فصل نهم

امنیّت (security)

امنیّت در سیستم‌های توزیع شده به دو بخش اصلی تقسیم می‌گردد:

۱. ارتباط بین کاربران و فرآیندها که احتمالاً در ماشین‌های مختلف قرار دارند. که راهکار اصلی آن ایجاد کانال امن است.

۲. مجوز که تضمین می‌کند فرآیند تنها مجوز دستیابی به منابعی را کسب می‌کند که برای آن در نظر گرفته شده است. راهکار اصلی آن کنترل دسترسی است. در این فصل در مورد این مطالب و جزئیات دیگری در زمینه امنیت بحث خواهد شد.

۱.۹ نکاتی کلی در مورد امنیت

در این بخش به صورت کلّی به مفاهیم، سیاستها و راهکارهای امنیتی خواهیم پرداخت.

۱.۱.۹ سیاستها و راهکارهای امنیتی

اگر بخواهیم به سیستم کامپیوتری اطمینان داشته باشیم باید دو اصل محرمانگی و جامعیت در نظر گرفته شوند.

محرمانگی: خاصیتی که باعث می‌شود، داده‌ها در سیستم تنها برای افراد مجاز قابل دیدن باشند.

جامعیت: تغییرات در منابع سیستم (اعم از سخت‌افزار، نرم‌افزار و داده‌ها) بایستی تنها به روش مجاز قابل انجام باشد. ضمناً تغییرات نامناسب باید قابل تشخیص و ترمیم باشند.

تهدیدهای امنیتی: هر سیستم کامپیوتری ممکن است با چهار تهدید امنیتی روبه‌رو شود:

۱. رهگیری یا دستبرد (Interception): زمانی که کاربری نامعتبر که هویت آن احراز نشده است به سرویس یا داده‌ها دستیابی پیدا کند. مانند جایی که ارتباط بین دو طرف توسط یک شخص ثالث شنود شود یا داده‌ها به صورت نامعتبر کپی شوند.

۲. وقفه (Interruption): وقفه به معنی وضعیتی است که در آن سرویسها یا داده‌ها غیرقابل دسترس، بلااستفاده یا تخریب گردند. از نظر امنیتی اگر کسی سعی کند که سرویسی را برای کاربران غیرقابل دسترس کند یک تهدید امنیتی وقفه رخ داده است. به این حالت حمله عدم پذیرش سرویس (انکار سرویس/ service of denial) نیز گفته می‌شود.

۳. تغییر (Modification): هرگونه تغییر غیرمجاز در داده‌ها یا دستکاری سرویس‌ها از این گونه تهدید امنیتی است. مثل ایجاد تغییر در داده‌های بانک اطلّاعاتی به صورت غیرمجاز.

۴. جعل (Fabrication): زمانیکه به صورت غیرمجاز داده‌ها یا فعالیتهای دیگری تولید شود که به صورت عادی وجود ندارند. ممکن است یک نفوذگر (Intruder) تلاش کند که به بانک اطلاعاتی کلمه عبور داده اضافه کند.

از بین چهار تهدید امنیتی به جز اوّلی بقیه در داده‌ها به نحوی تحریف (falsification) ایجاد می‌کنند. برای مقابله با تهدیدهای امنیتی، ابتدا باید سیاستهای امنیتی تعریف نمود، سیاست امنیتی مشخص می‌کند که عناصر درون سیستم مجاز هستند که فعالیتهایی را انجام دهند و چه اعمالی را نباید انجام دهند. منظور از عناصر یا نهادهای سیستم، کاربران، سرویسها، داده‌ها، ماشینها و نظایر آنها است. پس از وضع سیاستهای امنیتی یکسری راهکارهای امنیتی به وجود می‌آید که در مقابل نقض سیاستهای امنیتی توسط افراد غیرمجاز عمل نماید.

مهمترین راهکارهای امنیتی عبارتند از:

۱. رمزنگاری (Cryptography): رمزنگاری که اساس امنیت کامپیوتر است، داده‌ها را طوری تغییر می‌دهد که نفوذگر نتواند معنی آن را بفهمد. رمزنگاری محرمانگی داده را پیاده‌سازی می‌کند و همچنین می‌توان به وسیله آن متوجه شد که آیا داده‌ها به صورت غیرمجاز تغییر یافته‌اند یا خیر.

۲. احراز هویت (Authentication): هویت ادعا شده توسط عناصری مانند: کاربران، سرویس‌گیرندگان و سرویس‌دهندگان را بررسی می‌کند. مهمترین روش احراز هویت استفاده از کلمه عبور (password) است. امّا این تنها راه احراز هویت نیست.

۳. مجوز (Authorization): مجوّز معمولاً بعد از احراز هویت عمل می‌کند. هرکاربری که در سیستمی احراز هویت می‌شود، لزوماً به همه منابع سیستم مجوّز دسترسی ندارد. این راهکار بررسی می‌کند که برنامه کاربردی یا هر عنصر احراز هویت شده دیگر تنها به قسمتهای مجاز دسترسی پیدا نمایند.

۴. حسابرسی (Auditing): این قسمت مقابله با تهدیدهای امنیتی به صورت جلوگیری از عملی نیست بلکه در این بخش ردیابی و ثبت می‌شود که کدام کاربر یا سرویس‌گیرنده به چه عناصری و به چه روشهایی دسترسی پیدا کرده است. با تحلیل داده‌های ثبت شده می‌توان رخنه‌های امنیتی و نفوذگرانی را که به سیستم حمله کرده‌اند شناخت و به همین دلیل نفوذگران تلاش می‌کند تا اثری در سیستم از خود باقی نگذارند.

۹. ۱. ۲ نکاتی در مورد طراحی

در طراحی سیستم امنیتی باید سه نکته اصلی مدنظر قرار گیرد: تأکید بر کنترل، لایه‌بندی راهکارهای امنیتی و سهولت.

تأکید بر کنترل: هنگام طراحی سیستم حفاظتی از برنامه‌های کاربردی می‌توان از سه روش پیروی کرد و در هر روش، تمرکز و تأکید کنترل بر بخشی خاص قرارگرفته است.

روش اوّل: تمرکز مستقیم بر روی حفاظت از داده‌های مربوط به برنامه کاربردی است. در اینجا مسأله اصلی جامعیت است و هر داده‌ای که تغییر می‌کند، به صورت خودکار بررسی می‌شود که آیا این تغییر مجاز بوده یا خیر.

روش دوم: تمرکز بر حفاظت در هنگام دسترسی به یک منبع خاص است. در این‌جا تأکید روی راهکارهای کنترل است. به این معنی که در هر دسترسی، مشخص می‌شود که چه اعمالی فراخوانی شده و چه کسانی آنها را فراخوانی می‌کنند. مثلاً در یک سیستم مبتنی بر شیء مشخص شود کدام سرویس‌گیرندگان مجاز به فراخوانی کدام متدها هستند.

روش سوم: مستقیماً روی کاربران تأکید دارد. در این حالت تنها افراد خاصی صرف نظر از عملیاتی که می‌خواهند انجام دهند، به برنامه‌کاربردی دسترسی دارند. برای مثال در سیستم اطلاعاتی یک دانشگاه برخی داده‌ها تنها توسط اعضای هیأت علمی یا کارکنان مجاز قابل دسترسی است . دانشجویان نمی‌توانند به آن دسترسی داشته باشند. در اینجا تأکید کنترل بر نقش کاربران است.

لایه‌بندی راهکارهای امنیتی: همانطور که قبلاً نیز گفته شده است، در یک سیستم توزیع شده برای برقراری ارتباط از یک سیستم لایه‌ای استفاده می‌شود. معمولاً لایه‌ها به دو سطح بالا و پایین تقسیم می‌شوند. در سطح پایین، معمولاً لایه‌هایی معادل لایه‌های فیزیکی، پیوند داده‌ها، شبکه و انتقال قرار دارند. به علاوه سخت‌افزار و هستهٔ سیستم عامل نیز به عنوان عناصر سطح پایین سیستم شناخته می‌شوند. در سطح بالا، سرویسهای سیستم عامل میان افزار و برنامه‌های کاربردی (لایه کاربرد) قرارگرفته است.

بین امنیت سیستم و اعتماد به سیستم تفاوتی مهم وجود دارد. سیستم یا اَمن است یا نیست امّا اینکه آیا کاربر سیستم را اَمن می‌داند یا خیر، مسأله اعتماد است. این که راهکارهای امنیتی در کدام لایه قرار گیرند، به اعتماد کاربر در ایجاد امنیت در لایه‌ای خاص بستگی دارد.

به عنوان مثال یکی از روشهای ارتباطی در شبکه‌های گسترده استفاده از سرویس دادهٔ چند مگابایتی سویچی (SMDS)[1] است. این سرویس روشی برای اتصال چند شبکه محلّی به هم و ایجاد یک شبکه گسترده است. در اینجا هر شبکه محلّی یک مسیریاب داد که از طریق آن به زیر شبکه ارتباطی SMDS متّصل می‌شود. یکی از راهکارهای ایجاد امنیت این است که در هر مسیریاب، ابزار رمزگذاری قرار داده شود و بسته‌هایی که بین شبکه‌های محلّی تبادل می‌شود، رمزگذاری و رمزگشایی گردد. این روش بین میزبانهای یک شبکه محلّی امنیت را برقرار نمی‌کند.

[1]. Switching Megabit Data Service

حال اگر یک کاربر به امنیت سیستم اعتماد نداشته باشد، می‌توان از یک سرویس امنیتی دیگر استفاده کند. یکی از سرویسهای پرکاربرد که در لایه انتقال کار می‌کند، سرویس لایه سوکت‌های امن (SSL)[1] است. این سرویس می‌تواند داده‌ها را به صورت امن از طریق اتصال TCP ارسال نماید. اعتماد کاربرد به سیستم به میزان اعتماد آن به SSL بستگی دارد.

در سیستم‌های توزیع شده برای ایجاد اعتماد بیشتر، راهکارهای امنیتی معمولاً به لایه‌های بالاتر به ویژه میان افزار اعمال می‌شود. سرویسهای امنیتی گفته شده در صورتی قابل اعتماد هستند که سرویسهایی که به آنها متکی‌اند، به اندازه کافی اَمن باشند. به مجموعه‌ای از راهکارهای امنیتی که باید سیاست امنیتی را اعمال کنند، اصطلاحاً پایه محاسبه قابل اعتماد (TCB)[2] گفته می‌شود. هر چه TCB بر محدوده کوچکتری از سیستم، اِعمال شود، بهتر است. برای مثال در یک سیستم توزیع شده کلاسیک که میان افزار روی سیستم عامل شبکه‌ای موجود پیاده‌سازی شده است، امنیت آن می‌تواند به سیستم عامل محلّی بستگی داشته باشد. در این‌حالت TCB شامل سیستم عامل‌های محلّی در میزبان‌های مختلف هستند. به عبارت دیگر ـ در سیستمهای توزیع شده مبتنی بر نرم‌افزار، بایستی به سیستم‌عامل محلّی اعتماد وجود داشته باشد. در صورتیکه این اعتماد وجود نداشته باشد، بایستی بخشهای مربوط در خود سیستم توزیع شده گنجانده شود. در سیستم‌عامل‌های ریزهسته (MicroKernel) که هسته سیستم عامل کوچک و با اعمال بنیادی می‌باشد و اکثر اعمال توسط یک سری فرآیند سطح کاربر صورت می‌پذیرد، انجام این تغییرات در سیستم عامل ساده‌تر است. برای مثال اگر اعتماد به امنیت سیستم فایل وجود نداشته باشد، فرآیندهای مربوط می‌تواند به راحتی با فرآیندهایی جابه‌جا کرد که، سیستم فایلی مجهز به ابزار امنیتی لازم را داشته باشند.

با این روش می‌توان سرویسهای مختلف سیستم را برحسب امنیت مورد نیاز، در ماشین‌های مختلف توزیع کرد. به این ترتیب می‌توان سرویس‌هایی که نیاز به امنیت بالا دارند (مانند سرویس‌دهندگان فایل) را در تعداد محدودی ماشین جمع نمود و روی آنها سرویسهای لازم برای مقابله با تهدیدهای امنیتی را قرار داد. این تفکیک، TCB را به تعداد کمی ماشین محدود می‌کند و اعتماد کلی در امنیت سیستم توزیع شده افزایش می‌یابد. روشی با عنوان واسطه‌های تقلیل یافته برای اجزاء امن سیستم (RISSC)[3] وجود دارد که در سیستمهای توزیع شده این چنین پیاده سازی می‌شود. این واسط در سرویس‌دهنده‌های امنیتی قرار می‌گیرد و سرویس‌گیرنده‌ها تنها از طریق این واسط‌ها می‌توانند به سرویس دهنده‌های اَمن دسترسی داشته باشند.

سهولت: لازم است، راهکارهایی که برای پیاده‌سازی پروتکل‌های امنیتی به کار می‌رود، نسبتاً ساده و درک آنها آسان باشد. این مسأله در اعتمادی که کاربران به برنامه‌های کاربردی خواهند داشت، مهم است همچنین طراحان راحت‌تر متقاعد می‌شوند که روزنه‌های امنیتی وجود ندارد. البته این سادگی نباید خود اصل امنیت را تحت الشعاع قرار دهد و باعث عدم حصول کافی امنیت شود. در مواردی مانند سیستمهای پرداخت الکترونیکی ممکن است خیلی از کاربران علاقه‌مند باشند که از امنیت سیستم، اعتماد حاصل نمایند.

۹.۱.۳ رمزنگاری

فرض کنید داده اولیه ما که به آن متن ساده (plaintext) می‌گوییم «p» نام داشته باشد. این داده تحت تابعی مانند E و پارامتری به نام K که به آن کلید می‌گوییم، به داده‌ای غیرقابل فهم به عنوان متن رمزی (ciphertext) یا «c» تبدیل می‌گردد. ما این تبدیل را به صورت $C = E_K(p)$ نمایش می‌دهیم. این امر در مبدأ انجام می‌شود و متن غیرقابل فهم C ارسال می‌گردد در مقصد تابع رمزگشا (D) عمل عکس را انجام می‌دهد $p = D_K(C)$ و متن اصلی بازیابی می‌شود. نکته مهم کلید (K) است که مبدأ و مقصد از آن اطلاع دارند. بنابراین اگر نفوذگری در بین مسیر، داده را استراق سمع کند، بدون داشتن کلید K، تقریباً غیرممکن است که p را به دست آورد. همچنین نفوذگر نمی‌تواند داده را تغییر دهد. داده رمز شده در صورت تغییر در مقصد به یک داده نامفهوم تبدیل می‌شود و مقصد متوجه می‌شود که متن رمزی تغییر کرده است. تنها راه نفوذگر این است که متن رمزی را رمزگشایی کند، تغییر دهد و مجدداً رمز نماید که بدون داشتن کلید رمز این امر تقریباً غیرممکن است. رمز، زمینه جعل

---

1. Secure Sockets Layer
2. Trusted Computing Base
3. Reduced Interfaces for Secure system components

داده‌ها را نیز از بین می‌برد، نفوذگر نمی‌تواند پیامی را درون سیستم درج کند و آن را به جای مبدأ خاصی به مقصد بفرستد زیرا پیامها باید رمز شده باشد و نفوذگر بدون کلید نمی‌تواند رمزنگاری انجام دهد. معمولاً دو روش مرسوم برای استفاده از کلید رمز وجود دارد:

ـ **سیستم رمزنگاری متقارن:** در آن برای رمزنگاری و رمزگشایی از یک کلید مشترک استفاده می‌شود.

ـ **سیستم رمزنگاری غیرمتقارن:** در آن برای رمزنگاری و رمزگشایی از دو کلید متفاوت استفاده می‌شود.

در سیستم نامتقارن معمولاً هر گره از سیستم مانند A دو کلید دارد. کلید عمومی (K_A^+) و کلید خصوصی (K_A^-) کلید خصوصی همواره نزد A است ولی کلید عمومی به دیگران اعلام می‌شود. فرض کنید گره دیگری به نام B می‌خواهد بسته‌ای را به A بفرستد، بسته را با K_A^+ رمزنگاری می‌کند و A پس از دریافت آن را با K_A^- رمزگشایی می‌نماید دقت کنید که داده با هر کدام از این کلیدها رمزنگاری شود با کلید دیگر رمزگشایی می‌گردد. بنابراین نفوذگران بدون داشتن K_A^- نمی‌توانند، رمزگشایی نمایند. حال فرض کنید B می‌خواهد مطمئن شود که بسته‌ای که فکر می‌کند از A دریافت کرده، واقعاً از طرف A باشد. A بسته را با کلید خصوصی خودش رمز می‌کند و به B می‌فرستد B آن را با کلید عمومی A باز می‌کند. اگر نفوذگری به جای A بسته فرستاده باشد، B پس از رمزگشایی با داده‌های نامفهوم روبه‌رو خواهد شد. ترکیب دو روش بالا روشی بسیارقوی در رمزنگاری را تولید خواهد نمود.

از نظر ریاضی توابع رمزنگاری و توابع درهم سازی (Hashing) به هم شبیه هستند. تابع درهم‌سازی H پیام m به طول دلخواه را می‌گیرد و رشته بیتی h را تولید می‌کند $h = H(m)$، h طول ثابتی دارد. تابع درهم‌سازی مشابه توابع تشخیص خطا مانند (CRC) در بسته‌های ارسالی در شبکه‌های کامپیوتری است.

توابع درهم‌سازی که در رمزنگاری مدنظر هستند، دارای چند خاصیت می‌باشند. نخست اینکه یک طرفه (one-way) هستند، یعنی از h نمی‌توان m را محاسبه کرد. دوم اینکه دارای خاصیت مقاومت برخورد ضعیف (Resistance Collision Weak) هستند، یعنی از نظر تئوری غیرممکن است که بتوان دو ورودی متفاوت m و m' پیدا کرد که $H(m) = H(m)'$ باشد. سوم خاصیت مقاومت برخورد قوی (Resistance Collision Strong) است که حتّی با دانستن تابع H نیز نتوان دو مقدار مختلف m و m' پیدا کرد که $H(m) = H(m')$ باشد.

توابع رمز دارای خواص مشابه هستند. برای تابع E اگر p و C معلوم باشد نباید بتوان کلید K را محاسبه کرد و همچنین نباید بتواند $K' \neq K$ پیدا کرد که $E_k(p) = E_{k'}(p)$ باشد.

۲.۹ کانال امن (Secure Channel)

در ارتباطات سرویس‌گیرنده ـ سرویس دهنده، ساخت سیستمهای امن ناشی از دو نکته است. نخست چگونگی ارتباط امن بین سرویس‌گیرنده و سرویس‌دهنده و دوم اینکه وقتی سرویس‌دهنده درخواستی را از سرویس گیرنده پذیرفت، چگونه متوجه می‌شود که آیا آن سرویس‌گیرنده مجاز به انجام آن درخواست است یا خیر؟

این مسائل با تشکیل کانال امن حل می‌شود. کانال امن فرستنده‌ها و گیرنده‌ها را در مقابل رهگیری، تغییر و جعل حفاظت می‌کنند (لزوماً در برابر وقفه حفاظت نمی‌کنند). در کانال امن به دلیل وجود محرمانگی، احراز هویت متقابل و جامعیت پیام در برابر تهدیدهای امنیتی گفته شده مقاومت می‌شود.

۱.۲.۹ احراز هویت

احراز هویت به معنای اطمینان از هویت کسی است که با شما ارتباط برقرار می‌کند. پس از احراز هویت، جامعیت نیز ضروری است. به این معنی که بسته‌های ارسالی توسط فرد احراز هویت شده آیا حتماً توسط خودش ارسال می‌گردد؟

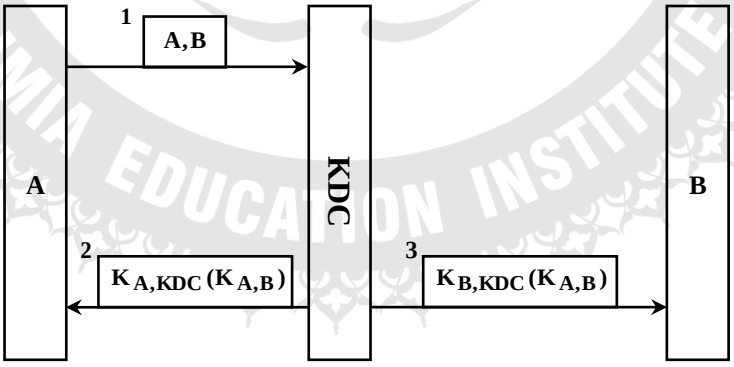
برای این منظور یکی از روشها، استفاده از رمزنگاری به وسیله یک کلید مشترک سرّی به نام کلید نشست (session key) است. چگونگی تبادل این کلید سرّی در آینده بحث خواهد شد. در اینجا فرض کنید A و B می‌خواهند با هم ارتباط برقرار کنند. آنها بین خود کلید سرّی مشترک $K_{A,B}$ را دارند. برای تشکیل یک نشست، ابتدا A مشخصات هویتی خود را به B ارسال می‌کند. سپس B یک پیام به نام R_B را که به چالش R_B

معروف است به A باز می‌گرداند. به این دلیل به R_B، چالش گفته می‌شود که با این پیام B، A را به چالش می‌طلبد. حال A در پاسخ باید R_B را با $K_{A,B}$ رمز کند و به صورت یک پیام به B بازگرداند. B با $K_{A,B}$ پیام را رمزگشایی می‌کند و با تولید مجدد R_B یعنی همان پیام خودش مطمئن می‌شود که A در طرف دیگر وجود دارد. حال B، R_A را با $K_{A,B}$ رمز می‌کند و به A می‌فرستد. A با رمزگشایی پیام از هویت B مطمئن می‌شود. مشکل این روش تعداد زیاد پیامها برای برقراری نشست است. در یک بهینه‌سازی تعداد پیامها از ۵ پیام به ۳ پیام کاهش یافته است. به این ترتیب که در پیام اول مشخصات A و R_A فرستاده می‌شود، در پاسخ $K_{A,B}(R_A), R_B, B$ را می‌فرستد و A در پیام سوم $K_{A,B}(R_B)$ را باز می‌گرداند. این روش ظاهراً ساده‌تر از روش اول است، امّا نفوذپذیر می‌باشد. یکی از روشهای حمله، موسوم به حمله انعکاسی (Reflection Attack) است. در این روش نفوذگری مانند C که $K_{A,B}$ را ندارد تلاش می‌کند که خود را به جای A به B بقبولاند. ابتدا A C و R_C را به B می‌فرستد، B نیز R_B و $K_{A,B}(R_C)$ را باز می‌گرداند. واضح است که C نمی‌تواند $K_{A,B}(R_B)$ را تولید کند. امّا یک نشست جدید را شروع می‌کند و اینبار A و R_B را می‌فرستد. B نیز در پاسخ $K_{A,B}(R_B)$ را تولید می‌کند و به C ارسال می‌دارد. حال C در پاسخ نشست اول $K_{A,B}(R_B)$ را باز می‌گرداند و B فکر می‌کند که پیام از A رسیده است.

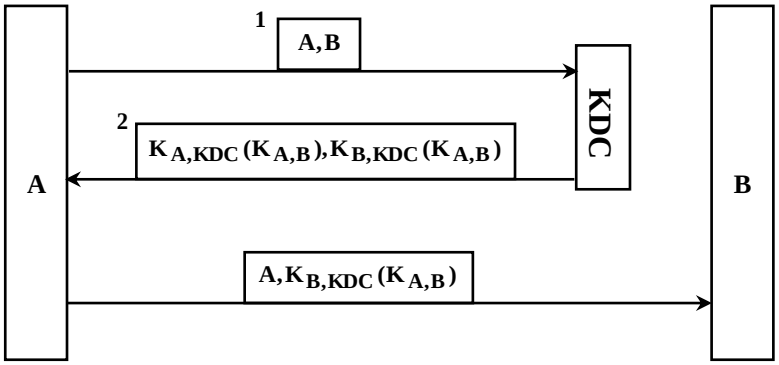
همانطور که دیده شد در روش بهینه شده این تکنیک که به پروتکل چالش ـ پاسخ (Response – challenge) معروف است، راه نفوذ وجود دارد که در روش نخست این راه نیست. یک راهکار استفاده از ساختار متفاوت پیامهای چالش R_A و R_B است مثلاً A تنها با اعداد فرد و B با اعداد زوج کار کند. امّا هنوز هم روشهای حمله دیگری وجود دارد.

مرکز تولید کلید (Key Distributing Center)

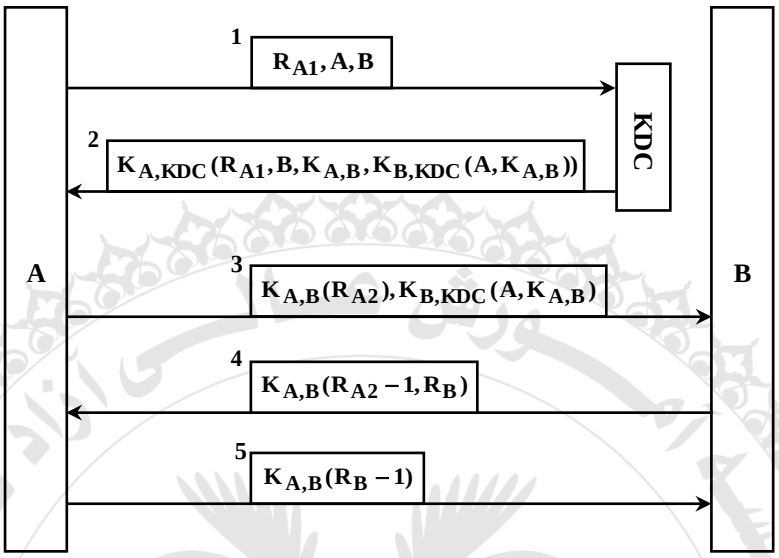
اگر به روش بالا بخواهیم بین N میزبان نشست داشته باشیم به صورتی که هر میزبان بتواند با $N-1$ میزبان دیگر نشست انجام دهد به $\frac{N(N-1)}{2}$ کلید نیاز است که تولید و رد و بدل کردن این همه کلید بسیار مشکل می‌باشد. در عوض ایجاد یک عضو قابل اعتماد در شبکه این میزبانها به عنوان مرکز تولید کلید (KDC) مفید است. بین KDC و N میزبان تنها N کلید نشست وجود خواهد داشت حال اگر A بخواهد با B ارتباط برقرار کند، از KDC یک $K_{A,B}$ دریافت می‌کند. البته KDC کلید جدید یعنی $K_{A,B}$ را به A و B به صورت رمز شده با رمزهای نشست خودشان می‌فرستد.



در اینجا KDL باید با B نیز مستقیماً تماس برقرار کند. در روشی دیگر می‌توان ارتباط KDC با B را حذف کرد.



در اینجا $K_{B,KDC}(K_{A,B})$ به عنوان بلیت (Ticket) تلقی می‌شود که به وسیله آن A می‌تواند با B ارتباط برقرار کند. ترکیب روش فوق و روش چالش ـ پاسخ به یک پروتکل پیشرفته به نام پروتکل احراز هویت نیدهام ـ شرودر منتهی می‌شود که در شکل بعد خواهید دید:

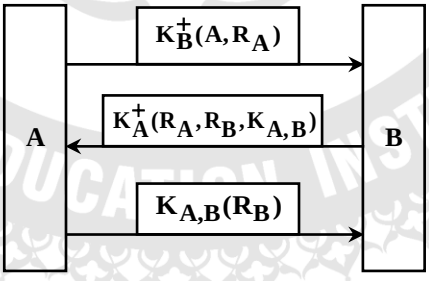


R_{A1} یک عدد تصادفی است که به نانس (Nonce) نیز معروف است. نانس در سیستم غیرتکراری است با قرار دادن R_{A_1} در پیام 1 و 2 مشخص می‌گردد که پیام 2 پاسخی به پیام 1 است و نفوذگران نمی‌توانند از پیامهای قدیمی به صورت جعلی استفاده کنند.

همچنین تفریق عدد یک از R_{A2} و R_B بیشتر مشخص می‌کند که این چالشها در طرف مقابل رمزگشایی شده‌اند و احتمال استفاده نفوذگران را کم می‌نماید.

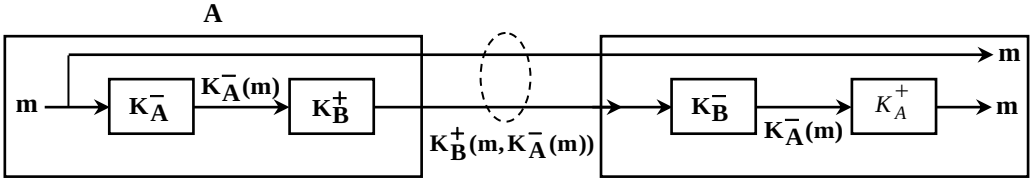
احراز هویت با استفاده از کلید عمومی

وجود KDC که یک عنصر مرکزی است، مشکلات خود را دارد. در اینجا به کمک استفاده از کلید عمومی و خصوصی می‌توان بدون نیاز به KDC، کلید نشست را رد و بدل کرد و در عین حال احراز هویت نیز نمود.



۹. ۲. ۲ محرمانگی و جامعیت

دیدیم که به کمک رمزنگاری می‌توان محرمانگی را به دست آورد. امّا مسأله جامعیت کمی پیچیده‌تر است. یکی از روشهای جامعیت پیام، امضای دیجیتال است. امضای دیجیتال نوعی پیوند جدانشدنی با پیام اصلی است، به طوری‌که پیام از تغییر به دور باشد. مرسوم‌ترین روش امضای دیجیتالی، رمزکردن پیام با کلید خصوصی است.



اگر دو مقدار m در نهایت، یکسان باشند، امضای دیجیتالی صحیح است.

البته این روش تا وقتی عمل می‌کند که کلید خصوصی A ربوده نشده باشد یا A ادعا نکند که کلیدش ربوده شده، همچنین مرسوم است که در فواصل زمانی کلید خصوصی و عمومی A تغییر کند. برای اجتناب از مشکل امضا با کلید خصوصی قدیمی، از مهر زمان در هنگام امضای دیجیتال استفاده می‌شود.

مشکل دیگر طولانی بودن پروسه رمزگذاری کل پیام m توسط کلید خصوصی A است به جای اینکار تنها می‌توان از تعداد بیتهای محدود نتیجه تابع درهم‌سازی $(h = H(m))$ برای امضا استفاده نمود.

دیدیم که حتی با وجود کلیدهای اصلی مثل کلید خصوصی و عمومی با هم در هر نشست یک کلید نشست موقّتی بین طرفین در نظر گرفته می‌شود، علّت این است که اصولاً کلیدهایی که در مدت طولانی استفاده شوند، احتمال بیشتری برای شکسته شدن دارند چون نفوذگران زمان بیشتری دارند. پس استفاده از کلیدهای نشست موقتی و کوتاه‌مدت مفید است. همچنین ممکن است درصد اعتماد طرفین کم باشد و کلیدهای دراز مدت و گران قیمت خود را تنها برای ارتباطات خاصی با افراد خاصی نگاه دارند.

۹. ۲. ۳ ارتباطات گروهی اَمن

در بسیاری از موارد بایستی بین یک گروه ارتباط اَمن برقرار باشد. برای مثال در سرویس‌دهنده‌های تکرار، باید ارتباطات بین همه کپی‌ها در مقابل تهدیدهای امنیتی حفظ شوند.

ارتباط گروهی محرمانه ممکن است از طریق یک کلید سرّی مشترک بین تمام اعضاء حفظ شود. البته حفاظت از این کلید خیلی سخت‌تر از حالتی است که کلید تنها بین دو گره بود. بنابراین، این روش آسیب‌پذیر است.

راهکار دیگر این است که هر دو عضو گروه یک کلید سرّی مجزا داشته باشند. این راه نیاز به $\frac{N(N-1)}{2}$ کلید دارد که دشوار است. راهکار بهتر استفاده از کلید عمومی و خصوصی است. هر عضو کلید خصوصی خود را دارد و کلید عمومیش را بقیه اعضاء می‌دانند.

در سرویس‌دهنده‌های تکرار از یک روش به نام اشتراک سرّ (Secret Sharing) استفاده شده است و ایده‌آل این است که فرآیندها در یک سرّ مشترک هستند. امّا هیچ‌کدام کلّ سرّ را نمی‌دانند. سرّ وقتی نمایان می‌شود که همه با هم به آن برسند. مجموعه سرویس‌دهنده‌های تکراری هر کدام داده موردنظر سرویس‌گیرنده را امضاء می‌کنند و ارسال می‌دارند و گیرنده با رمزگشایی رأی‌گیری بین داده‌ها، داده درست را انتخاب می‌کند، این روش سیستم را در برابر خرابیهای بیزانتین نیز حفظ می‌کند.

۹. ۳ کنترل دسترسی

همانطور که قبلاً گفته شد، مجوز به معنای اعطای حقوق دسترسی عناصر سیستم به منابع است. اثبات حقوق دسترسی، کنترل دسترسی نامیده می‌شود.

۹. ۳. ۱ نکات کلی کنترل دسترسی

در یک مدل ساده بین هر عنصری که می‌خواهد به منبعی دست یابد (مانند اینکه بخواهد متریک شیء را فراخوانی کند) و خود منبع، یک فرآیند تحت عنوان ناظر مرجع وجود دارد که مجوزهای دسترسی را کنترل و دسترسی عناصر مختلف را ثبت می‌کند. در این قسمت به صورت خاص روی اشیاء توزیع شده مدل خود را ارائه می‌کنیم و در این بحث به فرآیندی که درخواست فراخوانی یک متد را داشته باشد یک موضوع (Subject) می‌گوییم.

یک روش متداول برای مدل‌سازی حقوق دسترسی استفاده از ماتریس کنترل دسترسی (ACM)[1] است. سطرهای این ماتریس (M) به موضوع‌ها و ستونها به اشیاء نسبت داده شده درون $M[S,O]$ متدهایی که موضوع S می‌تواند از شیء O فراخوانی کند، قرارگرفته است. مشکل این روش این است که معمولاً اندازه این ماتریس بزرگ می‌شود در حالیکه تعداد زیادی از عناصر این ماتریس خالی هستند.

---

[1]. Access Control Matix

روش دیگر تولید لیست کنترل دسترسی (ACL)[1] است. هر شیء ACL خود را دارد و آن لیستی از موضوع‌هایی است که می‌توانند به شیء دسترسی داشته باشند، به همراه متدهایی که هر کدام مجاز به فراخوانی هستند. حسن ACL نسبت به ACM در این است که داده‌ها دیگر به صورت مرکزی نیستند و همچنین عناصر خالی ACM اینجا وجود ندارند. سرویس‌دهنده شیء پس از دریافت درخواست موضوع، چک می‌کند که آیا مجاز هست یا خیر:

روش سوم: توزیع سطری داده‌ها است، به این معنی که هر موضوع لیستی از قابلیتهای (Capability) مربوط به هر شیء را دارد. اگر برای یک شیء قابلیتی وجود نداشته باشد، موضوع حق دسترسی به شیء را ندارد. قابلیت مانند بلیتی است که به موضوع اعطا می‌شود و با آن بلیت مجوّز دسترسی به برخی متدهای شیء را دارد. بدیهی است که موضوع نباید بتواند قابلیتها را تغییر دهد. در بسیاری از موارد، این بلیتها با امضای دیجیتال محافظت می‌شوند. در اینجا موضوع همراه درخواست خود، یک نسخه از قابلیت را نیز ارسال می‌کند. در نتیجه سرویس‌دهنده شیء با توجه به قابلیت اجازه یا عدم اجازه استفاده از شیء را صادر می‌نماید.

در زمانیکه تعداد موضوع‌ها و اشیاء زیاد هستند روشهای دوم و سوم یعنی ACL و قابلیت نیز حجیم می‌شوند. یک روش برای کاهش حجم، استفاده از دامنه‌های حفاظت (Protection Domain) است که در حقیقت مجموعه‌ای از زوج مرتب‌هایی به شکل (حقوق دستیابی و شیء) می‌باشد. هر زوج برای هر شیء مشخص می‌کند که کدام عملیات اجازه اجرا دارند. هر زمان که موضوعی درخواست انجام عملیاتی روی یک شیء را داشت، ناظر مرجع دامنه حفاظت مربوط به آن درخواست را جستجو می‌کند تا مشخص شود که عملیات مجاز است یا خیر.

در شبکه‌های محلّی و اینترنت از دامنه حفاظت به شکل گروه استفاده می‌شود. مثلاً یک صفحه وب را در نظر بگیرید که تنها باید کارمندان مجاز به استفاده از آن باشند. به جای اینکه برای صفحه یک ACL ساخته شود و برای هر کارمند ستونی درون آن درج گردد، می‌توان یک گروه برای کارمندان ساخت و برای دسترسی به صفحه تنها چک کرد که آیا کاربر یک کارمند هست یا خیر. این گروه‌بندی وقتی به صورت سلسله مراتبی در آید، هم قابلیت انعطاف سیستم بالاتر می‌رود و هم مدیریت عضویت گروه‌ها ساده‌تر می‌گردد. عیب عمده‌اش این است که زمانیکه بانک اطلاعاتی عضویت، توزیع شده باشد، جست‌وجوی یک عضو هزینه‌بر می‌گردد.

به جای این‌کار می‌توان به هر موضوع اجازه داد که گواهی‌نامه‌ای (Certificate) را حمل کند تا مشخص شود به چه گروههایی تعلّق دارد. گواهی‌نامه را می‌توان با قابلیت مقایسه کرد.

در مثال گفته شده برای کارمندان یک عنصر دیگر به عنوان نقش کارمند نیز مطرح می‌شود. ممکن است یک کارمند یک یا چند نقش در سازمان مربوطه داشته باشد که این نقشها تغییرپذیر هم هستند. بسته به نقشها انواع مختلف دسترسی نیاز است. پس علاوه بر گروه‌بندی کارمندان، نقشها نیز مهم هستند.

۹. ۳. ۲ دیواره آتش (Fire Wall)

دستیابی خارجی به هر بخش از سیستم توزیع شده، توسط ناظر مرجع خاصی به نام دیوارهٔ آتش کنترل می‌شود و تمام بسته‌های خروجی و ورودی، به کامپیوتر خاصی هدایت می‌شوند و مورد بازرسی قرار می‌گیرند. بنابراین دیواره آتش باید در مقابل هرگونه تهدیدی محافظت گردد.

دو نوع دیواره آتش وجود دارد: دروازه فیلترسازی بسته (Packet-Filtring Gateway) و دروازه سطح کاربرد (Application Level Gateway) که در ترکیب با هم نیز می‌توانند استفاده شوند. نوع اوّل دیواره‌های آتش، براساس آدرس مبدأ و مقصد بسته تصمیم می‌گیرند که بسته عبور کند یا خیر. نوع دوم محتویات پیام‌های ورودی و خروجی را نیز بازرسی می‌کند. برای مثال دروازه نامه (Mail Gateway) نوعی دیواره آتش است که می‌تواند نامه‌هایی را که از حدی طولانی‌تر هستند حذف کند یا هرزنامه‌ها (Spam Mail) را فیلتر کند.

نوع خاصی از دروازه سطح کاربرد به نام دروازه پراکسی یا وکیل (Proxy Gateway) معروف است. این نوع دروازه آتش به عنوان نقطه جلویی نوع خاصی از برنامه‌های کاربردی عمل می‌کند و تضمین می‌کند که فقط پیام‌هایی با معیار خاص اجازه عبور داشته باشند. پراکسی برای کاربر به صورت سرویس‌دهنده وب عمل می‌کند. امّا ترافیک ورودی و خروجی را فیلتر می‌نماید و برخی از درخواستها و صفحات را حذف می‌کند.

---

[1]. Access Control List

۹. ۳. ۳ کد قابل جابه‌جایی آمن

قبلاً دیده شد که در سیستمهای توزیع شده امروزی، جابه‌جایی کدها نقش مهمی دارند و فرآیندهای خاصی مانند عامل‌ها (Agent) هستند که به صورت فراوان و خودکار جابه‌جا می‌شوند. در این جابه‌جایی خیلی مهم است که داده‌ها مورد دستبرد یا تغییر قرار نگیرند.

از سوی دیگر کاربران سیستم باید به نحوی از صحت برنامه‌هایی که از کامپیوترهای دیگر برای آنها ارسال می‌شود مطمئن باشند و سنجش‌های امنیتی نیاز دارند.

حفاظت عامل

گفته شد که در جابه‌جایی عامل‌ها حفاظت از آن بسیار مهم است. حفاظت کامل عامل‌ها عملاً ممکن نیست. در روشی که گفته خواهد شد، عامل‌ها در مورد یکی از مهمترین تهدیدهای امنیتی، یعنی تغییرات غیرمجاز، محافظت می‌شوند. این روش در سیستمی موسوم به Ajanta استفاده شده است. در این سیستم سه راهکار برای تشخیص دستکاری در داده‌های عامل وجود دارد:

روش اوّل حالت فقط خواندنی است که در آن عامل Ajanta قبل از ارسال به سایر ماشینها توسط مالک آن امضا می‌شود. به این ترتیب که مالک ابتدا به کمک یک تابع درهم‌سازی، ابتدا خلاصه پیام را می‌سازد و سپس این خلاصه را با کلید خصوصی خود رمز می‌کند. وقتی عامل به میزبانی دیگر رسید. میزبان جدید از حالت فقط خواندنی خلاصه پیام می‌سازد و آن را با خلاصه پیام امضا شده توسط مالک مقایسه می‌کند. در صورت هر تغییری، مسأله مشخص خواهد شد.

روش دوم که به ثبت‌های فقط فزاینده (Append-Only Log) معروف است، زمانی استفاده می‌شود که عامل اجازه داشته باشد در حین جابه‌جایی، اطلاعاتی را جمع‌آوری کند. یک سابقه یا کارنامه همراه عامل است که در ابتدا خالی است تنها چیزی که در آن است یک عدد به نام C_{init} است که می‌توان آن را یک جمع تطبیقی (Checksum) به حساب آورد. C_{init} از فرمول $C_{init} = K^{+}_{owenr}(N)$ محاسبه می‌شود که K^{+}_{owner} کلید عمومی مالک است و N یک نانس است که تنها مالک می‌شناسد. وقتی عامل به سرویس‌دهندهٔSS می‌رود و قرار است داده x به داده‌های عامل اضافه شود. s، x را امضا کرده و همچنین $C_{new} = K^{+}_{owenr}(C_{old}, Sig(S,x), S)$ را محاسبه کرده و به سابقه اضافه می‌نماید.

وقتی عامل به مالک خود باز می‌گردد، مالک سابقه را از انتها به ابتدا باز می‌کند تا به Cinit برسد اگر به Cinit نرسیم یا در هر مرحله امضایی دستکاری شده باشد، مشخص می‌گردد که تغییر غیرمجاز صورت گرفته است.

روش سوم آشکار ساز انتخابی (Selective Revealing) است در این حالت آرایه‌ای وجود دارد که هر عنصر آن به سرویس‌دهنده خاصی تعلّق دارد که توسط کلید عمومی آن سرویس‌دهنده رمز شده و کل آرایه نیز توسط کلید خصوصی مالک امضا گردیده. اگر توسط میزبان مغرضی یک عنصر تغییریابد سرویس‌دهنده مربوط به آن متوجه تغییر می‌شود.