رمزگذاری

Bahram Arbabi

مقدمه

رمزگذاری یعنی تبدیل اطلاعات به یک شکل غیر قابل فهم و انتقال آن و سپس برگرداندن اطلاعات رمز شده به حالت اولیه و قابل خواندن. به مرحله تبدیل به شکل غیر قابل فهم عمل رمزنگاری یا Encryption می گویند و برگرداندن به حالت قابل فهم را رمزگشایی یا Decryption می گویند. رمزنگاری علمی است که به وسیله آن می‌توان اطلاعات را بصورتی امن منتقل کرد حتی اگر مسیر انتقال اطلاعات (کانالهای ارتباطی) ناامن باشد.

الگوریتم های رمزگذاری دو دسته اند یکی الگوریتم های کلید متقارن و دیگری الگوریتم های کلید عمومی، که ما دراین مقاله به الگوریتم های مبتنی بر کلید عمومی و کاربرد نظریه اعداد در آن پرداخته ایم. الگوریتم های رمزنگاری متقارن الگوریتم هایی هستند که که کلید رمزگزاری و کلید رمز گشایی آنها می تواند حساب بشوند و بلعکس، و در تمام الگوریتم های رمزگاری متقارن کلید رمز نگاری و کلید رمز گشایی یکی هستند، این الگوریتم‌ها نیز الگوریتم های کلید رمزی یا تک کلیدی نیز نامیده میشود. امنیت الگوریتم های متقارن در کلید آن می‌باشد، به این معنی کی اگر یک کلید فاش یا شکسته شود، هر کس میتواند آن داده های رمزگزاری شده را باز و همچنین رمزنگاری کند. و اگر بخواهیم ارتباط امنیت داشته باشد، در ان موقع کلید باید امنیت داشته باشد. و الگوریتم های آن‌به صورت زیر می‌باشد. که در آن M پیغام و C داده رمزنگاری شده، E عمل رمزنگاری و D عمل رمز گشایی می‌باشد.

$$E_{key}(M) = C$$

$$D_{key}(C) = M$$

الگوریتم های متقارن به دو دسته تقسیم میشود، یکی الگوریتم های مبتنی بر بیت که عمل رمزنگاری بر روی بیت انجام میشود، دستهٔ دوم الگوریتم های بلوکی هستند که عمل رمز نگاری روی یک دسته از بیتها انجام میشود که امروزه بیشتر از الگوریتم های بلوکی استفاده میشود.

دسته دوم الگوریتم های کلید عمومی هستند که به این الگوریتم ها نیز الگوریتم نامتقارن نیـز گفتـه میـشود. تفاوت بارز این الگوریتم ها با الگوریتم های متقارن این است که کلیدی که برای رمزنگاری استفاده میشود با کلیدی که برای رمزگشایی استفاده میشود فرق می‌کند. و یا بـه بیـان دیگـر کلیـد رمزگـشایی نمیتوانـد از کلید رمزگزاری استخراج شود و یا بسیار سخت و زمانگیر می‌باشد. به این الگوریتم ها کلید عمـومی گفتـه میشود چون کلید رمزنگـاری میتوانـد عمـومی باشـد. بـه‌ایـن ترتیـب کـه یـک تعـداد افـراد میتواننـد عمـل رمزگزاری را انجام دهند ولی یک فرد خاص فقط میتواند داده های رمزگزاری شده را رمز گشایی کند. در این سیستم کلید رمزگزاری را کلید عمومی public-key و کلید رمزگشایی راکلید خصوصی private-key میگویند.

که الگوریتم کلی آن به صورت زیر می‌باشد.

$$E_{public\text{-}key}(M) = C$$
$$D_{private\text{-}key}(C) = M$$

مقایسه رمزنگاری الگوریتم های متقارن و الگوریتم های کلید عمومی:

بحث های زیادی شده که کدام یک از این الگوریتم ها بهترند اما جواب مشخصی ندارد. البته بررسی هایی روی این سوال شده به طور مثال Needham و Schroeder بعد از تحقیق به این نتیجه رسیدند که طول پیغامی که با الگوریتم های متقارن میتواند رمزنگاری شود از الگوریتم های کلید عمومی کمتـر اسـت. و بـا تحقیق به این نتیجه ریسیدند که الگوریتم های متقارن الگوریتم های بهینه تری هستند. امـا وقتـی کـه بحـث امنیت پیش می آید الگوریتم های کلید عمومی کارایی بیشتری‌دارند. و بطور خلاصـه مـی‌تـوان گفـت کـه الگوریتم های متقارن دارای سرعت بالاتر و الگوریتم هـای کلیـد عمـومی دارای امنیـت بهتـری هـستند. در ضمن گاهی از سیستم ترکیبی از هردو الگوریتم استفاده میکننـد کـه بـه ایـن الگـوریتم هـا الگـوریتم هـای ترکیبی (hybrid) گفته میشود.

اما اگر به طور دقیق تر به این دو نگاه کنیم آنگاه متوجه خواهیم شـد کـه الگـوریتم هـای کلیـد عمـومی و الگوریتم های کلید متقارن دارای دو ماهیت کاملا متفاوت هستند و کار برد هـای متفـاوتی دارنـد بـه طـور مثال در رمزنگاری های ساده که حجم داده ها بسیار زیاد است از الگوریتم متقارن استفاده میشود زیـرا داده ها با سرعت بالاتری رمزنگاری و همچنین رمزگشایی شود. امـا در پروتکـل هـایی کـه در اینترنـت اسـتفاده میشود، برای رمز نگری کلید هایی که نیاز به مدیریت دارند از الگوریتم های کلید عمومی استفاده میشود.

الگوریتم های کلید عمومی

مفهوم الگوریتم های کلید عمـومی و یـا public-key توسـط Whitfield Diffie و Hellman ارائـه شده که هدف آن ها ارائه یک سیستم رمزنگـاری کـه دو عـدد کلیـد داشـته باشـد یکـی بـرای رمزنگـاری اطلاعات و دیگری برای رمزگزاری به این صورت که این دو از هم جدا باشند که نتوان از یکـی دگـری را بدست آورد. Diffie و Hellman اولین بار الگوریتم های کلید عمومی را در سال 1976 بیـان کردنـد و و مدتی بعد مقاله "New Directions in Cryptography " از آن دو را منتشر شـد، Merkle در سال 1978 روش خود را ارائه داد.

از سال 1976 تا کنون الگوریتم های متفاوتی برای رمزنگاری کلید عمومی ارائه شـده اسـت کـه خیلـی از این الگوریتم هادارای امنیت مناسبی نمیباشند، از این الگوریتم ها تعداد کمی الگورتیم هستند که بسیار کارا اند که هم در سیستم های دیجیتالی و هم در رمزنگاری اطلاعات استفاده میـشوند. ایـن الگـوریتم هـا دارای سرعت پایینی هستند ولی دارای امنیت بسیار خوبی هـستند. سـرعت آنهـا بـه قـادری پـایین اسـت کـه بـرای رمزنگاری دادهای بسیار بزرگ کاریی پایینی دارند البته همان طور که گفته شد می‌توان این الگورتیم ها را با الگوریتم های کلید متقارن ترکیب نمود و الگورتم هایی با کارایی بهتر بدست آورد.

در ادامه این تحقیق به الگوریتم های معروف رمز نگاری که بر پایه نظریه اعداد می باشند، می پردازیم.

الگوریتم کوله پشتی

اولین الگوریتمی که برای رمزنگاری کلید عمومی ارائه شد الگوریتم کوله پشتی بود که توسط Ralph Merkle و Martin Hellman ارائه شد. اگر چه بعدها معلوم شد که این الگوریتم از نظر امنیتی امنیت مطلوب را ندارد ولی درزمان خود الگوریتم کارایی بوده است. به این الگورتیم الگوریتم کل پشتی صفر و یک نیز میگویند. اما استفاده ای که Ralph Merkle و Martin Hellman از الگوریتم کوله پشتی کردند این‌بود که یک پیغام را به وسیله یک تعداد از راه حل های مساله کوله پشتی رمز نگاری کنند.

برای پرداختن به این مساله ابتدا دنباله Superincreasing را تعریف می‌کنیم. دنباله Superincreasing دنبال ای است که هر عدد از مجموع اعداد قبل بزرگتر باشد. مانند دنباله چهار عنصری زیر :

3,6,10,21

ما به این دنباله از وزنهای موجود برای کوله پشتی Superincreasing می گوییم که در واقع کلید اصلی عمل رمز نگاری است. حال به رمز نگاری پیغام زیر توجه کنید. ابتدا پیغام را به صورت دودویی در میاوریم اکنون این داده دودویی داده خام یا داده رمزنگاری نشده ما هستند.

حال این داده خام را به قسمت های مساوی تقسیم می‌کنیم که به هر قسمت یک بلوک میگوییم، سپس یک کوله پشتی Superincreasing که تعداد عناصر آن برابر تعداد عناصر هر بلوک است در نظر میگیریم. عنصر اول از هر بلوک داده خام را به عنصر اول دنباله نسبت میدهیم، عنصر دوم را به عنصر دوم دنباله تا عنصر آخر بلوک. حال ما داده خام را بلوک به بلوک رمزنگاری می‌کنیم یعنی هر بلوک یک عدد رمزنگاری شده دارد مثلا اگر ما عناصر 3 بلوک داشته باشیم داده رمزنگاری شده ما دارای دنباله 3 عدد می باشد. این عدد به این صورت به دست می آیند که عناصر کوله پشتی Superincreasing را باهم جمع می‌کنیم اگر عنصر متناظر آن در داده خام 1 باشد. مثلا اگر طول هر بلوک N عنصر باشد و M داده خامی که در این بلوک می‌باشد و K دنباله کوله پشتی Superincreasing باشد، داده رمزنگاری شده مربوط به این بلوک به صورت زیر بدست می آید.

$$C = M_1.K_1 + M_2.K_2 + \ldots + M_N.K_N$$

اکنون با یک مثال این کار را توضیح میدهیم. اگر

111001010110000000011000

داده خام می باشد که میخواهیم عمل رمزنگاری را روی آن انجام دهیم باشد. ما این داده را به بلوک های شش عنصری تقسیم می‌کنیم. و یک دنباله که به آن کوله پشتی Superincreasing میگوییم و تعداد

عناصر آن نیز شش عدد است در نظر میگیریم، این دنباله کلید رمزنگاری ماست. حال ما عناصر هر بلوک را جداگانه رمزنگاری می‌کنیم. به این صورت که اعداد دنباله کوله پشتی را باهم جمع می‌کنیم اگر عنصر متناظر آن در بلوک مربوطه 1 باشد.

داده خام	111001	010110	000000	011000
دنباله کوله پشتی	1,3,6,13,27,52	1,3,6,13,27,52	1,3,6,13,27,52	1,3,6,13,27,52
داده رمزنگاری شده	1+3+6+52 = 62	3+13+27 = 43	0 = 0	3+6 = 9

و با توجه به جدول 1 داده های رمز نگری شده ما {9 ,0 ,43, 62} می باشد.

حال ما برای رمزگشایی دو مساله کوله پشتی داریم یکی مساله ای با مرتبه زمانی n و دیگری مساله ای ساخت با مرتبه زمانی نمایی.

حالت اول حالتی است که ما اعداد رمزنگاری شده و تعداد عناصر هر بلوک و دنباله کل پشتی Superincreasing را داریم در این حالت ما به راحتی میتوانیم داده را رمزگشایی کنیم مثلا برای بلوک اول جدول 1 ما میخواهیم از عدد 62 به داده خام یعنی به 111001 برشیم برای این کار عدد 62 را با عدد آخر دنباله کوله پشتی یعنی 52 مقایسه می‌کنیم وچون بزرگتر است رقم آخر بلوک 1 است حال 62 را از 52 کم می‌کنیم و حاصل که 10 میباشد با عنصر یکی مانده به‌آخر کل پشتی مقایسه می‌کنیم چون 10 از 27 کوچکتر است رقم یکی مانده به آخر بلوک 0 است. و همین طور تا آخر. مانند شبه کد زیر:

که در آن آرایه K دنباله کوله پشتیSuperincreasing عدد C داده رمزنگاری شده مربوط به بلوک N تعداد عناصر بلوک و M ارایه ای که عناصر داده خام در آن میباشد.

```
for i = N downto 1 do
begin
        if C > K[ i ] do
        begin
                M[ i ] = 1
                C = C - K[ i ]
        end
        else M[ i ] = 0
end
```

و اما حالت دوم که ما در آن داده رمزنگاری شده و تعداد اعضای بلوک را داریم ولی دنباله کوله پشتی Superincreasing را نداریم، که در این حالت برای رسیدن به بلوک رمزنگاری نشده یا بلوک خام راه معمول چک کردن تمام حالت هست که بلوک میتوانند داشته باشند، که مرتبه زمانی آن 2^n می باشد که ممکن است برای بلوک ما یعنی بلوکی با تعداد عناصر 6 مرتبه زمانی آن 2^6 شود یعنی حل شدنی باشد ولی برای بلوک های مثلا 256 عنصری مرتبه آن 2^{256} میباشد که بسیارطولانی و عملا حل نشدنی می باشد.

همان طور که دیدیم الگوریمی که در بالا گفته شد هم برای رمزنگاری و هم برای رمزگشایی یک کلید داشت، یعنی الگوریتم فوق یک الگوریتم کلید متقارن می‌باشد. که برای بالا بردن امنیت ما به یک الگوریتم کلید عمومی نیاز داریم. کلید عمومی کلیدی است که میتوانیم داده را رمزنگاری کنیم ولی رمزگشایی آن سخت باشد و کلید خصوصی همان کلیدی است که میتوانیم به راحتی داده را رمزگشایی کنیم. برای این کار همان دنباله کوله پشتی را کلید خصوصی در نظر میگیریم و کلید عمومی را از آن به دست می آوریم.

بدست آوردن کلید عمومی از کلید خصوصی

همان طور که گفتیم دنباله کوله پشتی کلید خصوصی ماست و ما برای بدست آوردن کلید عمومی از این دنباله استفاده می‌کنیم. برای این کار ما هر عدد از این دنباله را دریک عدد خاص (n) ضرب و سپس باقیمانده را بر عدد دیگری (m) بدست میاوریم. عدد n باید از مجموع عناصر دنباله Superincreasing ما یا همان کلید خصوصی یا دنباله کوله پشتی بزرگتر باشد وعدد m نباید یک فاکتور از n باشد. برای مثال اگر {2,3,6,13,27,52} دنباله کوله پشتی ما باشد، ما عدد 105 را به n در

نظر میگیریم چون از از مجموع عناصر دنباله {2,3,6,13,27,52} بزرگتر می باشد. و m را 31 در نظر میگیریم زیرا یک فاکتور از 105 نیست و به صورت زیر کلید عمومی را بدست میاوریم.

$2 \times 31 \bmod 105 = 62$
$3 \times 31 \bmod 105 = 93$
$6 \times 31 \bmod 105 = 81$
$13 \times 31 \bmod 105 = 88$
$27 \times 31 \bmod 105 = 102$
$52 \times 31 \bmod 105 = 37$

در نتیجه کلید عمومی ما عبارت است از : {62,93,81,88,102} که ما به این دنباله کوله پشتی Normal میگوییم. در واقع ما کوله پشتی Superincreasing را به کوله پشتی Normal تبدیل کردیم، زیرا حل کوله پشتی نرمال بسیار سختتر از حل کوله پشتی Superincreasing میباشد.

رمزنگاری

برای رمزنگاری ابتدا پیغام را به چند بلوک تقسیم می‌کنیم به طوری که تعداد ارقام هر بلوک به اندازهٔ تعداد عناصر یا تعداد وزن های کوله پشتی نرمال باشد. سپس هر بلوک را رمزنگاری می‌کنیم به این صورت که وزن های کوله پشتی رابا هم جمع می‌کنیم اگر رقم متناظر با آن در بلوک خام (رمزنگاری نشده) باشد. یعنی اگر K کوله پشتی باشد و B بلوک (از صفر و یک تشکیل شده) و N تعداد وزنها (تعداد عناصر) کوله پشتی باشد. عدد رمزنگاری شده متناظر با این بلوک به صورت زیر بدست می آید.

$$C = B_1.K_1 + B_2.K_2 + \dots + B_N.K_N$$

و سپس مقادیر بدست آماده را پشت سر هم قرار میدهیم، یعنی اگر M بلوک داشته باشیم داده رمزنگاری شده ما عبارت خواهد بود از :

$$C_1, C_2, \dots, C_M$$

به عنوان مثال برای پیغام 011000110101101110 که به صورت دودویی می باشد داریم:

$Message = 011000\ 110101\ 101110$
011000 corresponds to $93 + 81 = 174$
110101 corresponds to $62 + 93 + 88 + 37 = 280$
101110 corresponds to $62 + 81 + 88 + 102 = 333$

به این ترتیب داده رمزنگاری شده ما خواهد بود: {174,208,333}

رمزگشایی

برای رمزگشایی کردن پیغام همان طور که گفتیم نیاز داریم به کلید خصوصی یعنی به کوله پشتی Superincreasing و همینطور مقادیر n و m که برای محاسبه کردن کوله پشتی normal استفاده میشد. به این ترتیب که ابتدا ما n^{-1} را از فرمول $n(n^{-1}) \equiv 1 (mod\ m)$ بدست اورده و سپس تمامی اعداد رمزنگاری شده را در $n^{-1}\ mod\ m$ ضرب می‌کنیم سپس عدد بدست آماده را به صورت حاصل جمع اعدادی که در کوله پشتی Superincreasing است نوشته و نسبت به جایگاه آنها در دنباله کوله پشتی Superincreasing عدد را بصورت دودویی در میاوریم.

به عنوان مثال {2,3,6,13,27,52} کوله پشتی Superincreasing است و m برابر 105 و n برابر 31 است و داده رمزنگاری شده ما {174,208,333} است در نتیجه n^{-1} برابر خواهد بود با 61 پس داده های رمزنگاری را در 105 mod 61 ضرب می‌کنیم:

$$174 \times 61\ mod\ 105 = 9 = 3 + 6,\ which\ corresponds\ to\ 011000$$

$$280 \times 61\ mod\ 105 = 70 = 2 + 3 + 13 + 52,\ which\ corresponds\ to\ 110101$$

$$333 \times 61\ mod\ 105 = 48 = 2 + 6 + 13 + 27,\ which\ corresponds\ to\ 101110$$

درنتیجه پیغام ما 011000110101101110 است.

پیاده سازی

مثالی که در اینجا ارائه شده حل کردن آن آسان است ولی در پیاده سازی عملی که در واقعیت است کوله پشتی ها حداقل 250 عنصر دارند یعنی طول هر بلوک آنها از 250 رقم بیشتر است. در پیاده سازی واقعی اعدادی که در کوله پشتی هستند از یک توالی از اعداد Random استفاده میشود وقتی ما یک کوله پشتی با چنین حالتی داشته باشیم حتی با راه حل brute force هم نمی‌توان آن را حل کرد به طوری که در حالت brute force اگر با یک کامپوتری که بتواند یک ملیون حالت را در یک ثانیه حساب کند چیزی حدود 10^{46} سال به طول می کشد که بتواند رمز یک پیغام را باز کند. حتی اگر هم یک ملیون کامپوتر به صورت موازی کار کنند تا این مساله را حل کنند حل آن از طول عمر خورشید بیشتر خواهد بود !

الگوریتم RSA

خیلی زود وقتی که Ralph Merkle و Martin Hellman الگوریتم کوله پشتی را ارائه دادند، یک الگوریتم بالغ و کامل که مبتنی بر کلید عمومی بود ارائه شد.که این الگوریتم برای رمزنگاری کردن و امضای دیجیتالی استفاده میشود. چیزی که در RSA قابل لمس است این است که RSA هم در پیاده سازی و هم در فهم بسیار ساده و آسان است. این روش بعد از سال ها هنوز هم مورد استفاده قرارمی گیرد. و همچنین معروف ترین الگوریتم در این زمینه می‌باشد.

RSA از اول نام سازنده های آن یعنی Leonard Adleman,Ron Rivest, Andi Shamir گرفته شده است. کارایی و امنیت RSA بر اساس سختی پیدا کردن فاکتور برای‌اعداد بسیار بزرگ می باشد. حال به بررسی الگوریتم فوق می پردازیم.

p و q را به صورت دو عدد Ranom انتخاب می کنیم و برای بالا بودن امنیت آن بهتر است p و q طول آن ثابت باشد مثلا هر دو عدد اولی هستند که تعداد ارقام آنها 100 می‌باشد. و یک کلید رمزنگاری e انتخاب می کنیم به طوریکه e نسبت به $(p-1)(q-1)$ اول باشد. و عدد n را طبق رابطه زیر حساب مکنیم:

$$n = pq$$

و هم چنین عدد d را طبق رابطه زیر بدست میاوریم.

$$ed \equiv 1 \ (mod(p-1)(q-1))$$

$$\rightarrow d = e^{-1} \ mod((p-1)(q-1))$$

عدد n و e کلید عمومی ما میباشند و عددd کلید خصوصی ما می‌باشد.

و دو عدد pو q که اول استفاده کردیم از بین می بریم چون دیگر استفاده ای ندارند و نباید فاش هم شوند(برای جلوگیری از حمله به سیستم رمزنگاری ما).

و برای رمزنگاری کردن باید پیغامی را که میخواهیم رمزنگاری کنیم به صورت بلوک هایی که طول رقمهای آنها از طول رقمهای n کمتر باشد تقسیم بندی می کنیم. برای مثال اگر pو q هرکدام 100 رقم باشند آنگاه n 200 رقم در نتیجه م اطول بلوک را کمتر از 200 در نظر میگیریم. آنگاه هر بلوک را بصورت زیر رمزنگاری می کنیم. که در آن m_i بلوک i ام و c_i داده رمزنگاری شده بلوک i ام می‌باشد.

$$c_i = m_i^e \ mod \ n$$

و برای رمزگشایی به صورت از رابطه زیر استفاده می‌کنیم :

$$m_i = c_i^d \bmod n$$

خلاصه سیستم رمزنگاری RSA در جدول زیر اورد شده است.

کلید عمومی	
n	pq (p و q دو عدد اول)
e	یک عدد Random که نسبت به $(p-1)(q-1)$ اول است
کلید خصوصی	
d	$e^{-1} \bmod((p-1)(q-1))$
رمزنگاری	
$c_i = m_i^e \bmod n$	
رمزگشایی	
$m_i = c_i^d \bmod n$	

حال به یک مثال کوتاه عددی توجه کنید.

اگر $p = 47$ و $q = 71$ آنگاه :

$$n = pq = 3337$$

برای پیدا کلید رمزنگاری e حاصل عبارت $(p-1)(q-1)$ را بدست میاوریم:

$$(p-1)(q-1) = 46 \times 70 = 3220$$

عدد 79 را بطور Random برای e انتخاب می‌کنیم که نسبت به 3220 اول است.

و d را به صورت زیر بدست میاوریم

$$d = 79^{-1} \bmod 3220 = 1019$$

حال e و n را به صورت کلید عمومی منتشر می‌کنیم و d را برای کلید خصوصی نگاه میداریم. و p و q را از بین میبریم.

اکنون برای رمزنگاری کردن پیغام

$$m = 6882326879666683$$

ابتدا آنرا به بلوک های کوچکتر تقسیم می‌کنیم به طوری که تعداد ارقام آن از 4 کمتر باشد. (زیرا n 4 رقمی است.) :

$m_1 = 688$

$m_2 = 232$
$m_3 = 687$
$m_4 = 966$
$m_5 = 668$
$m_6 = 003$

بلوک اول را بصورت زیر رمزنگاری می‌کنیم.

$688^{79} \bmod 3337 = 1570 = c_1$

برای بقیه بلوک ها نیز به همین صورت عمل می‌کنیم به طوری که داده رمزنگاری شده ما به صورت زیر خواهد بود.

$c=1570275620912276242 3158$

و همچنین بلوک اول داده رمزنگاری شده را به صورت زیر رمزگشایی میشود.

$15701019 \bmod 3337 = 688 = m_1$

RSA در سخت افزار

RSA در سیستم های مختلف سخت افزاری پیاده سازی شده که در جدول زیر قسمتی از آن نشان داده شده است.

Existing RSA Chips

Company	Clock Speed	Baud Rate Per 512 Bits	Clock Cycles Per 512 Bit Encryption	Technology	Bits per Chip	Number of Transistors
Alpha Techn.	25 MHz	13 K	.98 M	2 micron	1024	180,000
AT&T	15 MHz	19 K	.4 M	1.5 micron	298	100,000
British Telecom	10 MHz	5.1 K	1 M	2.5 micron	256	—
Business Sim. Ltd.	5 MHz	3.8 K	.67 M	Gate Array	32	—
Calmos Syst. Inc.	20 MHz	28 K	.36 M	2 micron	593	95,000
CNET	25 MHz	5.3 K	2.3 M	1 micron	1024	100,000
Cryptech	14 MHz	17 K	.4 M	Gate Array	120	33,000
Cylink	30 MHz	6.8 K	1.2 M	1.5 micron	1024	150,000
GEC Marconi	25 MHz	10.2 K	.67 M	1.4 micron	512	160,000
Pijnenburg	25 MHz	50 K	.256 M	1 micron	1024	400,000
Sandia	8 MHz	10 K	.4 M	2 micron	272	86,000
Siemens	5 MHz	8.5 K	.3 M	1 micron	512	60,000

سرعت RSA

در سخت افزار سرعت RSA هزار برابر کمتر از روش DES است. و در پیاده سازی نرم افزاری هم RSA صد برابر سرعتی کمتر از DES دارد. اما آنچه که می‌توان گفت این است که سرعت RSA هیچ وقت به سرعت الگوریتم های متقارن نمیرسد. جدول زیر سرعت های مختلف RSA را نشان میدهد.

RSA Speeds for Different Modulus Lengths with an 8-bit Public Key (on a SPARC II)

	512 bits	768 bits	1,024 bits
Encrypt	0.03 sec	0.05 sec	0.08 sec
Decrypt	0.16 sec	0.48 sec	0.93 sec
Sign	0.16 sec	0.52 sec	0.97 sec
Verify	0.02 sec	0.07 sec	0.08 sec

افزایش سرعت نرم افزاری

اگر مقدار e را به سرعت هوشمندانه و خوب انتخاب شود رمزنگاری RSA سریعتر خواهد شد. سه حالت معمول که برای e وجود دارند عبارت اند از 3 ،17 و 65537 . و هیچ مشکل امنیتی با انتخاب این مقدار های e وجود ندارد. و مزیت آن این است که اگر هر کدام از این اعداد را به سرعت دودویی تبدیل کنیم میبینیم که هر کدام فقط دو عدد یک دارند. و در قسمت کلید خصوصی هم می‌توان سرعت را بالا تر ببریم به طوری که اعداد p و q را از بین نرود. اگر این اعداد را نگاه داری کنیم باز هم سرعت الگوریتم بالاتر میرود.

امنیت

امنیت در RSA به مساله پیدا کردن فاکتور برای اعداد بسیار بزرگ (n) بستگی دارد . در کل می‌توان گفت امنیت RSA به پیدا کردن دو عدد اول بستگی دارد. مثلا پیدا کردن دو عدد اول 100 رقمی. از لحاظ ریاضی هنوز ثابت نشده که بتوان هر عدد اول را پیدا کرد.

یک راه حمله به RSA حدس زدن $(p-1)(q-1)$ می‌باشد که این حمله هم به سختی پیدا کردن فاکتور می‌باشد.

البته باز هم تغییراتی می‌توان در RSA داد که باز هم فاکتور پیدا کردن از n باز هم سختتر میشود. تهدید دیگری که امروزه وجود دارد این است که اعداد حدود 130 رقم فاکتور گیری میشوند، ولی به راحتی می‌توان n را خیلی بزرگ تراز این مقدار قرار داد. و در آخرمیتوان گفت امروزه هنوز راهی برای شکستن این سیستم رمزنگاری پیدا نشده است.

یک حمله معمول بر RSA

یک حالت ممکن در پیاده سازی RSA این است که همه یک n ثابت میگیرند ولی مقادیر متفاوتی از e و d میگیرند. البته‌این حالت مشکل ایجاد نمیکند، مشکل وقتی ایجاد میشود یه یک متن دو بار رمزنگاری شود با e های مختلف که این e ها نسبت به هم اول میباشند. در این صورت می‌توان متن را بدون نیاز به کلید خصوصی رمزگشایی کرد.

حال اگر m پیغام اصلی باشد و e_1 و e_2 کلید های رمزنگاری باشند که e_1 و e_2 نسبت به هم اول هستند. و همینطور یک n داریم بنابراین دو پیغام رمزنگاری شده داریم:

$$c_1 = m^{e1} \bmod n$$
$$c_2 = m^{e2} \bmod n$$

از e_1 و e_2 که نسبت به هم اول هستند و از الگوریتم extended Euclidean میتوانیم r و s را پیدا کنیم بطوری که :

$$re_1 + se_2 = 1$$

r بدست آماده از بالا منفی می‌باشد. حال میتوانیم پیغام را طبق رابطه زیر رمزگشایی کنیم:

$$(c_1^{-1})^{-r} \times c_2^{s} = m \bmod n$$

حمله دیگر بر RSA وجود دراد یکی از نظریه احتمال برای پیدا کردن فاکتور های n استفاده می کند. و نتیجه این که وقتی تعداد زیادی از کاربران از این سیستم رمزنگاری استفاده میکنند n آنها نباید یکسان باشد.

الگوریتم POHLIG-HELLMAN

الگوریتم POHLIG-HELLMAN خیلی شبیه RSA می‌باشد. و این الگوریتم یک الگـوریتم متقـارن نیست زیرا کلید های متفاوتی برای رمزنگاری و رمزگشایی آن وجود دارد. و همچنین مـی‌تـوان گفـت کـه این الگوریتم یک الگوریتم کلید عمومی هم نیست زیرا به راحتی کلید ها از هم بدست میاینـد یعنـی کلیـد رمزنگاری براحتی از کلید رمزگشایی بدست می آید و بلعکس.

مانند RSA :

$$C = P^e \bmod n$$

$$P = C^d \bmod n$$

و همینطور:

$ed \equiv 1$ (یک عدد پیچیده)

و برخلاف RSA n از ضرب دو عدد اول بزرگ بدست نمی آید. بلکه n باقیمانده یـک قـسمت از یـک کلید رمزی می‌باشد. حال اگر کسی e و n را داشته باشد مـی توانـد d را حـساب کنـد ولـی اگـر e و n را نداشته باشد باید معادله زیر راحل کند :

$$e = \log_P C \bmod n$$

که حل این معادله بسیار سخت است.

الگوریتم RABIN

الگوریتم RABIN از سختی بدست آوردن یک ریشهٔ دوم از یک عدد مرکب استفاده می‌کند. و این کار به سختی پیدا کردن فاکتور در RSA میباشد.

ابتدا دو عدد p و q که هردو اول هستند انتخاب می‌کنیم، این دو عدد کلید خصوصی هستند، سپس عدد n را طبق $n = pq$ بدست میاوریم که n کلید عمومی ما میباشد.

برای رمزنگاری کردن M (که M باید کوچکتر از n باشد.) به صورت زیر عمل می‌کنیم.

$$C = M^2 \bmod n$$

و رمزگشایی آن هم ساده ولی تعداد معادلات آن کمی زیاد است. به این صورت که ابتدا 4 تا عدد m_1 تا m_4 را طبق معادلات بدست اورده

$$m_1 = C^{(p+1)/4} \bmod p$$
$$m_2 = (p - C^{(p+1)/4}) \bmod p$$
$$m_3 = C^{(q+1)/4} \bmod q$$
$$m_4 = (q - C^{(q+1)/4}) \bmod q$$

و سپس دو عدد a و b را طبق روابط زیر بدست می آیند.

$$a = q(q^{-1} \bmod p)$$
$$b = p(p^{-1} \bmod q)$$

آنگاه :

$$M1 = (am1 + bm3) \bmod n$$
$$M2 = (am1 + bm4) \bmod n$$
$$M3 = (am2 + bm3) \bmod n$$
$$M4 = (am2 + bm4) \bmod n$$

یکی از اعداد $M1$ تا $M4$ درست میباشند. مثلا اگر M یک متن انگلیسی باشد به راحتی قابل تشخیص است و اگر هم M یک پیغام عددی هم باشد میتوانیم یک مقدار برای چک کردن آخر آن اضافه کنیم.

منابع

1.APPLIED CRYPTOGRAPHY, Second Edition, By Bruce Schneier

2. Applied Cryptography, by A. Menezes, P. van Oorschot, and S. Vanstone, CRC Press, 1996

3. CRYPTOGRAPHY AND NETWORK SECURITY:Principles and Practice, Second Edition, William Stallings

4. ورودی به نظریه اعداد، حمید رضا امیری،انتشارات مدرسه،1378

5.طراحی الگوریتم ها، Rechard Neapolitan، ترجمه جعفر نژادقمی، انتشارات علوم رایانه،1383