیز وجود دارند که عبارتند از:

- محرمانه نگه داشتن کلید رمز ساده تر از محرمانه نگه داشتن الگوریتم آن است.
- الگوریتم توسط اشخاص گوناگونی نوشته و آزمایش می‌شود که سری نگه داشتن یک راز بین افراد زیاد کاری مشکل است.
- در دسترس بودن الگوریتم امکان بررسی آن الگوریتم توسط تمام متخصصان را فراهم می‌کند، و در صورت یافتن مشکل می‌توان آن الگوریتم را بهبود و یا تغییر داد.
- اگر الگوریتم‌ها در دسترس باشند می‌توان آن‌ها را در اختیار افراد مختلف قرار داد و برای تفسیر یک نوشته رمز شده تنها از کلیدهای مشترک بهره جست.

البته شاید بتوان انقلاب دانش رمزنگاری را با روی کار آمدن کامپیوترهای امروزی مرتبط دانست. کامپیوترهای امروزی با قدرت بالای پردازشی خود امکان استفاده از الگوریتم‌های پیچیده و سخت گیری را فراهم کرده است که در گذشته شاید اصلا امکان پیاده سازی چنین الگوریتم‌هایی وجود نداشت.

از نمونه های عملی رمزنگاری کلاسیک میتوان به ماشین انیگما (Enigma) اشاره کرد. ماشین انیگما در دهۀ ۱۹۲۰ میلادی برای محافظت از ارتباطات تجاری، دیپلماتیک و نظامی عرضه شد. «انیگما» بدست مهندس آلمانی آرتور شربیوس در پایان جنگ جهانی اول اختراع شد. مدل‌های گوناگونی از انیگما ساخته شد که مدل ارتش آلمان رایج‌ترین آن‌ها به‌شمار می‌رود؛ ولی مدل‌های ایتالیایی و ژاپنی نیز دارد. ارتش آلمان نازی مدل ویژه‌ای از این ماشین به نام انیگمای ورماخت را تولید نمود و به منظور رمزنگاری و رمزگشایی پیام‌های نظامی در طول جنگ جهانی دوم بکار می‌برد.

الگوریتم های رمزنگاری پیشرفته یا مدرن

با پدید آمدن رایانه‌ها و افزایش قدرت محاسباتی آنها، دانش رمزنگاری وارد حوزه علوم رایانه گردید و این پدیده، موجب بروز سه تغییر مهم در مسائل رمزنگاری شد:

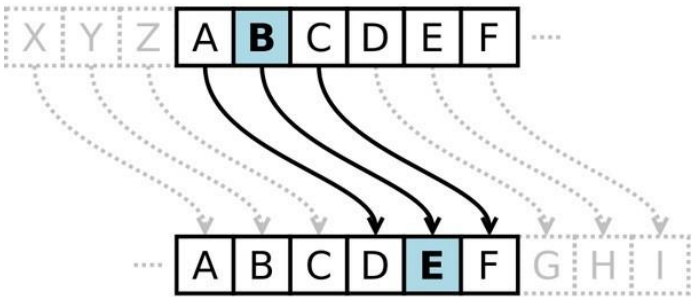
1. وجود قدرت محاسباتی بالا این امکان را پدید آورد که روش‌های پیچیده‌تر و مؤثرتری برای رمزنگاری به وجود آید.
2. روش‌های رمزنگاری که تا قبل از آن اصولا برای رمز کردن پیام به کار می‌رفتند، کاربردهای جدید و متعددی پیدا کردند.
3. تا قبل از آن، رمزنگاری عمدتاً روی اطلاعات متنی و با استفاده از حروف الفبا انجام می‌گرفت؛ اما ورود رایانه باعث شد که رمزنگاری روی انواع اطلاعات و بر مبنای بیت انجام شود.

گسترش و رشد بی سابقه اینترنت باعث ایجاد تغییرات گسترده در نحوه زندگی و فعالیت شغلی افراد ، سازمانها و موسسات شده است. امنیت اطلاعات یکی از مسائل مشترک شخصیت های حقوقی و حقیقی است . اطمینان از عدم دستیابی افراد غیر مجاز به اطلاعات حساس از مهمترین چالش های امنیتی در رابطه با توزیع اطلاعات در اینترنت است . اطلاعات حساس که ما تمایلی به مشاهده آنان توسط دیگران نداریم، موارد متعددی را شامل می شود. برخی از اینگونه اطلاعات بشرح زیر می باشند:

- اطلاعات کارت اعتباری

- شماره های عضویت در انجمن ها
- اطلاعات خصوصی
- جزئیات اطلاعات شخصی
- اطلاعات حساس در یک سازمان
- اطلاعات مربوط به حساب های بانکی

الگوریتم رمزنگاری سزار(Cesar)



الگوریتم رمز سزار یکی از ساده ترین الگوریتم های رمز کلاسیک می باشد که برای اولین بار توسط ژولیوس سزار سردار رومی مورد استفاده قرار گرفت در این الگوریتم جایگزینی حروف ، بر اساس ترتیب در حروف الفبا انجام می شود. بنابراین داریم:

رشته مبنا (حروف الفبای انگلیسی):

ABCDEFGHIJKLMNOPQRSTUVWXYZ

رشته تبدیل یافته:

DEFGHIJKLMNOPQRSTUVWXYZABC

در این تبدیل هرکدام از کلمه ها به کلمه ای تبدیل می شود که در ترتیب حروف الفبا ، سه مرحله بعد از آن قرار دارد. به عنوان مثال A به 3 تا بعد از خودش یعنی D تبدیل می شود. عدد 3 به عنوان کلید این رمزنگاری می باشد.

برای مثال میخواهیم جمله the quick brown fox jumps over the lazy dog را طبق الگوریتم سزار و با کلید رمزنگاری 3 ، رمز کنیم. داریم:

Plaintext: the quick brown fox jumps over the lazy dog

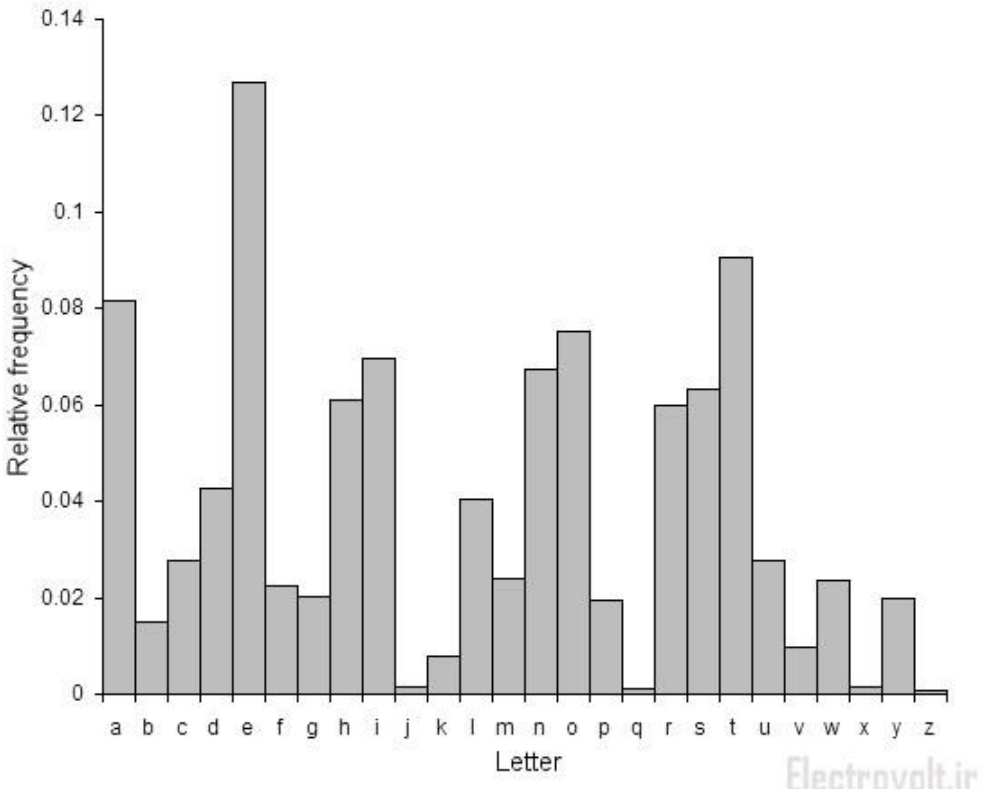
Ciphertext:WKH TXLFN EURZQ IRA MXPSV RYHU WKH ODCB GRJ

برای رمز گشایی نیز عمل عکس فوق انجام می گردد. یعنی هر کلمه به کلمه ی 3 مرحله قبل تر از خود تبدیل می گردد. اکنون وقت آن رسیده است که یک کاغذ و قلم دست بگیرید و خودتان رمزگشایی این عبارت را امتحان کنید.

شکستن الگوریتم رمز سزار

رمز سزار به راحتی و به وسیله روش هایی نظیر تحلیل فراوانی کلمات و نیز الگوی کلمات شکسته می شود.

در روش تحلیل فراوانی حروف ، اگر به جملات در هر زبان توجه کنیم می توانیم این نکته را دریابیم که تعداد وقوع حروف در یک جمله از الگوی خاصی پیروی می کند. به طوری که در زبان انگلیسی حرف E دارای بیشترین وقوع هستند. بنابراین در یک متن رمز شده کلمه ای که دارای بیشترین تکرار است به احتمال زیاد E می باشد. حروف دیگر نیز هر یک فراوانی خاص خود را دارند که میتواند در این روش حدس زده شود. شکل زیر فراوانی حروف را در زبان انگلیسی نشان می دهد.



این نمودار نشان می‌دهد که به طور میانگین 13.5 درصد از حروف به کار رفته در متن های انگلیسی، Eاست که فراوان ترین حرف الفبا می باشد. بنابراین وقتی رمزی از نوع بالا داریم، احتمالاً علامتی که بیش از همه تکرار می‌شود، علامت متناظر E است و فراوان ترین علامت بعد از آن متناظر "T" است. سرنخ های دیگری هم وجود دارد. مثلاً تنها دو کلمه ی یک حرفی در انگلیسی وجود دارد "I" :و "A" و هم چنین "AND"و "THE"کلمات معمولی هستند. با کمک این سرنخ‌ها و کمی آزمایش و خطا میتوان این گونه رمزها را شکست. برای شکستن متون رمز شده به روش سزار در زبان فارسی نیز از همین خاصیت استفاده می کنیم. در زبان فارسی حرف الف دارای بیشترین فراوانی و بعد از آن هم حروف (ی ، ر ، ن ، د) دارای بیشترین فراوانی هستند.

تجزیه و تحلیل رمز (Cryptanalysis)

تجزیه و تحلیل رمز یا شکستن رمز، به کلیه اقدامات مبتنی بر اصول ریاضی و علمی اطلاق می‌گردد که هدف آن از بین بردن امنیت رمزنگاری و در نهایت باز کردن رمز و دستیابی به اطلاعات اصلی باشد. در تجزیه و تحلیل رمز، سعی می‌شود تا با بررسی جزئیات مربوط به الگوریتم رمز و یا پروتکل رمزنگاری مورد استفاده و به کار گرفتن هرگونه اطلاعات جانبی موجود، ضعف‌های امنیتی احتمالی موجود در سیستم رمزنگاری یافته شود و از این طریق به نحوی کلید رمز به دست آمده و یا محتوای اطلاعات رمز شده استخراج گردد.

تجزیه و تحلیل رمز، گاهی به منظور شکستن امنیت یک سیستم رمزنگاری و به عنوان خرابکاری و یک فعالیت ضد امنیتی انجام می‌شود و گاهی هم به منظور ارزیابی یک پروتکل یا الگوریتم رمزنگاری و برای کشف ضعف‌ها و آسیب‌پذیری‌های احتمالی آن صورت می‌پذیرد. به همین دلیل، تجزیه و تحلیل رمز، ذاتاً یک فعالیت خصومت‌آمیز به حساب نمی‌آید؛ اما معمولاً قسمت ارزیابی و کشف آسیب‌پذیری را به عنوان جزئی از عملیات لازم و ضروری در هنگام طراحی الگوریتم‌ها و پروتکل‌های جدید به حساب می‌آورند و در نتیجه تجزیه و تحلیل رمز بیشتر فعالیت‌های خرابکارانه و ضد امنیتی را به ذهن متبادر می‌سازد. با توجه به همین مطلب از اصطلاح حملات تحلیل رمز برای اشاره به چنین فعالیت‌هایی استفاده می‌شود.

تحلیل رمز، در اصل اشاره به بررسی ریاضی الگوریتم (یا پروتکل) و کشف ضعف‌های احتمالی آن دارد؛ اما در خیلی از موارد فعالیت خرابکارانه، به جای اصول و مبنای ریاضی، به بررسی یک پیاده‌سازی خاص آن الگوریتم (یا پروتکل) در یک کاربرد خاص می‌پردازد و با استفاده از امکانات مختلف سعی در شکستن رمز و یافتن کلید رمز می‌نماید. به این دسته از اقدامات خرابکارانه، حملات جانبی گفته می‌شود.

الگوریتم رمز متقارنDES

الگوریتم DES در دهه ۷۰ میلادی در آمریکا به‌عنوان یک استاندارد کدگذاری مطرح شد. به طوری که در سال ۱۹۷۲ مؤسسه بین‌المللی استاندارد و فناوری آمریکا موسوم به NIST (مخفف National Institude Of Standard Technology) اعلام

کرد که به یک الگوریتم برای حفاظت از اطلاعات غیر رده‌بندی شده خود نیاز دارد. این الگوریتم می‌بایست ارزان، قابل دسترس و بسیار مطمئن می‌بود. در سال ۱۹۷۳ ، NISTفراخوانی برای چنین الگوریتمی اعلام نمود ولی هیچ‌یک از الگوریتم‌هایی که در پاسخ به این فراخوان ارائه شدند شرایط لازم را نداشتند. دومین فراخوان در سال ۱۹۷۴ مطرح شد در این زمان شرکت IBM الگوریتم خود را مطرح نمود که به نظر می‌رسید می‌تواند نیازهای NIST را بر طرف کند. این الگوریتم به عنوان یک استاندارد فدرال در سال ۱۹۷۶ تصویب تحت عنوان DES (مخفف Data Encryption Standard) تصویب شد و در سال ۱۹۷۷ منتشر شد. این الگوریتم این‌گونه عمل می‌کند که رشته‌ای از متن اصلی با طول ثابت 64 بیت را به عنوان ورودی می‌گیرد و پس از انجام یک سری اعمال پیچیده روی آن خروجی را که طولی برابر طول ورودی دارد تولید می‌کند DES .همچنین از یک کلید 56 بیتی برای ایجاد رمز استفاده می‌کند و تنها کسانی قادر به رمزگشایی خواهند بود که مقدار کلید را می‌دانند.

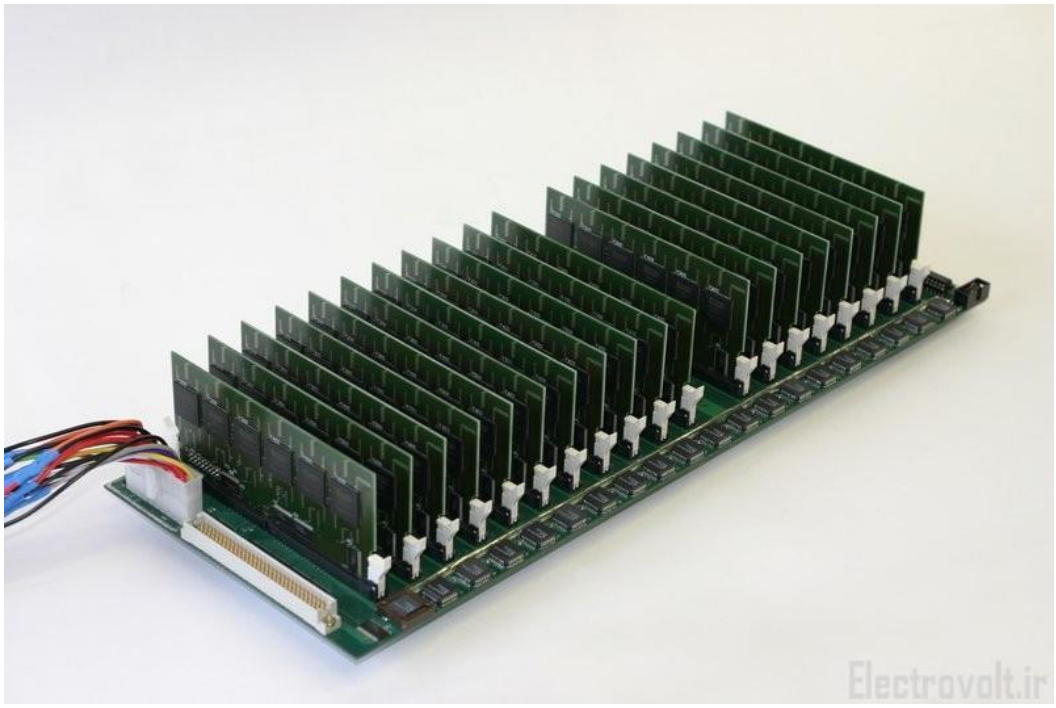
DES

(Data encryption standard)

اساسی‌ترین حمله برای هر رمزی امتحان کردن کلیه مقادیر ممکن برای کلید است. طول کلید، تعداد مقادیر ممکن برای کلید و هم‌چنین عملی بودن این روش را مشخص می‌کند. برای مثال برای DES که طول کلید آن 56 bitمی باشد ، تمام مقادیر ممکن برای این کلید برابر با 2 به توان 56 (یعنی 72057594037927936) می باشد. تردیدی که از ابتدا و حتی قبل از اینکه DESبه‌عنوان استاندارد شناخته شود در مورد DES وجود داشت کافی بودن طول کلید بود. این طول کلید برای آن زمان کافی و غیرقابل شکستن بود اما با گذشت زمان و پیشرفت تکنولوژی دستگاه هایی روی کار آمد که میتوانست DES را بشکند.

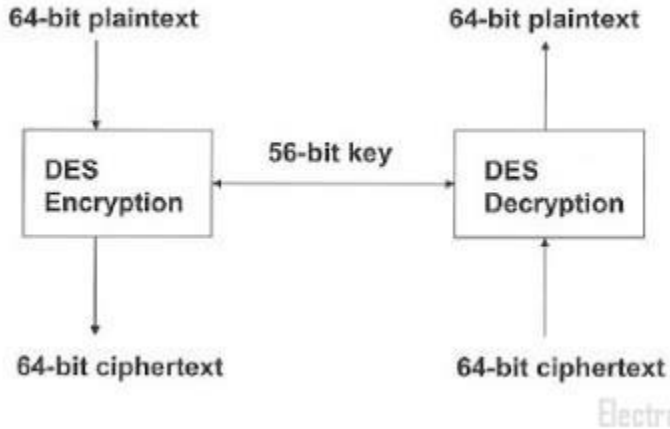
طرح‌های متنوعی برای یک ماشین که قادر به شکستن کلیدهای DES باشد مطرح گردیده است. در سال ۱۹۷۷، Hellmanو Diffieماشینی طراحی کردند که بیست میلیون دلار قیمت داشت و می‌توانست کلید DES را در یک روز پیدا کند. در سال 1993 Wienerیک ماشین جستجوی کلید را پیشنهاد داد که یک‌میلیون دلار قیمت داشت و قادر بود کلید را در مدت هفت ساعت پیدا کند؛ ولی هیچ‌یک از این طرح‌های ابتدایی پیاده‌سازی نشد و هیچ پیاده‌سازی مورد تأیید قرار نگرفت. در سال ۱۹۹۷ مؤسسه RSA security اعلام کرد که به اولین تیمی که بتواند یک پیغام را که با استفاده از DES رمزگذاری شده است را بشکند یک جایزه ده هزار دلاری اعطا خواهد نمود. پروژه DESCHALL برنده این رقابت شد که این کار را با استفاده از زمان بیکاری (idle cycle)هزاران کامپیوتر در اینترنت انجام داد. عملی بودن شکست DES با اختراع یک DES-cracker توسط EFF در سال ۱۹۹۸ بر همگان روشن شد این ماشین قیمتی حدود دویست و پنجاه‌هزار دلار داشت و انگیزه این گروه بر ای اختراع این

ماشین، این بود که نشان دهند که DES هم چنان‌که ازلحاظ تئوری قابل شکست است ازلحاظ عملی نیز می‌توان آن را شکست. این ماشین کلید را با استفاده از روش جستجوی جامع فضای کلید درطی مدت‌زمان کمی بیش از دو روز پیدا می‌کند.



تنها DES-cracker تأییدشده پس از ماشینEFF ، ماشین COPOCOBANA که در آلمان ساخته شد و برخلاف EFF از مدارات مجتمع در دسترس و قابل پیکربندی دوباره ساخته‌شده است در این ماشین صد وبیست عدد FPGA از نوع XILINX Spartan- 1000موازی باهم کار می‌کنند آن‌ها در ماژول‌های DIMM 20 گروه‌بندی‌شده‌اند هرکدام از این ماژول‌ها شامل شش FPGA می‌باشند. استفاده از سخت‌افزارهای قابل پیکربندی دوباره سبب می‌شود که این ماشین برای شکستن کدهای دیگر نیز قابل‌استفاده باشد. یکی از جنبه‌های جالب این ماشین، فاکتور هزینه آن است این ماشین با ده هزار دلار می‌تواند ساخته شود کاهش هزینه با ضریب ۲۵ نسبت به EFF نشان‌دهنده پیشرفته‌ای متوالی درزمینهٔ سخت‌افزارهای دیجیتالی است. ضعف ساختاری DES مربوط به‌اندازه کوچک کلید و بلوک خود می‌باشد که این مشکل در ورژن 3 DESحل‌شده است. 3 DESاز یک کلید با طول برابر DES یعنی 112 بیت استفاده می کند. به طوری که ابتدا پیغام توسط 56 بیت اول و سپس توسط 56 بیت دوم رمز می گردد. در نتیجه فضای کلید های ممکن 2 به توان 112 می شود که عدد بسیار بزرگی بوده و شکستن آن کار بسیار دشواری است .

برنامه نویسی الگوریتم DES و DES3



برنامه نویسی این الگوریتم از 0 تا 100 برای یک برنامه نویس حرفه ای بین 2 تا 3 روز (تمام وقت) طول می کشد. باید الگوریتم جزء به جزء خوانده شده و برنامه نویسی شود. ما این الگوریتم را در کامپایلر CodeVision AVR و به زبان برنامه نویسی C پیاده کردیم که برای تمامی میکروکنترلرهای AVR (و با کمی تغییرات برای همه میکروکنترلرها) قابل استفاده می باشد. هم اکنون میتوانید این پروژه را از لینک زیر دانلود کرده و قابلیت های رمزنگاری را به پروژه خود بیافزایید.

لینک دانلود سورس پروژه پیاده سازی الگوریتم DES و DES3

الگوریتم رمز متقارنAES

به علت وجود ضعف های ساختاری در DES برای بار دوم موسسه NIST به همراه سازمان فدرال آمریکا یعنی FIPS (مخفف Federal Information Processing Standards) به همراهی هم رقابتی چند مرحله ای در ژانویه 1977 فراخوان دادند تا استاندارد جدیدی برای رمزنگاری انتخاب شود. در مرحله اول که سال 1998 برگزار شد این رقابت از میان صدها الگوریتم ، تنها 5 الگوریتم زیر پذیرش شد:

- Rijndael
- Serpent
- Twofish
- RC6
- MARS

اما در مرحله دوم از میان این 5 الگوریتم Rijndael برنده شد و توانست به عنوان استاندارد AES شناخته شود. این الگوریتم توسط دو جوان بلژیکی Rijmen و Daemen طراحی شده و توسط NIST استاندارد سازی و در سند FIPS 197 تحت نام AES (Advanced Encryption Standard) عرضه شد. این رمزنگاری به عنوان استاندارد دولت فدرال در ماه می ۲۰۰۲ بعد از تأیید توسط وزارت بازرگانی آمریکا به کار گرفته شد .AES همچنین در استاندارد ISO/IEC 18033-3 قرار گرفته‌است.

اندازه کلید استفاده شده در رمزAES ، تعداد تکرارهای چرخه‌های تبدیل (transformation) را تعیین می‌کند که ورودی، با نام متن عادی (plaintext) را به خروجی نهایی با نام متن رمز شده (ciphertext) تبدیل می‌نماید. تعداد چرخه‌های تکرار به صورت زیر است:

- ۱۰ چرخه تکرار برای کلیدهای ۱۲۸ بیتی.
- ۱۲ چرخه تکرار برای کلیدهای ۱۹۲ بیتی.
- ۱۴ چرخه تکرار برای کلیدهای ۲۵۶ بیتی.

هر تکرار شامل چندین مرحله پردازشی است، که یک مرحله بستگی به کلید رمزنگاری دارد. مجموعه‌ای از چرخه‌های معکوس برای تبدیل متن رمز شده به متن اصلی استفاده می شود که از همان کلید رمزنگاری استفاده می کند.

دلایل شگفتی سازی روش رمزنگاری Rijndael در دنیای رمزنگاری کلید متقارن:

1) عدم پیروی این روش از الگوی سنتی روشهای فیستلی و مبتنی بر حالت خاصی از «میدان های گالوا (Galios Field) « می باشد.
2) انتخاب این روش بعنوان استاندارد دولت فدرال آمریکا در فضائی آزاد و بدون اعمال نفوذ عوامل جاسوسی یا امنیتی ایالات متحده و ثبت تحت قوانین غیر انحصاری و بهره برداری آزاد از آن در محصولات مختلف
3) امنیت بسیار بالا، سرعت زیاد، پیاده سازی ساده، فضای حافظه مورد نیاز کم و قابلیت انعطاف بالا

اختلاف بین Rijndael و AES

1) درRijndael ، طول کلید و طول بلوک داده می تواند 128، 192 و 256 بیت باشند لذا می توان گفت که Rijndael دارای 9 انتخاب متفاوت برای رمزنگاری اطلاعات است.

2) در AES طول بلوک داده صرفاً باید 128 بیتی (معادل 4 کلمه 32 بیتی) باشد ولی طول کلید را از بین یکی از مقادیر 128، 192 و 256 بیتی انتخاب می کنند بدین ترتیب AES کلاً دارای سه انتخاب است.

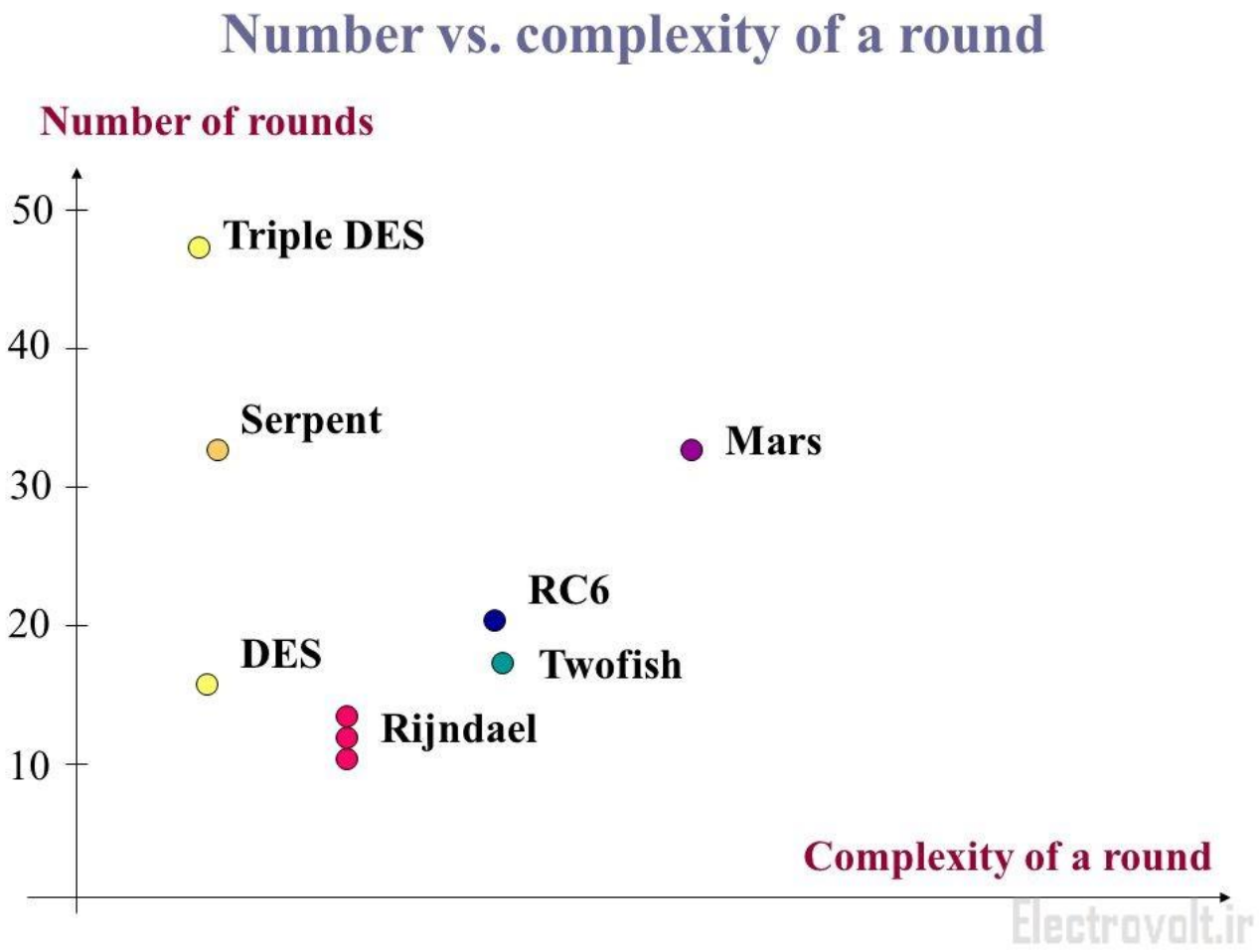
این پارامترها باید به الگوریتم رمزنگاری وارد شوند:

Nb: طول بلوک داده ورودی (در AES همیشه 4 است)

Nk: طول کلید (4 ، 6 یا 8)

Nr: تعداد دورهای رمزنگاری (به طول کلید وابسته است)

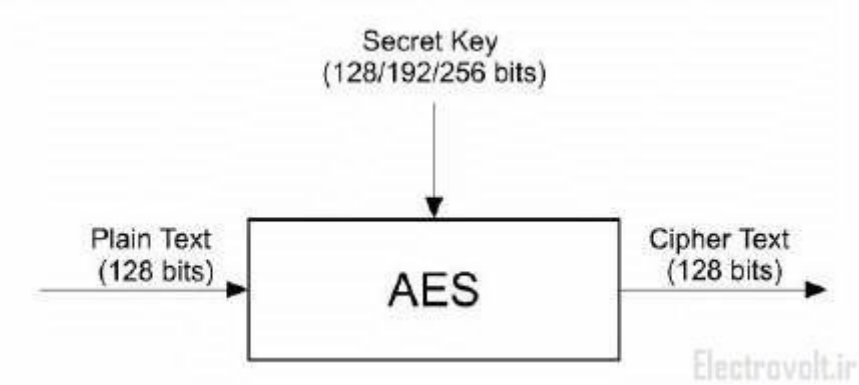
مقایسه پیچیدگی الگوریتم های مختلف رمزنگاری:



در حال حاضر رمزگشایی و ارزیابی الگوریتم های رمزنگاری توسط دو روش متداول یعنی رمزگشایی از طریق حملات کلاسیک و دیگری رمزگشایی از طریق حملات سخت افزاری انجام می گیرد. بر روی AES نیز حملات بسیار زیادی انجام شده است اما در

بخش کلاسیک تا کنون هیچ حمله موفقیت آمیزی صورت نگرفته است. و این بدین معنی است که AES در الگوریتم رمزنگاری خود دارای ضعف ساختاری نمی باشد. اما در سال های اخیر در بخش حملات سخت افزاری شکستن بسیاری از الگوریتم ها ممکن شده است. یکی از انواع حملات سخت افزاری حملات کانال جانبی است. در حملات کانال جانبی منبعی از اطلاعات مرتبط با پیاده سازی فیزیکی توابع رمز نگاری ، در سخت افزار مربوطه به دست می آید. این حملات از نشت اطلاعات همانند مصرف توان ، تشعشعات الکترومغناطیسی یا زمان محاسبات عملیات رمزنگاری استفاده می کند تا کلید رمز نگاری را به دست آورد. استفاده از حملات کانال جانبی نوعی میان بر در امر رمزگشایی است چرا که دارای پیچیدگی کمتر و تاثیر زیاد می باشد.

برنامه نویسی الگوریتمAES



برنامه نویسی این الگوریتم از 0 تا 100 برای یک برنامه نویس حرفه ای بین 2 تا 3 روز (تمام وقت) طول می کشد. باید الگوریتم جزء به جزء خوانده شده و برنامه نویسی شود. ما این الگوریتم را در کامپایلر CodeVision AVR و به زبان برنامه نویسی C پیاده کردیم که برای تمامی میکروکنترلرهای AVR (و با کمی تغییرات برای همه میکروکنترلرها) قابل استفاده می باشد. هم اکنون میتوانید این پروژه را از لینک زیر دانلود کرده و قابلیت های رمزنگاری را به پروژه خود بیافزایید.

لینک دانلود سورس پروژه پیاده سازی الگوریتمAES

سرویس های رمزنگاری

به طور کلی، سرویس رمزنگاری، به قابلیت و امکانی اطلاق می‌شود که بر اساس فنون رمزنگاری حاصل می‌گردد. قبل از ورود رایانه‌ها به حوزه رمزنگاری، تقریباً کاربرد رمزنگاری محدود به رمز کردن پیام و پنهان کردن مفاد آن می‌شده‌است. اما در رمزنگاری پیشرفته سرویس‌های مختلفی از جمله موارد زیر ارائه گردیده است:

- حفظ محرمانگی یا امنیت محتوا : ارسال یا ذخیره اطلاعات به نحوی که تنها افراد مجاز بتوانند از محتوای آن مطلع شوند، که همان سرویس اصلی و اولیه پنهان کردن مفاد پیام است.
- حفظ صحت داده‌ها یا سلامت محتوا : به معنای ایجاد اطمینان از صحت اطلاعات و عدم تغییر محتوای اولیهٔ آن در حین ارسال است. تغییر محتوای اولیه اطلاعات ممکن است به صورت اتفاقی (در اثر مشکلات مسیر ارسال) یا به صورت عمدی باشد.
- احراز هویت یا اصالت سنجی محتوا : به معنای تشخیص و ایجاد اطمینان از هویت ارسال‌کننده اطلاعات و عدم امکان جعل هویت افراد می‌باشد.
- عدم انکار پذیری : به این معنی است که ارسال‌کننده اطلاعات نتواند در آینده ارسال آن را انکار یا مفاد آن را تکذیب نماید.

چهار مورد بالا، سرویس‌های اصلی رمزنگاری تلقی می‌شوند و دیگر اهداف و سرویس‌های رمزنگاری، با ترکیب چهار مورد بالا قابل حصول می‌باشند.

این سرویس‌ها مفاهیم جامعی هستند و می‌توانند برای کاربردهای مختلف پیاده‌سازی و استفاده شوند. به عنوان مثال سرویس اصالت محتوا هم در معاملات تجاری اهمیت دارد و هم در مسائل نظامی و سیاسی مورد استفاده قرار می‌گیرد. برای ارائه کردن هر یک از سرویس‌های رمزنگاری، بسته به نوع کاربرد، از پروتکل‌های مختلف رمزنگاری استفاده می‌شود.

پروتکل های رمزنگاری

به طور کلی، یک پروتکل رمزنگاری، مجموعه‌ای از قواعد و روابط ریاضی است که چگونگی ترکیب کردن الگوریتم‌های رمزنگاری و استفاده از آن‌ها به منظور ارائه یک سرویس رمزنگاری خاص در یک کاربرد خاص را فراهم می‌سازد.

- معمولاً یک پروتکل رمزنگاری مشخص می‌کند که اطلاعات موجود در چه قالبی باید قرار گیرند.
- چه روشی برای تبدیل اطلاعات به عناصر ریاضی باید اجرا شود
- کدامیک از الگوریتم‌های رمزنگاری و با کدام پارامترها باید مورد استفاده قرار گیرند
- روابط ریاضی چگونه به اطلاعات عددی اعمال شوند
- چه اطلاعاتی باید بین طرف ارسال‌کننده و دریافت‌کننده رد و بدل شود
- چه مکانیسم ارتباطی برای انتقال اطلاعات مورد نیاز است

به عنوان مثال می‌توان به پروتکل تبادل کلید دیفی-هلمن برای ایجاد و تبادل کلید رمز مشترک بین دو طرف اشاره نمود.
پروتکل تبادل کلید دیفی ، هلمن یک پروتکل رمزنگاری است که با استفاده از آن، دو نفر یا دو سازمان، می‌توانند بدون نیاز به هر گونه آشنایی قبلی، یک کلید رمز مشترک ایجاد و آن را از طریق یک مسیر ارتباطی غیر امن، بین خود تبادل نمایند. این پروتکل، اولین روش عملی مطرح شده برای تبادل کلید رمز درمسیرهای ارتباطی غیر امن است و مشکل تبادل کلید رمز در رمزنگاری کلید متقارن را آسان می‌سازد. این پروتکل، در سال ۱۹۷۶ توسط دو دانشمند رمزشناس به نام‌های ویتفیلد دیفی و مارتین هلمن طراحی شده و در قالب یک مقاله علمی منتشر گردیده است. مطرح شدن این پروتکل، گام مهمی در معرفی و توسعهٔ رمزنگاری کلید نامتقارن به حساب می‌آید.

الگوریتم رمز نامتقارنRSA

در تمام الگوریتم های رمزنگاری با کلید متقارن، فرستنده و گیرنده پیام باید کلید رمز را بدانند. وقتی فرستنده پیام از کلیدی یکتا و سری برای رمزنگاری استفاده می کند و گیرندگان پیام از همان کلید برای رمزگشایی بهره می برند، افشای کلید رمز از طریق یکی از گیرندگان پیام ،امنیت همه را به خطر می اندازد. در چنینی وضعیتی فرستنده مجبور خواهد بود با یکایک گیرندگان بطور مجزا بر سر یک کلید سری متقارن توافق کند تا هر یک گیرنده کلید مخصوص خود را داشته و افشای آن در امنیت دیگران خللی ایجاد نکند. در این حالت فرستنده پیام باید به تعداد گیرندگان خود کلید تعریف کرده و از آنها نگهداری کند. تعریف مثلا ده ها هزار کلید متقارن برای کاربران و ذخیره و بازیابی مطمئن آنها به نوبه خود مشکل بزرگی است.

در الگوریتمهای کلید عمومی برای رمزنگاری و رمزگشایی از دو کلید کاملا متفاوت استفاده می شود: «کلید عمومی» و «کلید خصوصی«

کلید عمومی برای رمزنگاری اطلاعات بکار می رود و همه آن را میدانند ، زیرا از این کلید صررفا برای رمز کردن اطلاعات استفاده می شود و دشمنان با در اختیار داشتن آن نخواهند توانست داده ها ی رمز شده توسط دیگران را از رمز خارج کنند.

کلید خصوصی کلیدی است که داده های رمز شده با آن رمز گشایی می شوند. این کلید راهیچکس حتی معتمدین و دوستان نمی دانند. بدین ترتیب هر موجودیت در سطح شبکه (اعم از کاربر، ماشین یا پروسه ها) نیاز به دو کلید مستقل دارد که فقط یکی از آنها حساس و سری است و باید به دقت از آن نگهداری کرد.ماهیت الگوریتم رمزنگاری به گونه ای است که در عمل نمی توان با در دست داشتن کلید عمومی کلید خصوصی را استنتاج کرد.

در سال ۱۹۷۸ سه نفر به نامهای ری‌وست ،شامیر و اَدِلمن الگوریتمی را برای پیاده سازی رمزنگاری کلید عمومی با یک جفت کلید معرفی کردند که به RSA شهرت یافت و در طول سه دهه اخیر بطور گسترده ای مورد استفاده قرار گرفته و در گذر زمان، سخت افزار و نرم افزارهای بهینه آن به بازار عرضه شد. اگر چه بعدها الگوریتم قویتری بنام El Gamal ابداع شد اما هنوز هم روش RSA در صدر فهرست الگوریتم های کلید عمومی قرار دارد.