امنیت شبکه

جلسه سوم: رمزنگاری نامتقارن و تعیین اعتبار یا احراز هویت پیام

تهیه و تنظیم: دکتر آرش حبیبی لشکری

منبع: کتاب اصول و مبانی امنیت شبکه (استانداردها و کاربردها)

اولین نسخه: دی 1393
بروزرسانی: دی 1393

فهرست:

- تعاریف
- رمزنگاری نامتقارن و کاربردها
  - الگوریتم RSA
  - الگوریتم دیفی - هلمن
- امضاء الکترونیکی
- تعیین اعتبار یا احراز هویت پیام
- توابع درهم‌سازی یکطرفه

تعاریف اصلی

واژه های پایه در رمزگذاری نامتقارن

متن اصلی: این بخش پیام یا داده قابل خواندن است که بعنوان ورودی به الگوریتم داده می‌شود.

الگوریتم رمزنگاری: الگوریتم رمزنگاری تغییرات مختلفی را روی متن اصلی اعمال می‌نماید.

کلید عمومی و خصوصی: این یک جفت کلید انتخاب شده‌اند که اگر یکی برای رمزگذاری استفاده شود دیگری برای رمزگشایی استفاده خواهد شد. تغییرات واقعی اعمال شده توسط الگوریتم رمزگذاری در اصل وابسته به کلیدهای عمومی و خصوصی است که بعنوان ورودی انتخاب شده‌اند.

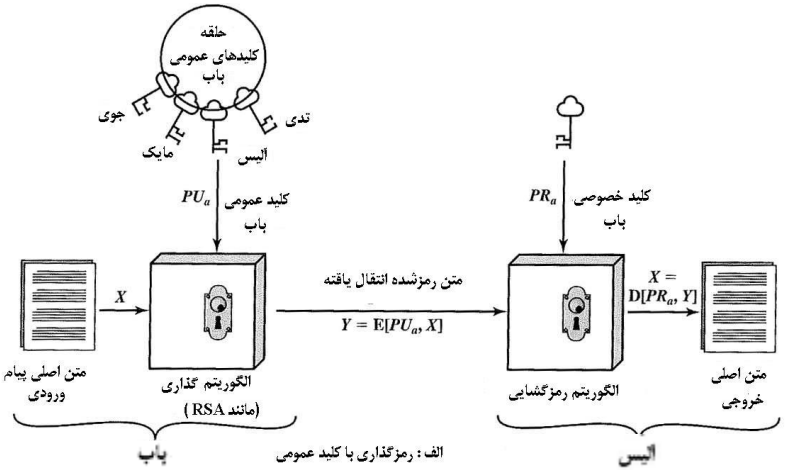
متن رمزشده: این بخش در واقع پیام بهم ریخته بعنوان خروجی است که وابسته به متن اصلی و کلیدهاست. برای یک پیام داده شده دو کلید مختلف دو خروجی مختلف را تولید خواهند نمود.

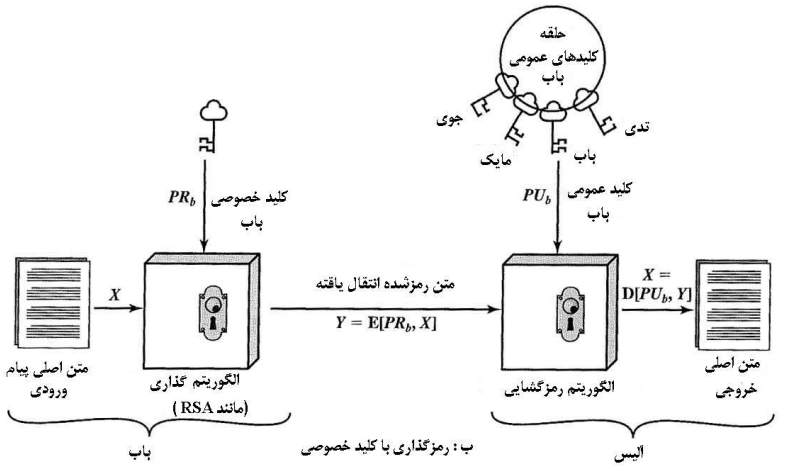
الگوریتم رمزگشایی: این الگوریتم متن رمزشده و کلید مرتبط را گرفته و متن اصلی را تولید می‌نماید.

یک الگوریتم رمزنگاری کلید - عمومی همه منظوره بر پایه یک کلید برای رمزگذاری و کلیدی متفاوت اما مرتبط برای رمزگشایی بنا نهاده شده است.

کلید مورد استفاده در رمزگذاری‌های متقارن بنام کلید محرمانه معروف است. دو کلید مورد استفاده در رمزگذاری‌های عمومی بنامهای کلید عمومی و کلید خصوصی معروف هستند که همواره، کلید خصوصی محرمانه نگه داشته شده ولی برای جلوگیری از اشتباه با کلید محرمانه رمزگذاری متقارن با نام کلید خصوصی شناخته می‌شود.

مدل ساده شده رمزنگاری متقارن





گامهای اساسی رمزنگاری نامتقارن

1. هر کاربر یک جفت کلید تولید نموده که برای رمزگذاری و رمزگشایی پیام بکار می‌روند.

2. هرکاربر یکی از دو کلید را در یک ثبات عمومی یا فایل قابل دسترس قرار می‌دهد. این کلید عمومی است. کلید بصورت محرمانه نگهداری می‌شود. همانطور که تصویر 3قبلی بخش الف پیشنهاد می‌نماید، هر کاربر یک مجموعه از کلیدهای عمومی تولید شده توسط سایر کاربران را نگهداری می‌نماید.

3. اگر باب بخواهد یک پیام خصوصی را برای آلیس ارسال نماید، باب پیام را با کلید عمومی آلیس رمزگذاری خواهد نمود.

4. وقتی آلیس پیام را دریافت می‌نماید، او پیام را با کلید خصوصی خود رمزگشایی می‌نماید. هیچ گیرنده دیگری نمی‌تواند پیام را رمزگشایی نماید چرا که تنها آلیس کلید خصوصی آلیس را می‌داند.

با این خطمشی، همه شرکت‌کنندگان به کلید عمومی دسترسی دارند درحالی که کلید خصوصی توسط هر کاربر بطور خصوصی تولید شده و لذا هیچ وقت نیاز به پخش شدن نخواهد داشت.

کاربردهای رمزنگاری کلید - عمومی

- **رمزگذاری/ رمزگشایی:** فرستنده یک پیام را با کلید عمومی گیرنده رمزگذاری می‌نماید.

- **امضای الکترونیکی:** فرستنده یک پیام را با کلید خصوصی "امضا می‌نماید". امضاء نمودن با اعمال یک الگوریتم رمزنگاری بر روی یک پیام یا یک بلوک کوچک از داده که یک تابع از پیام است، انجام می‌گردد.

- **تبادل کلید:** دو طرف با یکدیگر برای تبادل یک کلید جلسه همکاری می‌کنند.

چهار الگوریتم رایج در رمزنگاری نامتقارن و کاربرد آنها

نام الگوریتم	کاربردهای رمزنگاری نامتقارن		
	رمزگذاری / رمزگشایی	امضاء الکترونیکی	**تبادل کلید**
RSA	بله	بله	بله
دیفی - هلمن	خیر	خیر	بله
DSS	خیر	بله	خیر
ECC	بله	بله	بله

شرایطی که یک الگوریتم نامتقارن باید برآورده سازد

. این محاسبات برای شخص b جهت تولید یک جفت کلید (کلید عمومی PUB، کلید خصوصی PRB) آسان است.

2. از نظر محاسباتی برای فرستنده A تولید متن رمزشده، با دانستن کلید عمومی و متن پیام اصلی که باید رمزگذاری شود یعنی M، آسان است: $C = E(PU_b, M)$

3. از نظر محاسباتی برای گیرنده B رمزگشایی متن رمزشده به دست آمده با استفاده از کلید خصوصی برای بازیابی پیام اصلی آسان است : $M = D(PR_b, C) = D[PR_b, E(PU_b, M)]$

4. از نظر محاسباتی برای مهاجم، دانستن کلید عمومی PUB، برای تعیین کلید خصوصی PRB، غیر ممکن است.

5. از نظر محاسباتی برای مهاجم، دانستن کلید عمومی PUB، و متن رمزی C، برای بازیابی پیام اصلی M، غیر ممکن است.

6. (ضروری نیست) هر کدام از این دو کلید می‌توانند برای رمزنگاری استفاده شده و کلید دیگر نیز برای رمزگشایی بکار رود: $M = D[PU_b, E(PR_b, M)] = D[PR_b, E(PU_b, M)]$

الگوریتم رمزنگاری کلید عمومی RSA

الگوریتم RSA یک روش رمزنگاری بلوکی است که در آن متن اصلی و متن رمزشده اعداد صحیحی بین 0 و n از میان تعداد n موجودیت هستند.

رمزگذاری و رمزگشایی برای متن اصلی M و متن رمزشده C بصورت زیر خواهد بود:

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

$$M = C^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n$$

هم فرستنده و هم گیرنده باید مقادیر n و e را بدانند و تنها گیرنده مقدار d را خواهد دانست.

این یک الگوریتم رمزنگاری کلید- عمومی با کلید - عمومی : $KU = \{e, n\}$

و کلید خصوصی $KR = \{d, n\}$ است.

رضایتمندی RSA

برای آنکه این الگوریتم برای رمزنگاری کلید- عمومی رضایت بخش باشد، باید نیازمندیهای زیر برآورده شود:

1. این امکان برای پیدا کردن مقادیر e، d، n وجود دارد به طوری که برای همه چنین باشد:

$$M^{ed} \bmod n = M \langle M < n \rangle$$

2. نسبتا محاسبه M^e و C^d به ازاء تمام مقادیر M<N آسان باشد.
3. تعیین d به ازاء e و n غیرممکن است.

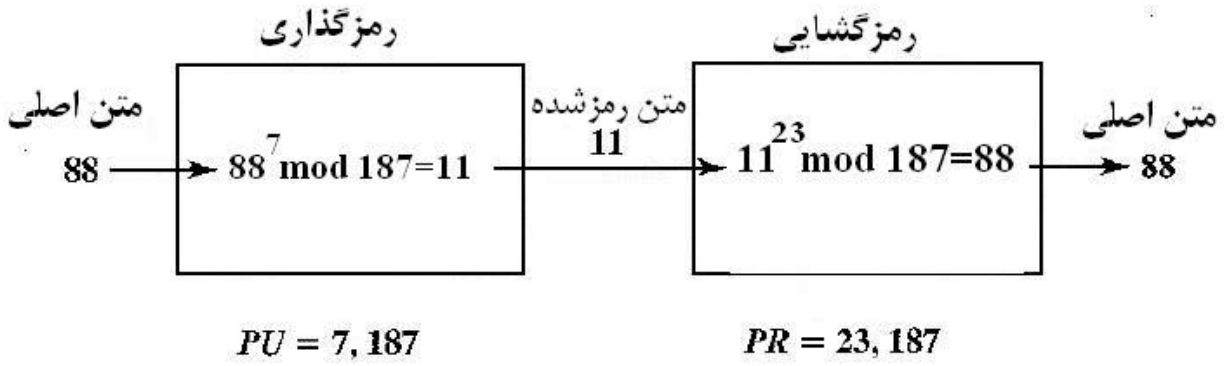
نکته: دو پیش نیاز اول به راحتی قابل برآورده شدن هستند. شرط سوم نیز برای مقادیر بزرگ e و n قابل برآورده نمودن است.

الگوریتم RSA

تولید کلید	
انتخاب p و q	P و q هردو اعداد اول هستند و نابرابرند
محاسبه n=p*q	
محاسبه	$\varphi(n)=(p-1)(q-1)$
مقدار صحیح e را انتخاب می‌کنیم	$\gcd(\varphi(n),e)=1:1<e<\varphi(n)$
محاسبه d	$de \bmod \varphi(n)=1$
کلید عمومی	$KU=\{e,n\}$
کلید خصوصی	$KR=\{d,n\}$
رمزگذاری	
متن اصلی پیام	M<N
متن رمز شده	$C=M^e(\bmod n)$
رمزگشایی	
متن رمز شده	C
متن پیام اصلی	$M=C^d(\bmod n)$

مثال برای RSA

فرض کنید که کاربر A کلید عمومی خود را منتشر کرده است و کاربر B می‌خواهد پیام M را به A ارسال نماید. سپس B مقدار $C = M^e (\mathrm{mod} n)$ را محاسبه نموده و C را منتقل می‌نماید. در بخش دریافت این متن رمزشده، کاربر A با محاسبه $M = C^d (\mathrm{mod} n)$ رمزگشایی می‌کند.



مثال برای RSA

1. دو عدد اول p = 17 و q = 11 را انتخاب نمایید.
2. حال n = pq = 17* 11 = 187 را محاسبه نمایید.
3. مقدار $\varphi(n) = (p-1)(q-1) = 16*10 = 160$ را محاسبه نمایید.
4. حال e را بگونه ای انتخاب کنید که e نسبت به $\varphi(n) = 160$ اول باشد و از $\varphi(n)$ کمتر باشد؛ فرض کنیم که e=7 را انتخاب کنیم.
5. مقدار d را بگونه‌ای تعیین نمایید که $de \bmod 160 = 1$ و d<1 باشد. از آنجاییکه

$(160+1) = 161 = 23*7$ لذا مقدار صحیح d = 23 خواهد بود.

کلیدهای حاصل برابر کلید عمومی PU={7,187} و PR={187,23} خواهند بود. مثال صفحه بعد استفاده از این کلیدها را برای ورودی متن اصلی M = 88 را نشان می‌دهد.

رمزنگاری و رمزگشایی مثال

برای رمزگذاری، نیاز به محاسبه $C = 88^7 \bmod 187$ داریم. با بهره‌گیری از خواص محاسبات پیمانه‌ای، می‌توانیم این کار را به صورت زیر انجام دهیم:

$$88^7 \bmod 187 = [(88^4 \bmod 187) * (88^2 \bmod 187) * (88^1 \bmod 187)] \bmod 187$$

$$88^1 \bmod 187 = 88$$

$$88^2 \bmod 187 = 7744 \bmod 187 = 77$$

$$88^4 \bmod 187 = 59969536 \bmod 187 = 132$$

$$88^7 \bmod 187 = (88 * 77 * 132) \bmod 187 = 897432 \bmod 187 = 11$$

برای رمزگشایی، $M = 11^{23} \bmod 187$ را محاسبه می‌کنیم:

$$11^{23} \bmod 187 = [(11^4 \bmod 187) * (11^2 \bmod 187) * (11^1 \bmod 187) * (11^{16} \bmod 87] \bmod 187$$

$$11^1 \bmod 187 = 11$$

$$11^2 \bmod 187 = 121$$

$$11^4 \bmod 187 = 14641 \bmod 187 = 55$$

$$11^8 \bmod 187 * 11^8 \bmod 187 = 214358881 \bmod 187 * 214358881 \bmod 187 = 33 * 33$$

$$11^{23} \bmod 187 = (11 * 121 * 55 * 33 * 33) \bmod 187$$

$$= 79720245 \bmod 187 = 88$$

امنیت RSA

دو روش ممکن برای شکست الگوریتم RSA وجود دارد:

اولین روش حمله جستجوس فراگیر است که تمام کلیدهای خصوصی ممکن را امتحان می-کند. بنابراین، هر چه تعداد بیت‌های e و d بیشتر باشد آنگاه امنیت الگوریتم بیشتر خواهد شد. با این حال، به دلیل اینکه محاسبات (هم در تولید کلید و هم در رمزگذاری/ رمزگشایی) پیچیده هستند، هرچه اندازه کلید بزرگتر باشد، سیستم کندتر اجرا خواهد شد.

دومین روش بر روی فاکتورگیری n به دو عامل اول آن متمرکز شده است. برای n بزرگ با فاکتورهای اول بزرگ، فاکتورگیری یک مشکل بزرگی است اما نه به آن سختی که باید باشد.

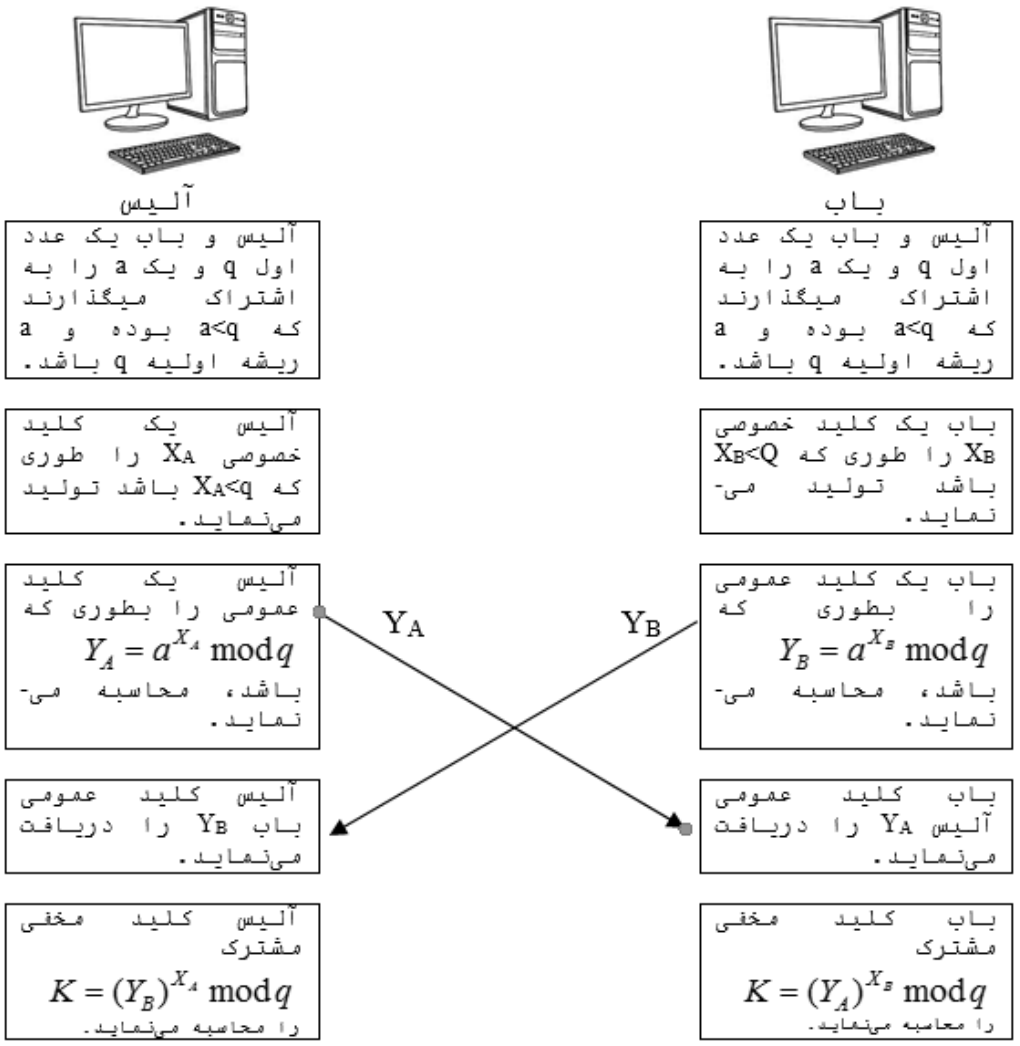
الگوریتم تبادل کلید دیفی - هلمن

هدف از این الگوریتم این است که دو کاربر قادر به تبادل کلید مخفی به شکلی امن باشند بطوری که در آینده بتوانند از آن برای رمزگذاریهای بعدی پیامها استفاده نمایند. الگوریتم برای موثر واقع شدن به دشواری و سختی محاسبه لگاریتم گسسته بستگی دارد.

اگر فرض کنیم که دو طرف بنامهای کاربر A و B میخواهند تبادل کلید نمایند، در این روش کاربر A مقدار کلید را با استفاده از $K = (Y_B)^{X_A} \bmod q$ محاسبه نموده و کاربر B نیز مقدار کلید را به صورت $K = (Y_A)^{X_B} \bmod q$ محاسبه می‌نماید (تصویر صفحه بعد). هر دوی این محاسبات منجر به تولید نتایج یکسانی خواهند شد:

$$
\begin{aligned}
K &= (Y_B)^{X_A} \bmod q \\
&= (a^{X_B} \bmod q)^{X_A} \bmod q \\
&= (a^{X_A})^{X_A} \bmod q \\
&= a^{X_B X_A} \bmod q \\
&= (a^{X_A})^{X_B} \bmod q \\
&= (a^{X_A} \bmod q)^{X_B} \bmod q \\
&= (Y_A)^{X_B} \bmod q
\end{aligned}
$$

الگوریتم تبادل کلید دیفی - هلمن



امنیت الگوریتم تبادل کلید دیفی - هلمن

افزون براین، به دلیل آنکه X_A و X_B خصوصی هستند، یک فرد مهاجم بایستی بتواند فکری برای مواردی چون Y_B، Y_A، a، q نیز بکند. در نتیجه یک فرد مهاجم مجبور است که برای تعیین کلید یک لگاریتم گسسته را محاسبه نماید. برای مثال، جهت تعیین کلید خصوصی کاربر B، یک فرد مهاجم بایستی مقدار $X_B = d \log_{a,q}(Y_B)$ را محاسبه نماید.

بعد از این فرد مهاجم می‌تواند مقدار کلید K را به همان روشی که کاربر B انجام می‌دهد، محاسبه نماید.

امنیت الگوریتم مبادله کلید دیفی - هلمن در این حقیقت نهفته است که با وجودی که محاسبه مقدارهای همنهشت با توان‌های یک عدد به پیمانه یک عدد اول کار نسبتاً ساده‌ای است، اما محاسبه لگاریتم‌های گسسته کار بسیار سختی خواهد بود.

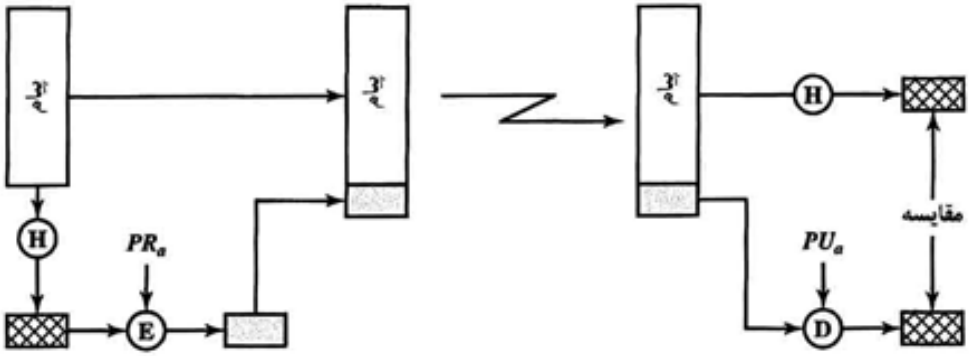
می‌توان چنین بیان نمود که برای اعداد اول بزرگ، عملیات دوم تقریباً غیرممکن است.

امضاء الکترونیکی

امضاء الکترونیکی

در گذشته برای تصدیق یک پیام از لحاظ منبع و تمامیت داده‌ها کل پیام را رمزگذاری نمودند. اگر چه هر دو اعتبار نویسنده و محتویات تایید می‌شد، اما این امر نیازمند سربار زیادی بود. چرا که:

- هر سند باید به شکل متن ساده نگه‌داری می‌شد تا برای اهداف کاربردی مختلف مورد استفاده قرار گیرد.
- همچنین یک کپی از آن نیز باید به شکل متن رمزشده ذخیره می‌شد به طوری که منشا و محتویات را بتوان در صورت اختلاف تایید نمود.



یکی از راههای کارآمدتر دستیابی به نتایج مشابه به رمز درآوردن یک بلوک کوچک ازبیتها است که یک تابع از این سند خواهد بود که به آن تایید کننده اعتبار میگویند و اگر این تایید کننده با کلید خصوصی فرستنده رمزگذاری شود به آن اصطلاحاً امضاء الکترونیکی میگویند (تصویر روبرو).

احراز هویت

احراز هویت؟

در اصل رمزنگاری در برابر حملات غیرفعال همانند استراق سمع مقاومت دارد در حالی که برای مقاومت برابر حملات فعال پیش نیازهای متفاوتی وجود دارد (تحریف داده‌ها و تراکنش‌ها) و در علم امنیت شبکه حمایت در مقابل چنین حملاتی به اصطلاح با عنوان احراز هویت پیام شناخته می‌شود.

برای احراز هویت پیام نیازی به رمزگذاری کل پیام نیست بلکه در اغلب موارد تنها یک برچسب احراز هویت تولید شده و به هر پیام برای انتقال افزوده می‌گردد. خود پیام رمز نشده و می‌تواند در مقصد مستقل از تابع احراز هویت مقصد، خوانده شده و مورد استفاده قرار گیرد.

نکته: طبق تعریق محرمانگی پیام، چون خود پیام رمزگذاری نمی‌شودف لذا محرمانگی پیام فراهم نخواهد شد. ولی ممکن است که احراز هویت و محرمانگی هر دو در یک الگوریتم از طریق رمزنگاری یک پیام بعلاوه برچسب احراز هویتش ترکیب شده و فراهم گردد.

احراز هویت پیام جدا از رمزنگاری پیام است.

چرا احراز هویت بدون محرمانگی؟

1. برنامه‌های مختلفی وجود دارند که یک پیام مشابه را به چند مقصد مختلف بصورت همگانی پخش می‌کنند. مثال ساده از این گروه برنامه‌ها میتوان به پیامهای «هشدار و اطلاع رسانی به کاربران یک شبکه که درحال حاظر شبکه غیرقابل دسترس می‌باشد» اشاره نمود.

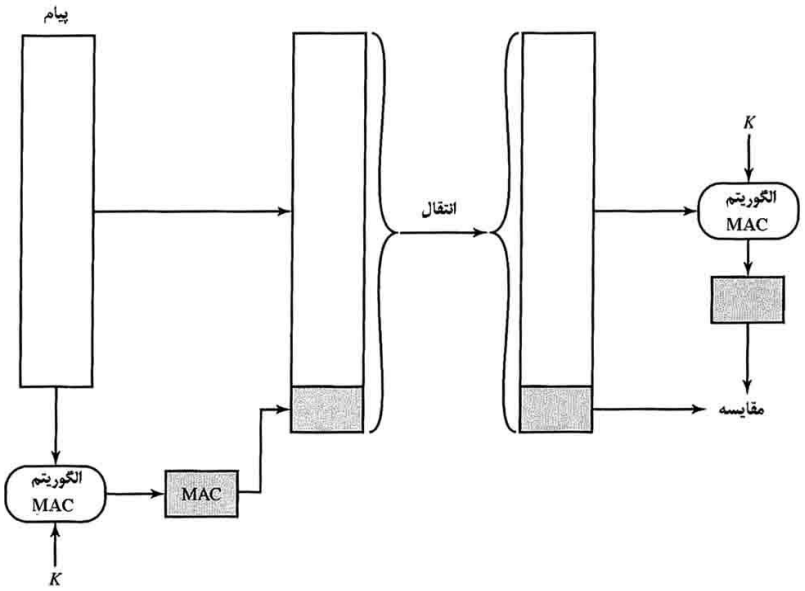
2. در برخی مبادلات با توجه به بار اطلاعاتی سنگینی موجود زمان لازم برای رمزگشایی همه پیام‌های ورودی فراهم نخواهد بود. در این حالت احراز هویت به طور گزینشی در مورد پیام‌های که به طور تصادفی برای چک‌کردن انتخاب شده اند، انجام خواهد شد.

3. احراز هویت یک برنامه کامپیوتری در شکل متن ساده در واقع سرویس بسیار خوبی خواهد بود، چرا که برنامه کامپیوتری می‌تواند در هر زمان بدون رمزگشایی اجرا گردد که رمزگشایی آن باعث هدررفتن منابع سیستم خواهد شد.

کد احراز هویت پیام

در اصل کد احراز هویت استفاده از یک کلید مخفی برای تولید یک بلوک کوچک از داده است، که با عنوان MAC شناخته می‌شود و به پیام اضافه می‌گردد.

این تکنیک فرض می‌کند که در دو طرف ارتباط، مثلا A وB، یک کلید مخفی رایج KAB را به اشتراک می‌گذارند. هنگامی که A یک پیام را به B می‌فرستد، کد احراز هویت پیام را به عنوان تابعی از پیام و کلید $MAC_M = f(K_{AB}, M)$ محاسبه می‌کند.



پیام به همراه کد به گیرنده ارسال می‌شود. گیرنده محاسبات مشابه‌ای را روی پیام رسیده با کلید مخفی مشابه، برای تولید کد جدید احراز هویت پیام اجرا می‌نماید.

کد تولید شده با کد محاسبه شده مقایسه می‌گردد (تصویر روبرو).

پیشنهاد NIST استفاده از الگوریتم DES برای این بخش است.

فواید کد احراز هویت پیام

1. گیرنده اطمینان حاصل خواهد نمود که پیام مخدوش نشده است. اگر یک مهاجم پیام را تغیردهد اما کد را تغییر ندهد، آنگاه کد محاسبه شده گیرنده با کد دریافت شده متفاوت خواهد بود.

2. گیرنده اطمینان دارد که پیام از فرستنده منتسب آمده است. از آنجایی که شخص دیگری کلید مخفی را نمی‌داند، هیچ فرد یگری نمی‌تواند پیام با کد مناسب را فراهم کند.

3. اگر پیام شماره توالی را دربرداشته باشد، آنگاه گیرنده می‌تواند از توالی مناسب اطمینان حاصل نماید، زیرا یک مهاجم نمی‌تواند به طور موفقیت آمیز شماره توالی را تغییر دهد.

توابع درهم‌سازی یکطرفه

تابع یکطرفه درهم‌سازی

یک جایگزین مناسب برای کد احراز هویت پیام

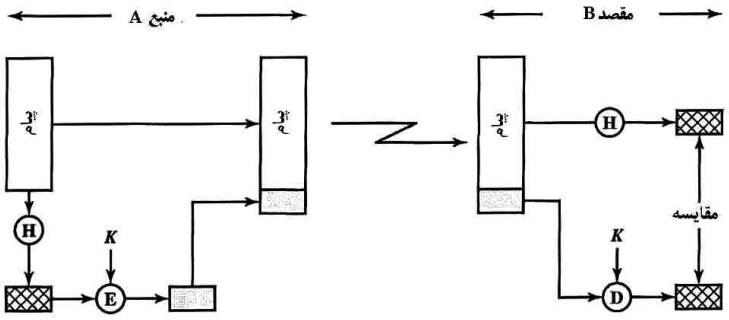
همانند کد احراز هویت پیام، یک تابع درهم‌سازی یک پیام با سایز متغییر M را به عنوان ورودی می‌پذیرد و یک پیام با طول - ثابت خلاصه $H(M)$ را به عنوان خروجی تولید می‌کند.

ولی برخلاف MAC، یک تابع درهم‌سازی کلید مخفی را به عنوان ورودی نمی‌پذیرد.

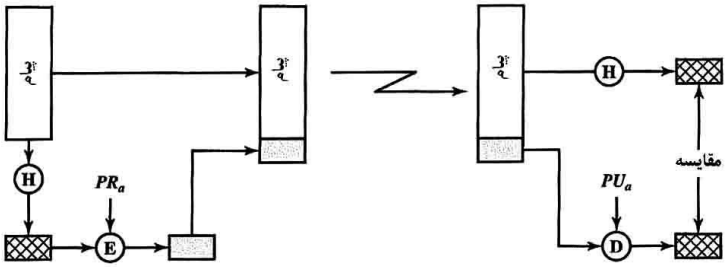
برای احراز هویت یک پیام، خلاصه پیام به همراه پیام به طریقی که خلاصه پیام احراز هویت شود، فرستاده می‌شود.

تصویر صفحه بعد سه روش معمول تابع یکطرفه درهم سازی با نشان میدهد.

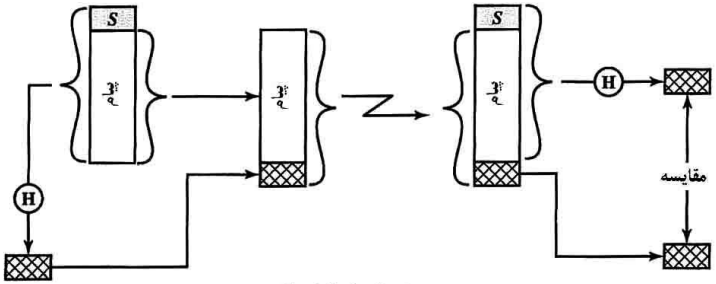
تابع یکطرفه درهم‌سازی



الف : استفاده از رمزگذاری متعارف



ب : استفاده از رمزگذاری کلید – عمومی



پ : استفاده از مقدار مخفی

توابع درهم‌سازی امن

تابع درهم‌سازی امن

تابع درهم‌سازی یکطرفه، یا همان تابع درهم‌سازی امن، نه تنها در احرازهویت پیام نقش مهم دارد بلکه در امضاهای الکترونیکی نیز نقش بسزایی دارد. هدف از تابع درهم‌سازی ایجاد یک" اثر انگشت" از فایل، پیام، یا سایر انواع بلوکهای داده است.

نیازمندیهای تابع درهم‌سازی

۱. تابع H می‌تواند بر روی هر بلاک داده با هر سایزی اعمال شود.
۲. تابع H یک خروجی با طول ثابت را تولید می‌نماید.
۳. محاسبه H(x) برای هر xبسیار آسان است و باعث می‌شود که پیاده‌سازی سخت افزار و نرم-افزار عملی گردد.
4. برای هر کد h داده شده، این امر غیرعملی است که x ای را پیدا کنیم که معادله H(x)=h درست باشد. یک تابع درهم‌سازی با این مشخصات تحت عنوان یکطرفه یا مقاومت پیش تصویر شناخته می‌شود. (خاصیتی یک طرفه است)
5. به ازای هر بلاک x داده شده، این امر غیرعملی است که بتوان$Y \neq x$ با مقدار درهم‌سازی یکسان H(y)=H(x) وجود داشته باشد.
6. این امر تقریبا غیر عملی است تا جفت(x,y) طوری پیدا شود که به ازای آن H(x)=H(y) باشد.

یک تابع درهم‌سازی ساده

تمام توابع درهم سازی از اصول کلی زیر استفاده می‌کنند:

- ورودی (پیام، فایل، و غیره) به عنوان دنباله‌ای از بلوک‌های n- بیتی
- بعنوان ورودی هر زمان یک بلوک برای تولید یک تابع درهم‌سازی n- بیتی پردازش می‌شود

یکی از ساده‌ترین توابع درهم سازی بیتی همان یای انحصاری (XOR) بیت - به - بیت است:

$$C_i = b_{i1} \oplus b_{i2} \oplus ... \oplus b_{im}$$

که در آن:

C_i = امین بیت از کد درهم سازی $1 \le i \le n$

m = تعداد بلوک‌های n - بیتی در ورودی

b_{ij} = بیت iام در بلوک jام

$\oplus$ = عملگر XOR

بیت n	...	بیت 2	بیت 1	
B_{n1}		B_{21}	B_{11}	بلوک 1
B_{n2}		B_{22}	B_{12}	بلوک 2
.	.	.	.	
.	.	.	.	
.	.	.	.	
B_{nm}		B_{2m}	B_{1m}	بلوک m
C_n		C_2	C_1	کد درهم‌سازی

سوالات مرتبط

1. سه خط مشی برای احراز هویت یک پیام را نام ببرید؟
2. منظور از کد احراز هویت پیام چیست؟
3. عناصر اصلی یک سیستم رمزنگاری کلید - عمومی کدامند؟
4. سه کاربرد سیستم رمزنگاری کلید - عمومی را نام برده و به اختصار توضیح دهید؟
5. فرق بین یک کلید خصوصی و کلید مخفی چیست؟
6. امضاء الکترونیکی چیست؟
7. تابع درهم سازی چیست؟

خلاصه:

رمزنگاری نامتقارن و کاربردها، الگوریتم RSA، الگوریتم دیفی - هلمن، امضاء الکترونیکی، تعیین اعتبار یا احراز هویت پیام، توابع درهم‌سازی

جلسه بعدی:
توزیع کلید و احراز هویت کاربر

منبع: کتاب اصول و مبانی امنیت شبکه (استانداردها و کاربردها)

ترجمه: دکتر آرش حبیبی لشکری، مهندس نسرین بدیع، مهندس فرنار توحیدی

هیچ راهی برای به دست آوردن تجربه به جز از
طریق تجربه وجود ندارد.