معرفی و اصطلاحات

رمزنگاری علم کدها و رمزهاست. یک هنر قدیمی است و برای قرنها به منظور محافظت از پیغامهایی که بین فرماندهان، جاسوسان، عشاق و دیگران ردوبدل می‌شده استفاده شده است، تا پیغامهای آنها محرمانه بماند.

هنگامی که با امنیت اطلاعات سروکار داریم، نیاز به اثبات هویت فرستنده و گیرنده پیغام داریم و در ضمن باید از عدم تغییر محتوای پیغام مطمئن شویم. این سه موضوع یعنی محرمانه بودن، تصدیق هویت و جامعیت در قلب امنیت ارتباطات مدرن قرار دارند و می‌توانند از رمزنگاری استفاده کنند.

اغلب این مساله باید تضمین شود که پیغام فقط می‌تواند توسط کسانی خوانده شود که پیغام برای آنها ارسال شده است و دیگران این اجازه را ندارند.

در الگوریتم‌های رمزنگاری همیشه دو جزء اصلی مطرح بوده است: یک الگوریتم و یک کلید. الگوریتم یک مبدل یا فرمول ریاضی است. تعداد کمی الگوریتم قدرتمند وجود دارد که بیشتر آنها بعنوان استانداردها یا مقالات ریاضی منتشر شده‌اند. کلید، یک رشته از ارقام دودویی (صفر و یک) است که بخودی‌خود بی‌معنی است. در الگوریتم‌های رمزنگاری مدرن فرض بر این است که الگوریتم شناخته شده است یا می‌تواند کشف شود و این کلید است که باید مخفی نگاه داشته شود و کلید است که در هر مورد استفاده تغییر می‌کند. رمزگشایی ممکن است از همان جفت الگوریتم و کلید یا جفت متفاوتی استفاده کند.

الگوریتم‌ها:

طراحی الگوریتمهای رمزنگاری مقوله‌ای است که بیشتر به ریاضیدانان مربوط می‌شود. از اولین الگوریتم‌های رمزنگاری، الگوریتم رمز سزار است. در این روش شماره حرف در الفبا با عدد کلید رمز نگاری جمع می‌شود و باقیمانده مجموع بر تعداد حروف الفبا حساب شده و حرف معادل با این عدد در الفبا بجای حرف اولیه از متنی اولیه (Plain text)، در متن رمز شده (Cipher text) قرار می‌گیرد.

طراحان سیستمهایی که در آنها از رمزنگاری استفاده می‌شود، باید از نقاط قوت و ضعف الگوریتمهای موجود مطلع باشند و برای تعیین الگوریتم مناسب قدرت تصمیم‌گیری داشته باشند. اگرچه رمزنگاری از اولین کارهای شانون(Shannon) در اواخر دهه40 و اوایل دهه50 بشدت پیشرفت کرده است، اما کشف رمز نیز پا به پای رمزنگاری به پیش آمده است و الگوریتمهای کمی هنوز با گذشت زمان ارزش خود را حفظ کرده‌اند. بنابراین تعداد الگوریتمهای استفاده شده در سیستمهای کامپیوتری عملی و در سیستمهای بر پایه کارت هوشمند بسیار کم است. الگوریتمهای مطرح شده در طی این دوران را می‌توان به دسته کلی تقسیم کرد. الگوریتم‌های کلید

متقارن و الگوریتم‌های کلید نامتقارن که در ادامه به شرح آنها می‌پردازیم.

الگوریتم‌های کلید متقارن:

یک الگوریتم متقارن از یک کلید برای رمزنگاری و از همان کلید برای رمزگشایی استفاده می‌کند. بیشترین شکل استفاده از رمزنگاری که در کارتهای هوشمند و البته در بیشتر سیستمهای امنیت اطلاعات وجود دارد data encryption algorithm یا DEA است که بیشتر بعنوان DES شناخته می‌شود و یک الگوریتم کلید متقارن است. DES یک محصول دولت ایالات متحده است که امروزه بطور وسیعی به عنوان یک استاندارد بین‌المللی شناخته می‌شود. بلوکهای64بیتی اطلاعات توسط یک کلید که معمولاً 56بیت طول دارد، رمزنگاری و رمزگشایی می‌شوند. DES از نظر محاسباتی ساده است و به راحتی می‌تواند توسط پردازنده‌های کند (بخصوص آنهایی که در کارتهای هوشمند وجود دارند) انجام گیرد.

امنیت این روش بسیار وابسته به مخفی ماندن کلید از جانب دو طرف ارتباطی دارد. بنابراین برای استفاده در دو موقعیت مناسب است: هنگامی که کلیدها می‌توانند به یک روش قابل اعتماد و امن توزیع و ذخیره شوند یا جایی که کلید بین دو سیستم مبادله می‌شوند که قبلاً هویت یکدیگر را تایید کرده‌اند و عمر کلیدها بیشتر از مدت تراکنش طول نمی‌کشد. رمزنگاری DESعموماً برای حفاظت اطلاعات از شنود در طول انتقال استفاده می‌شود.

الگوریتم‌های کلید نامتقارن:

الگوریتم‌های کلید نامتقارن از یک کلید برای رمزنگاری و از یک کلید دیگر برای رمزگشایی استفاده می‌کنند. این الگوریتم‌ها اجازه می‌دهند که یک جزء (کلید عمومی یا public key که به وسیله آن رمز نمودن انجام می‌شود) منتشر شود، در حالیکه دیگری (کلید اختصاصی یا private key) توسط صاحبش حفظ خواهد شد. فرستنده پیام، متن را با کلید عمومی گیرنده کد می‌کند و گیرنده آن را با کلید اختصاصی خودش رمزگشایی می‌کند. به عبارتی تنها با کلید اختصاصی گیرنده می‌توان متن کد شده را به متن اولیه صحیح تبدیل کرد. یعنی حتی فرستنده نیز اگرچه از محتوای اصلی پیام مطلع است اما نمی‌تواند از متن کدشده به متن اصلی دست برسد، بنابراین پیام کدشده برای هر گیرنده‌ای به جز گیرنده مورد نظر فرستنده، بی‌معنی خواهد بود. معمولترین و مشهورترین الگوریتم نامتقارن به عنوان RSA شناخته می‌شود (RSA مخفف حروف اول نام پدیدآورندگان آن یعنی Rivest، Shamir و Adelman است). می‌توان از یک سیستم نامتقارن برای نشان دادن اینکه فرستنده پیام همان شخصی است که ادعا می‌کند استفاده کرد که این عمل اصطلاحاً امضای دیجیتال نام دارد. امضا، متن اصلی را با استفاده از کلید اختصاصی رمز می‌کند؛ رمزگشایی عملیات مشابه‌ای روی متن رمزشده اما

با استفاده از کلید عمومی است. برای تایید امضا بررسی می‌کنیم که آیا این نتیجه با اطلاعات اولیه یکسان است یا خیر.

به بیان ساده‌تر چنانچه متنی از شخصی برای دیگران منتشر شود، این متن شامل متن اصلی و همان متن اما رمز شده توسط کلید اختصاصی همان شخص است. حال اگر متن رمز شده توسط کلید عمومی آن شخص که شما از آن مطلعید رمزگشایی شود، مطابقت متن حاصل و متن اصلی نشان دهنده صحت فرد فرستنده آن است، به این ترتیب امضای فرد تصدیق می‌شود. افرادی که از کلید اختصاصی این فرد اطلاع ندارند قادر به ایجاد متن رمز شده نیستند، به طوری که با رمزگشایی توسط کلید عمومی این فرد به متن اولیه تبدیل شود. بدین ترتیب فرد فرستنده نمی‌تواند منکر فرستادن متن شود، زیرا کسی به جز او نمی‌توانسته آن متن را به شکل مطلوب امضا کند.

برای رمزنگاری در الگوریتم RSA از این فرمول استفاده می‌شود:

$$X=Y^k(\text{mod } r)$$

که Xمتن کد شده، Yمتن اصلی، kکلید اختصاصی و r حاصلضرب دو عدد اولیه بزرگ است که با دقت انتخاب شده‌اند. در بخش بعد این روش را با جزئیات بیشتر بررسی خواهیم کرد.

سایر الگوریتم‌های کلید نامتقارن شامل سیستمهای لگاریتم گسسته می‌شوند مانند Diffie-Hellman، ElGamal و سایر طرحهای چندجمله‌ای و منحنی‌های بیضوی. بسیاری از این طرحها عملکردهای یک‌طرفه‌ای دارند که اجازه تایید هویت را می‌دهند اما رمزنگاری ندارند.

شرح الگوریتم RSA:

شخص A را به عنوان گیرنده و شخص B را به عنوان فرستنده در نظر بگیرید:

1. ابتدا شخص A دو عدد اول انتخاب می کند. ما از اعداد p=23 و q=41 استفاده می‌کنیم.
2. شخص A دو عدد p و q را در هم ضرب می‌کند تا به N=p*q=(23)(41)=934 برسد. 934 کلید عمومی A است که آنرا به B می‌دهد (و همچنین به بقیه افرادی که قصد فرستادن اطلاعات را به A دارند).
3. شخص A همچنین عدد دیگری چون e را که نسبت به M=(p-1)(q-1) اول باشد منتشر می‌کند. یعنی(M, e) =1.در این مورد خاص M= (p-1)*(q-1)= 22*40=880 بنابراین

e=7مناسب است. e نیز قسمتی از کلید عمومی است و باید علاوه بر p.q ، e نیز به شخص B گفته شود.

4. حالا B اطلاعات کافی برای به رمز در آوردن یک پیغام برای A را دارد. فرض کنید که در این مثال پیغامی مورد نظر عدد P=35 باشد.

5. شخص B مقدار $C=P^e \ mod \ N = 35^7 (mod\ 943)$را حساب می کند.

64339296875

6. $35^7=64339296875$ و 64339296875 mod 943=5455

عدد 545 در واقع پیغام رمزگذاری شده‌ای است که B به A می‌فرستد.

7. حالا A می خواهد 545 را رمزگشایی کند برای این کار او عددی را چون d نیاز دارد طوری که:

$e*d \ mod \ M=1$

یا در این مثال $7*d(mod\ 880)=1$ که برای فرد A مقدار d عدد 503 خواهد بود، زیرا داریم

$7*503 = 3521 = (4*(880) +1) \ (mod\ 800)$

8. برای پیدا کردن متن رمزگشایی شده، A باید

$C^d \ mod \ N = 545^{503} \ mod \ 943$

را حساب کند. در ابتدا به نظر می‌آید که با توجه به توان‌های بزرگی که در این محاسبات وجود دارد، این کار بسیار دشوار است اما توجه کنید که:

503 = 256+128+64+32+16+4+2+1

(که در واقع بسط دودویی عدد 503 است) بنابر این خواهیم داشت

$545^{503} = 545^{256+128+64+32+16+4+2+1} = 545^{256} 545^{128} \ \dots \ 545^1$

اما از آنجا که ما تنها علاقمند به پیدا کردن باقیمانده به پیمانه 943 هستیم، می توانیم با محاسبه باقیمانده تمام عامل‌های ضربی در این پیمانه به مقصودمان برسیم و این کار را می توانیم با به توان 2 رساندن‌های متوالی عدد 545 انجام دهیم زیرا تمام توان‌های عامل‌های ضربی، مضربی از 2 هستند. به عنوان مثال برای اینکه به باقیمانده توان 4ام عدد 545 برسیم داریم:

$545^2 \bmod 943 = 545*545 = 297025 \bmod 943 = 923$

با به توان 2 رساندن مجدد داریم:

$545^4 \bmod 943 = (545^2)^2 (\bmod 943) = 923*923 = 851929 (\bmod 943) = 400$

و به همین ترتیب برای توان های بالا تر عمل می کنیم و در نهایت به نتیجه زیر می رسیم:

$545^{256} \bmod 943 = 324$

بنابراین عدد مورد نظر ما به صورت زیر بدست می آید:

$545^{503} \bmod 943 = 324*18*215*795*857*400*923*545 \bmod 943 = 35$

بوسیله این عملیات شخص A می تواند پیام B را رمزگشایی کند و به پیام اصلی یعنی P=35 برسد.

تفاوت رمزنگاری و توابع Hash:

الگوریتم‌های رمزنگاری توابعی معکوس‌پذیر هستند و در صورت دانستن کلید، معکوس آنها به راحتی محاسبه می‌شود. در حالی که توابع Hash توابعی معکوس‌ناپذیرند. کاربرد رمزنگاری در جاهایی که بخواهیم پیغامی را رمز کرده و از کانال ناامن عبور دهیم و سپس در مقصد آن را رمزگشایی کرده و متن اولیه را استخراج کنیم می‌باشد، در حالی که توابع Hash در جاهایی استفاده می‌شوند که بخواهیم درستی یک عبارت (مثلا یک رمز عبور) را بررسی کنیم، در این صورت با توجه به اینکه ذخیره کردن مستقیم کلمه عبور جهت مقایسه، به علت امکان دسترسی دیگران امنیت را کاهش می‌دهد، می‌توان Hash شده آنها را به همراه تابع Hash ذخیره کرده و عبارت ورودی را Hash کرده و با Hashهای ذخیره شده مقایسه می‌کنیم، در این حالت حمله کننده در صورت داشتن Hashهای ذخیره شده نیز با توجه به معکوس ناپذیر بودن توابع Hash نمی‌تواند اصل کلمه‌های عبور را بیابد. از توابع Hash می‌توان به خانواده‌ی MDx، SHA-1، RIPEM، D-128 و RIPEMD-160 اشاره کرد که البته حملاتی جهت کشف معکوس MD-4 و MD-5 یافته شده است.