امنیت تجارت الکترونیک

فصل هشتم:
امنیت چک الکترونیکی

تهیه و تنظیم: دکتر آرش حبیبی لشکری
منبع: کتاب "اصول امنیت برای تجارت الکترونیکی"

اولین نسخه: دی 1393
بروزرسانی: فروردین 1396

فهرست:

- امنیت چک الکترونیکی
  - انتقال مجوز پرداخت
  - پروکسی ها:
    - کربروس
    - پروکسی محدود شده
    - پروکسی آبشاری

امنیت چک الکترونیکی

چکهای الکترونیک دراصل اسناد الکترونیکی شامل همان داده‌های موجود در چکهای کاغذی سنتی هستند.
علاوه بر استفاده یک یا دو مکانیزم امنیتی تراکنش پرداخت که در جلسات قبلی گفته شد یک مکانیزمی که بطور عمومی چکها و معادل الکترونیکی آنها به آن نیاز دارند: انتقال مجوز پرداخت است.

یک انتقال مجوز پرداخت در واقع همان امضاء و تایید یک چک کاغذی (برای مثال، امضا کردن پشت چک) است.
داده‌های دیگر که روی یک چک کاغذی نوشته می‌شود عبارتند از نام پرداخت کننده و اطلاعات حساب، نام دریافت کننده، مقدار پولی که باید به پرداخت شونده داده شود، واحد پول، و تاریخ صدور.

دریافت کننده توسط صاحب حساب (پرداخت کننده) برای برداشت مبلغ معینی از پول تایید می‌شود.

مجوز پرداخت تحت محدودیتهای خاصی از پرداخت کننده به دریافت کننده منتقل می‌شود.

در اینجا به شرح یک نمونه از مکانیزمهای موجود برای امضاهای الکترونیکی روی چکها بر اساس "پروکسی محدود" می‌پردازیم، که برای پیاده سازی NetCheque استفاده می‌شود.

پروکسی‌ها

سیستم NetCheque در موسسه‌ی علوم اطلاعات دانشگاه کالیفرنیای جنوبی (1995) توسعه داده شده و در اصل به عنوان سرویس حسابداری توزیع شده برای سهمیه بندی منابع سیستمی توزیع شده، طراحی شده بود.

در اصل از مدل *اعتباری- بدهکاری* مربوط به پرداخت پشتیبانی می‌کند.

در مدل *اعتباری* مبالغ به یک حساب ارسال می‌شوند و مشتری بعدا مبلغ مورد نیاز را به سرویس پرداخت، پرداخت می‌کند.

در مدل *بدهکاری* وقتی یک چک (یک تراکنش بدهکاری) پردازش می‌شود، حساب مورد نظر بدهکار می‌شود.

مکانیزم شرح داده شده در این جلسه برپایه مدل بدهکاری است. یک NetCheck یک سند الکترونیک است که حاوی داده‌های زیر است:

- نام پرداخت کننده
- شناسه‌ی حساب پرداخت کننده (شماره) و نام بانک
- نام دریافت کننده
- مبلغی که باید پرداخته شود
- واحد پولی
- تاریخ صدور
- امضای الکترنیکی پرداخت کننده
- تایید الکترونیکی دریافت کننده

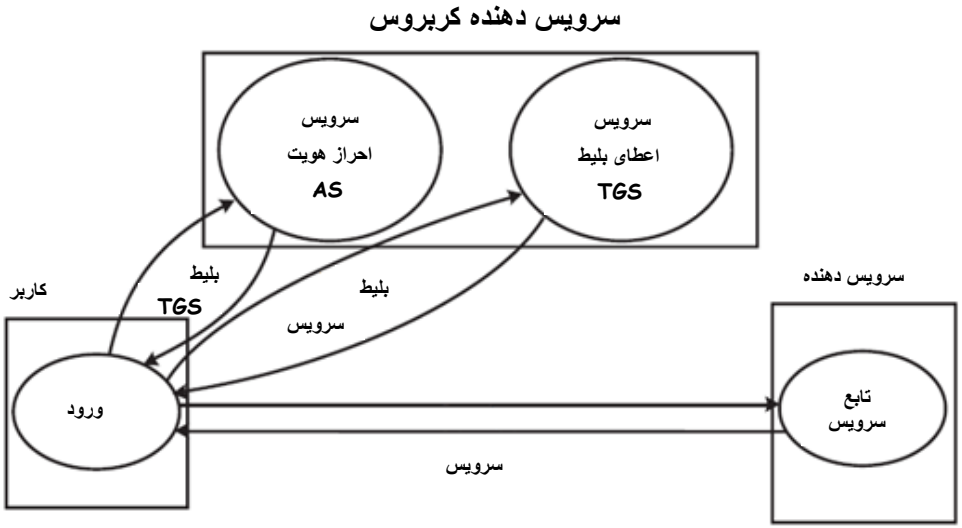
کربروس

یک پروکسی یک نشانه است که به شخصی این اجازه را می‌دهد تا با حقوق و امتیازاتی که به پروکسی اعطا شده بتواند عملیاتی را انجام دهد.

پروکسی محدود شده، پروکسی است که شرایطی برای استفاده از آن وجود دارد. در مثال چک، محدودیتها، پرداخت شونده (مشتری منتخب)، مقدار پولی که باید پرداخت شود، و تاریخ صدور می‌باشند.

پروکسی های NetCheck بر اساس بلیطهای کربروس طراحی شده اند.

در اصل کربروس در موسسه‌ی تکنولوژی ماساچوست (MIT) به عنوان یک سیستم احراز هویت توزیع شده در سال 1986 طراحی شده است. (شکل).



کربروس

وقتی یک کاربر در یک سیستم توزیع شده می‌خواهد از سرویس S استفاده نماید (مثل یک پرینتر)، باید یک *بلیط سرویس* از سرورس اعطای بلیط (TGS) بگیرد.

اما قبل از درخواست هر بلیطی، کاربر باید هویت خود را توسط سرویس احراز هویت (AS)، تایید نماید.

اگر احراز هویت موفقیت آمیز باشد، کلاینت (C) یک *بلیط* TGS و یک کلید نشست K_{C-TGS} می‌گیرد تا برای درخواست یک بلیط سرویس از TGS استفاده نماید:

$$\{C, TGS, t_1, t_2, K_{C-TGS}\}K_{TGS}, \{K_{C-TGS}, n_1\}K_C$$

در اینجا t_1 و t_2 شروع و پایان دوره‌ی اعتبار بلیط هستند و همچنین n_1 و n_2 شماره های یکبار مصرفی اند (رشته های تصادفی) که برای تازگی پیام استفاده شده‌اند.

متغیر K_{TGS} کلید رمز TGS است، بنابراین فقط TGS می‌تواند بخش اول (کلید TGS) را رمزگشایی نماید.

K_C نیز کلید رمز کاربر است (گذر واژه‌ی درهم شده)، بنابراین کاربر می‌تواند کلید نشست K_{C-TGS} را برای استفاده در برقراری ارتباط با TGS رمزگشایی نماید.

حال کاربر می‌تواند یک بلیط سرویس درخواست کند. TGS، بلیط سرویس و یک کلید جلسه‌ی K_{C-S} را برای کاربر ارسال می‌نماید تا بتواند سرویس را خودش درخواست نماید:

$$\{C, S, t_1, t_2, K_{C-S}\}K_{uc}, \{K_{C-S}, n_2\}K_{C-TGS}$$

در اینجا K_S کلید رمز سرویس‌دهنده است، بنابراین فقط سرویس‌دهنده می‌تواند بخش اول (بلیط سرویس) را رمزگشایی و تصدیق نماید. اگر بلیط سرویس معتبر باشد، سرویس به کاربر اعطا خواهد شد. تمام کلیدها (به جز K_{C-S}) برای سرویس‌دهنده کربروس شناخته شده‌اند، و هر سرویس‌دهنده باید یک کلید رمز با دیگر سرویس‌دهنده‌ها به اشتراک بگذارد.

پروکسی محدود شده

بلیط TGS کربروس در واقع یک پروکسی محدود شده است. محدودیت در اینجا، بازه‌ی زمانی (t_1, t_2) است که در این بازه بلیط معتبر خواهد بود. یک فرم عمومی شده‌ی پروکسی محدود را می‌توان مثل زیر نوشت:

$$\{< \text{check} >, K_{proxy}\} K_{payer}, \{K_{proxy}, \text{nonce}\} K_{payee}$$

اعطا کننده مسئولی است که به نمایندگی از او یک پروکسی اجازه ی دسترسی پیدا می‌کند (برای مثال، TGS). اعطا شونده مسئولی است منتخب برای ایفای نقش به جای اعطا کننده. در مورد یک چک، محدودیتها توسط داده-های چک نمایش داده می‌شوند

پروکسی‌های آبشاری:

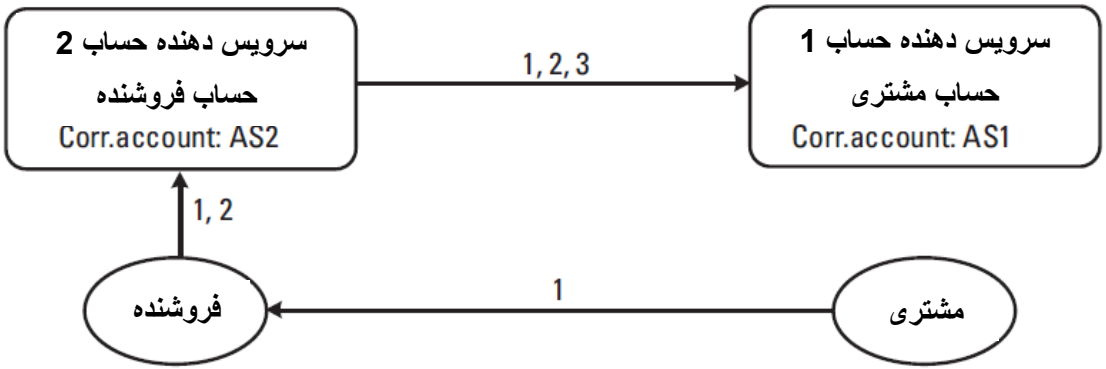
متاسفانه، در واقعیت پدیده‌ها معمولا خیلی ساده نیستند، چراکه یک پرداخت کننده و یک پرداخت شونده لزوما در یک بانک حساب ندارند.

اگر این گونه باشد، چک از میان چندین *سرویس‌دهنده حسابداری* در سیستم NetCheque تایید می‌شود. یک نمونه سلسله مراتب حسابداری در شکل صفحه بعد نشان داده شده است.

مشتری یک بلیط کربروس که برای احراز هویت کاربر به سرویس‌دهنده حسابداری استفاده خواهد شد، تولید می‌کند. این بلیط در فیلد امضای چک قرار داده شده و برای فروشنده ارسال می‌شود (بخش "1" در شکل).

فروشنده یک عامل احراز هویت برای حمایت از چک به نام دریافت کننده برای اینکه سپرده‌ی بانکی فقط به حساب دریافت کننده واریز شود، تولید می‌کند (بخش "2" تصویر). فروشنده آنرا، همراه با پیام اصلی مشتری، برای اولین سرویس‌دهنده حسابداری ارسال می‌نماید (AS_1):

پروکسی‌های آبشاری



AS_1 کلید رمز $K_{merchent}$ را با فروشنده به اشتراک می گذارد، بنابراین می تواند K_{proxy1} را از پروکسی 1 بگیرد و از آن برای رمزگشایی بلیط در پروکسی 2 استفاده کند. نهایتا، AS_1یک احراز هویت کننده برای حمایت چک به نام سرور حسابداری پرداخت شونده، برای اینکه سپرده ی بانکی به حساب متناظر در AS_1 AS_2 واریز شود ("3" در شکل) انجام می دهد.

هر سه پروکسی آبشاری به سرور حسابداری مشتری AS_2 فرستاده می شوند. این سرور شروع کننده ی تصدیق پروکسی های آبشاری با بلیط در پروکسی 1 می باشد، از آنجاییکه کلید رمز $K_{customer}$ را با مشتری به اشتراک می گذارد. با این کلید، AS_2 می تواند K_{proxy1} را بگیرد و از آن برای رمزگشایی بلیط در پروکسی 2 استفاده می کند. با K_{proxy2}از پروکسی 2، AS_2 می تواند بلیط را از پروکسی 3 رمز گشایی کند. در نهایت، این بلیط بیانگر این است که چک بهتر است که در حساب متناظر AS_1 سپرده گذاری شود.

$$\text{Proxy 1: } \{<\text{check}>, K_{proxy1}\}K_{customer}, \{K_{proxy1}, n_1\}K_{merchant}$$

$$\text{Proxy 2: } \{\text{deposit} < \text{check} > \text{to } AS_1, K_{proxy2}\}K_{proxy1}, \{K_{proxy2}, n_2\}K_{AS_1}$$

$$\text{Proxy 3: } \{\text{deposit} < \text{check} > \text{to } AS_2, K_{proxy3}\}K_{proxy2}, \{K_{proxy3}, n_3\}K_{AS_2}$$

خلاصه: امنیت چک الکترونیکی، انتقال مجوز پرداخت ، پروکسی ها: کربروس، پروکسی محدود شده، پروکسی آبشاری

جلسه بعدی: پروتکل تجاری باز اینترنتی

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