Debreceni Egyetem

Informatika Kar

BIZTONSÁGOS ELEKTRONIKUS FIZETÉSI RENDSZEREK

Témavezető:
Dr. Huszti Andrea
Egyetemi adjunktus

Készítette:
Lőrinczi Balázs
Mérnök Informatikus (B.Sc)

Debrecen

2010

Tartalomjegyzék

Bevezetés

Napjainkban szinte mindenki találkozik elektronikus fizetési rendszerekkel. Elég, ha elmegyünk a boltba vásárolni, vagy interneten rendelünk valamit, máris elektronikus úton fizethetünk. Az utóbbi években a fizetések ilyen fajta lebonyolítása egyre népszerűbb lett, habár az emberek még mindig óvatosan viszonyulnak az ilyen típusú fizetésekhez, mivel a vásárlás nem személyes kontaktusra épül.

Nem túlzás azt mondani, hogy a legtöbb ember rendelkezik bankkártyával. A dolgozat végére egy olyan protokoll létrehozása a cél, amely az elterjedt SSL és a kevésbé elterjedt SET protokoll mellett egy harmadik lehetőséget vázol fel a biztonságos kártyás fizetés megvalósításához. Nagyító alá vesszük az SSL alapú fizetés lépéseit. Itt egy konkrét fizetés esetén megvalósuló lépéseket ismerhetjük meg. Azt, hogy mi játszódik le egy ilyen háromszereplős (vevő, kereskedő, bank) fizetés során. Létre akarunk hozni egy olyan arany középutat a kártyás fizetéseknél, amely biztonságosabb, mint az SSL, de ugyanakkor ne legyen olyan sok lépése, mint a SET protokollnak és költségeiben is maradjon reális. Az így létrejött protokoll tulajdonságait hasonlítjuk majd össze az SSL alapú fizetéssel, hogy miben bizonyul jobbnak, miben bizonyul rosszabbnak nála, tulajdonságait összevetjük a SET tulajdonságaival. A protokoll létrehozásához azonban a dolgozatban először tisztáznunk kell, mi is az elektronikus kereskedelem. A kereskedelem mozgatórugója a pénz áramlása, amit az elektronikus kereskedelemben az elektronikus fizetési rendszerek valósítanak meg. A kérdés adott: vajon milyen tulajdonságai lehetnek, milyen tulajdonságokkal kell rendelkeznie egy elektronikus fizetési rendszernek? Mint látni fogjuk az emberek számára (érthető módon) a biztonság a legfontosabb. Arra keressük a választ, hogy a biztonság megvalósításhoz, milyen kriptográfiai szempontokat kell szem előtt tartani, megvalósítani a rendszerekben. A második egységben a kriptográfiai eszközöket ismerhetjük meg, amelyeket a mai elektronikus fizetési rendszerek alkalmaznak. Azok az eszközök, amelyek az előbb említett kriptográfiai szempontokat megvalósítják. Hogyan lehet az biztosítani, hogy adataink ne kerüljenek illetéktelen kezekbe? Milyen algoritmusok és milyen módszerek vannak a biztonságos kommunikáció fenntartására? Ezekre a kérdésekre keressük a választ ebben az egységben. Tulajdonképpen azokat a folyamatokat ismerhetjük meg, amelyekkel a felhasználó nem találkozik egy elektronikus fizetés során, de attól azok még a háttérben végbemennek. A következő egységben megnézzük a manapság használt különböző alapon működő fizetési

rendszereket. Megnézzük, hogy ezek a rendszerek a kriptográfiai eszközök széles palettájáról mit használnak, hogy kielégítsék a velük szemben támasztott követelményeket. Bemutatásra kerülnek, hogy az adott rendszerben hogyan zajlik le egy fizetés, milyen lépések sorozatával lehet leírni egy tranzakciót, milyen egyedi tulajdonsága van az adott rendszernek, milyen jellemzői vannak, milyen típusú fizetésekhez találták ki őket. Ezek ismeretében fogjuk megalkotni a javasolt protokollunkat.

Az elektronikus kereskedelem

Maga a kereskedelem szó már ősidők óta létezik, hiszen nagyon ritkán adatott meg, hogy a környezet ahol az emberek éltek, mindennel ellátta őket. Viszont amiből többletük volt, az képezhette kereskedelmük alapját, így létrejöttek a nagy kereskedelmi utak. Manapság már nem kell feltétlenül a kereskedelmi utakat bejárnunk, hogy hozzájussunk egyes termékekhez. Az elektronikus kereskedelem leegyszerűsítette a dolgainkat. Az elektronikus kereskedelem osztályozható a fogyasztó és az ellátó szerepe szerint. A három leggyakrabban előforduló változata:

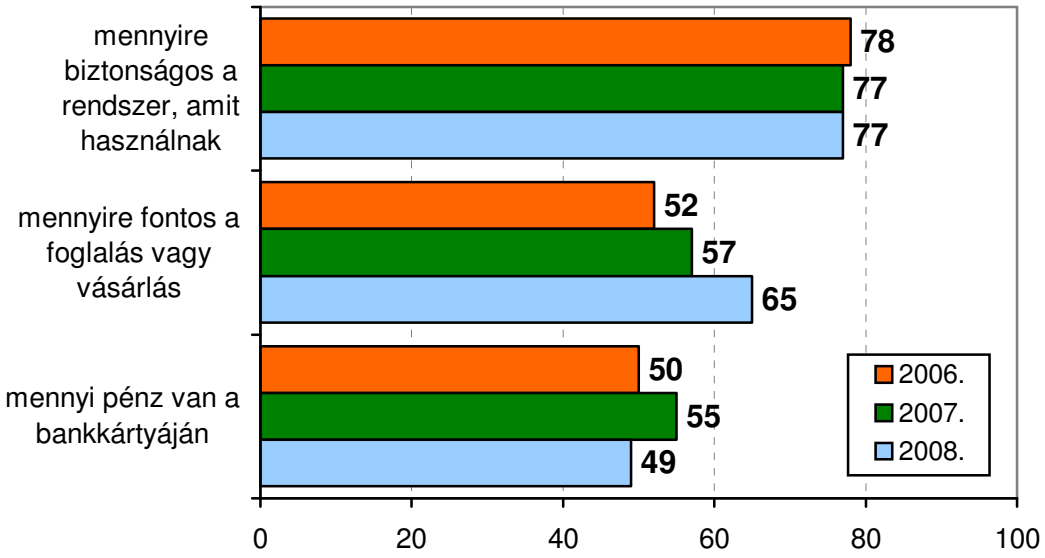
C2C (consumer-to-consumer): két fogyasztó között jön létre az elektronikus kereskedelem egy harmadik félen keresztül. Tipikus példa az on-line aukciók, ahol a harmadik félnek csak közvetítő szerepe van (a termékek minőségét nem ellenőrzi).

B2C (business-to-consumer): ide sorolunk minden interneten történő vásárlást, hiszen a termékét vagy szolgáltatását kínáló vállalkozás és fogyasztó között alakul ki a kapcsolat.

B2B (business-to-business): mondhatjuk, hogy ez a vállalkozás-vállalkozás típus a leggyakoribb, ugyanis egy tipikus ellátási lánc sok B2B (egy autónál az üveg vagy abroncsok beszerzése… stb.), viszont csak egy B2C kapcsolatból áll (amikor átveszi az autót a fogyasztó).

Elektronikus fizetések megvalósítása

Habár napjainkban elengedhetetlen az elektromos fizetési rendszerek jelenléte, gondoljunk csak a hitelkártyával való fizetésre boltokban, vagy az interneten végzett banki átutalásokra, számlák rendezése elektronikus úton vagy csak egyszerűen az ATM – ből való pénzfelvételre; mégis az emberek többsége szkeptikusan áll hozzá az ilyen szolgáltatásokhoz. Ennek az lehet az oka, főleg az idősebbeknél, hogy inkább már megmaradnak a régi módszereknél és nem akarnak megismerkedni az újításokkal. Azoknál, akik hajlandók lennének nyitni az újítások felé olyan tényezők merülnek fel, hogy mennyire fontos a szóban forgó vásárlás vagy szolgáltatás, mennyi pénz van az ember számláján és a legfontosabb kérdés minden elektronikus fizetési rendszernél: a biztonság.



1. ábra: Mitől függ, hogy megadná-e a bankkártyája számát? (azok százalékában, akik rendelkeznek bankkártyával és internetes vásárláskor lehet, hogy megadnák)

Az elektronikus fizetés rendszer (EPS – Electronic Payment System)

Hagyományos kereskedelemben mindig beszélhetünk egy vásárlóról és egy kereskedőről, aki termékéért vagy szolgáltatásáért pénzt kap cserébe. Azonban, ha elektronikus kereskedelemről van szó, mindig van legalább egy olyan pénzintézet, amely kapcsolatot teremt az elektronikus környezet és a pénz között (hitelesíteni tudja a vásárlást fedezeti szempontból).

Az előbb említett felek (vásárló, kereskedő, bank) képezik a fizetési rendszer modelljét[6]. A harmadik fél azaz a bank, nem mindig egy intézményt takar. Sok esetben ugyanis a kereskedőnek és a vevőnek nem ugyanannál a pénzintézetnél van számlája, így a harmadik fél tulajdonképpen a vevő és a kereskedő bankját jelenti. Természetesen ez a két bank kommunikál egymással, amiből mi csak annyit veszünk észre, hogy számlánkról levették a fizetendő összeget. A valódi pénzmozgás tehát a két bank között a háttérben zajlik.

Egy elektronikus fizetési rendszer lehet:

- off-line: ilyenkor nem beszélhetünk harmadik félről, a tranzakció a vásárló és a kereskedő között megy végbe, a hitelességről a kereskedő győződik meg nem a bank.

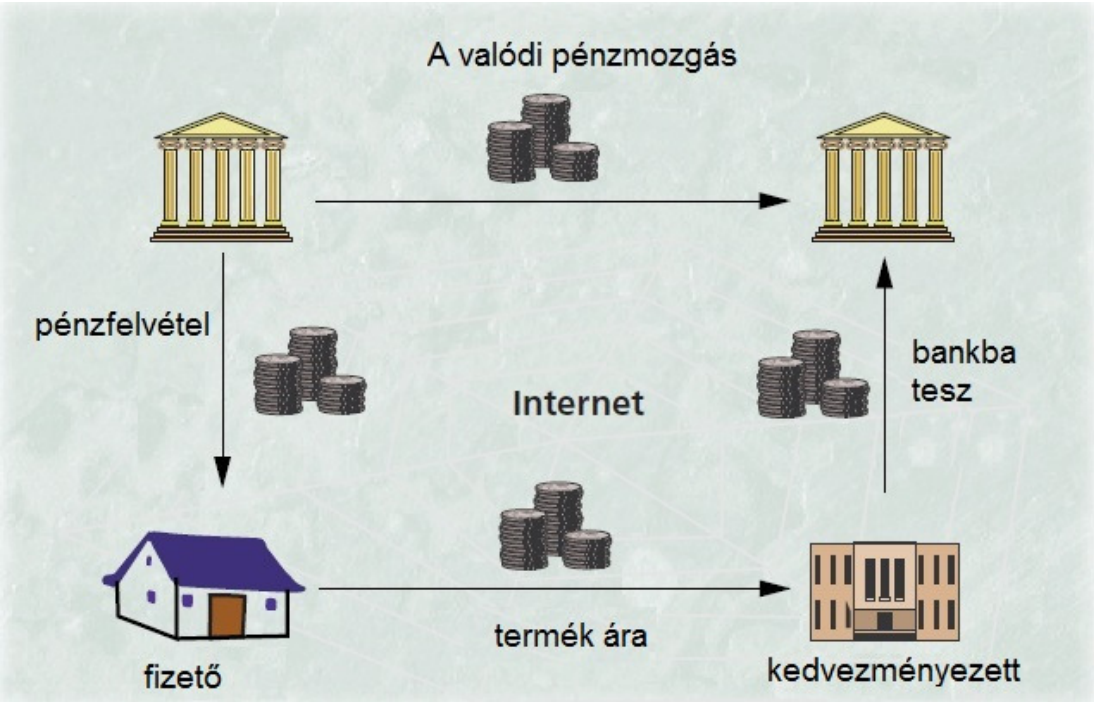
- on-line fizetésnél a kereskedő a bank segítségével tudja leellenőrizni a vásárló hitelképességét. Több kommunikációt igényel a felek között, de biztonságosabb is.

- kredit rendszer: más néven hitel alapú, ahol egy vásárlás során először a számlánkat terheljük, amíg hitelkeretünk engedi, majd később fizetünk.

- debit rendszer: itt pedig pontosan fordítva, először pénzt kell tennünk a számlánkra, majd abból fizetjük ki a terméket vagy szolgáltatást. Itt a vásárlás során a fizetőképesség ellenőrzése megtörténik.

- azonosított: azonosított on-line rendszernél a vásárlót a bank és a kereskedő is azonosítja, visszaigazolást vár a kereskedő a banktól. Azonosított off-line esetén, mivel az on-line visszaigazolás nem lehetséges, ezért ilyen esetekben tipikusan egy chipkártya látja el ezt a feladatot.

- anonim: anonim on-line rendszernél a kereskedő nem azonosítja a vásárlót, csak visszaigazolást vár a banktól. Csak a bank azonosítja a vásárlót. Anonim off-line esetben még ez sem valósul meg: a vásárló kiléte titokban marad mind a kereskedő, mind a bank előtt.



2. ábra: Fizetési modell

Kriptográfiai szempontok

Habár egy rendszer biztonságának kialakításakor sok összetevőt kell figyelembe vennünk: mire használjuk a rendszert; milyen műveletek végezhetők vele; milyen környezetben használjuk; a biztonságra költött pénz ne legyen több, mint a védeni kívánt érték… stb. az alábbi összetevőknek meg kell jelennie:

- partnerazonosítás (autentikáció)
- jogosultság
- adatintegritás
- bizalmasság
- rendelkezésre állás, megbízhatóság

Partnerazonosítás (autentikáció)

Az autentikáció szó jelentése azonosítás: azonosítjuk a résztvevő felet. Valamilyen művelet elvégzése előtt a műveletet végrehajtani kívánó személy jogosultságának ellenőrzése. Erre a feladatra az alábbi módszerek léteznek napjainkban:

- jelszavas: az azonosítás legjobban elterjedt módszere. A jó jelszavakkal szemben támasztanunk kell néhány elvárást. Legalább 8-10 karakter legyen; betűkön kívül számokat és egyéb karaktereket is tartalmazzanak; olyan legyen, amit könnyen megjegyezhetünk, de ne kapcsolódjon személyes adatainkhoz, például születési év vagy nevünk ne legyen a jelszó része, hiszen adataink birtokában könnyen lehet jelszavakat generálni. Nem szerencsés értelmes szavakat jelszóként választani, mert ilyenkor úgynevezett szótáras támadásnak tesszük ki magunkat, azaz egy algoritmus a jelszavakat tároló password fájl és egy nagyméretű szavakat tartalmazó szótárból próbálgatással hasonlítja össze a szavakat. Az sem utolsó szempont, hogy jelszavunk olyan legyen, hogy meg tudjuk jegyezni, ha pedig ez nem megy, akkor se ragasszuk ki monitorunkra.

- kulcsok használata: hagyományos titkos autentikációs kulcsok, chippel ellátott kártyák, mágneskártyák tartoznak ebbe a kategóriába. A birtoklás azonban lehet hátránya is ennek a módszernek, hiszen elhagyhatjuk, esetleg ellophatják kulcsunkat.

- biometrián alapuló azonosítás: egyedi tulajdonság alapján történik az azonosítás, úgymint: ujjlenyomat, gépelési szokások, arckép vagy retina. Az egyetlen mód, ahol elsősorban kilétünket igazoljuk, nem csupán a kártya birtoklása vagy a jelszó ismerete igazol minket. Azt, hogy az adott személyhez milyen jogosultságok tartoznak, már az aktuális rendszer dönti el. Ennek az azonosításnak a hátránya, hogy speciális, drága hardvert igényel és még így sincs kizárva a megtévesztés lehetősége.

Jogosultság

Azt írja elő egy adott informatikai rendszerben, hogy az egyed, milyen információkhoz, erőforrásokhoz férhet hozzá. Így kialakul egy hierarchia, amelynek alján az egyszerű felhasználó van. Az egyedeket pedig lehet csoportosítani aszerint, hogy milyen jogosultságokkal rendelkeznek.

Adatintegritás

Az integritás az jelenti, hogy az adatok nem változnak meg, nem módosulhatnak helytelen kezelés vagy támadás miatt, továbbá üzemzavar esetén, átviteli hibák vagy merevlemez sérülése miatt.

Bizalmasság

Bizalmasság: egyes tranzakciók során a résztvevők kilétüket titokban akarják tartani. Ilyenkor nem csupán a vásárló vagy a kereskedő személyéről van szó, hanem a tranzakcióhoz köthető információkról is: a tranzakció tárgya, esetleg a mennyisége és így tovább. Alapvető követelmény, hogy a kereskedő ne rendelkezzen a vásárló személyes adataival (pl.: bankkártya száma), a bank viszont ne tudjon a vásárlás bizonyos adatairól (pl.: milyen termékről van szó).

Rendelkezésre állás és megbízhatóság

Minden résztvevő akármikor kezdeményezhet tranzakciót vagy részt vehet tranzakcióban. A megbízhatóságot úgy tartják fenn, hogy a tranzakciók elemiek. Vagy nem történnek meg vagy teljesen végbemennek. Az nem történhet meg, hogy ismeretlen vagy ellentmondásos, következetlen állapotba kerüljön egy tranzakció. Ez off-line rendszerek esetén nem jelent

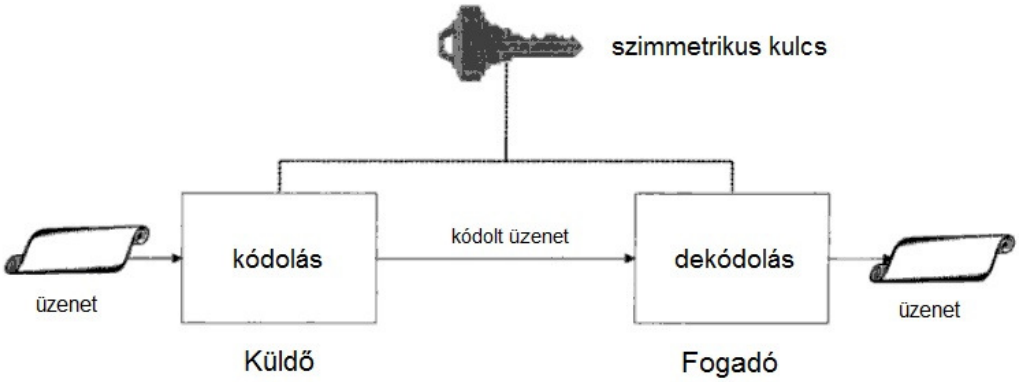
jelentős problémát a tranzakció elhalasztásán kívül. On-line rendszerek használatakor összeomlás esetén megbízható tárolókra van szükség és speciális újraszinkronizáló protokollokra.

A biztonságos elektronikus fizetési rendszerek kriptográfiai eszközei

Gyakran megesik, hogy vannak olyan információk egy kommunikáció során, amit a felek bármely más kívülálló féllel nem akarnak megosztani. Ezért üzeneteiket titkosítják, amit megtehetnek szimmetrikus vagy aszimmetrikus kriptorendszer használatával.

Szimmetrikus titkosítás

A szimmetrikus kriptorendszer alapvető gondolata, hogy a szereplők egy közös kulcs segítségével kommunikálnak, erre utal a szimmetrikus kifejezés. A kulcs, amivel a küldő titkosítja üzenetét, ugyanaz a kulcs - vagy a dekódoló kulcs a kódoló kulcsból könnyen (polinomiális időn belül) kiszámítható - amivel a címzett visszafejti a hozzá érkező üzenetet. A felek megállapodnak, melyik kriptorendszert használják, és a kulcscsere megtörténik a kommunikáció kezdete előtt. A kulcscserének biztonságos csatornán keresztül kell megtörténnie. Zárt rendszerben ez nem jelent problémát, de nyílt hálózaton igen sebezhetővé válik lehallgatás útján: a kulcs birtokában bármely üzenet visszafejthető. Gondot jelenthet, hogy n felhasználó esetén n*(n-1)/2 kulcsra van szükség.



3. ábra: A szimmetrikus titkosítás modellje

Formálisan leírva:

Kódolás: $Enc_{KA}(m) = c$,

Dekódolás: $Dec_{KA}(c) = m$,

ahol m az üzenet, KA a közös kulcs, c a titkosított üzenet.

A szimmetrikus titkosító algoritmusoknak két kategóriája van:

- blokkrejtjelező: egy nyílt blokkot (rendszerint 128 vagy 64 bit) ugyanakkora rejtett blokkba transzformál.
- folyótitkosítás: a forrásfájl bitjeit és egy véletlen bitsorozat bitjei között kizáró vagy (xor) műveletet hajt végre.

A blokkrejtjelezővel szembeni általános elvárások:

- teljesség: a kimenet minden bitje minden bementi bittől és minden kulcsbittől függjön.
- statisztikai függetlenség: ne legyen statisztikai kapcsolat a nyílt blokkok és a rejtjeles blokkok között.
- lavina hatás: a bemenet egy bitjének változására a kimeneten minden bit 0.5 valószínűséggel változzon meg. Ugyanígy egy kimeneti bit változása eredményezzen véletlen változásokat a bemenet bitjeinél.

A DES algoritmus

Az egyik legelterjedtebb, legismertebb algoritmus. A hetvenes években fejlesztették ki az IBM-nél. A Shannon-féle keverő transzformáción alapul. A DES[11] struktúrájáról elmondható, hogy a bementi és kimeneti mérete 64 bit, a kulcs mérete pedig 56 bit. Meg kell jegyezni, hogy az 56 bites méret manapság már kevés, hiszen az úgynevezett Brute Force technikával, kimerítő kulcskereséssel feltörhető (2^{56} lehetőség végig vizsgálatával), épp ezért használják a 3DES, azaz 168 bites változatot.

Röviden a DES-ről:

A DES algoritmus a bementi biteken egy kezdeti permutációt hajt végre: egy 8*8 M mátrixon megadjuk, hogy az eredeti bitek hogyan helyezkedjenek el. Például: M[1,1] = 58 azt jelenti, hogy az eredeti 58 bitje lesz az első bit. Ezt követően a biteken 16-szor végrehajtja a Feistel transzformációt. Minden réteghez tartozik egy rétegkulcs, ami a titkos kulcsból származtatott érték. A rétegkulcsokat a kulcsütemező algoritmus készíti el. Utolsó lépésként szintén permutációt hajtunk végre, mégpedig az eredeti permutáció inverzével.

A dekódolás igazából nem más, mint a rétegkulcsok fordított sorrendben történő használata. Manapság a 3DES az elterjedt, ami tulajdonképpen a DES kódolási műveleteinek (és dekódolási műveleteinek) háromszor egymás utáni elvégzése.

Aszimmetrikus titkosítás

Az aszimmetrikus vagy nyilvános kulcsú titkosítás során a résztvevő felekhez egy kulcspár tartozik: van egy privát kulcsa (SK), illetve van egy nyilvános kulcsa (PK). A nyilvános kulcs, a nevéből adóan bárki számára elérhető például egy adatbázisból. Ennek megfelelően a két kulcsot úgy szerkesztik meg, hogy a nyilvános kulcsból polinomiális időn belül ne lehessen a hozzá tartozó privát kulcsot rekonstruálni. Az aszimmetrikus titkosítás azon alapul, hogy még nem találtak polinomiális idejű algoritmust, amely az előbb említett kulcsgenerálást meg tudná oldani, azonban ez nem jelenti azt, hogy nem is létezik ilyen algoritmus.

Legyen A és B a két kommunikáló fél, akik két-két kulcspárral rendelkeznek (SK_A és PK_A, illetve SK_B és PK_B), ebben az esetben egy üzenetváltás folyamata:

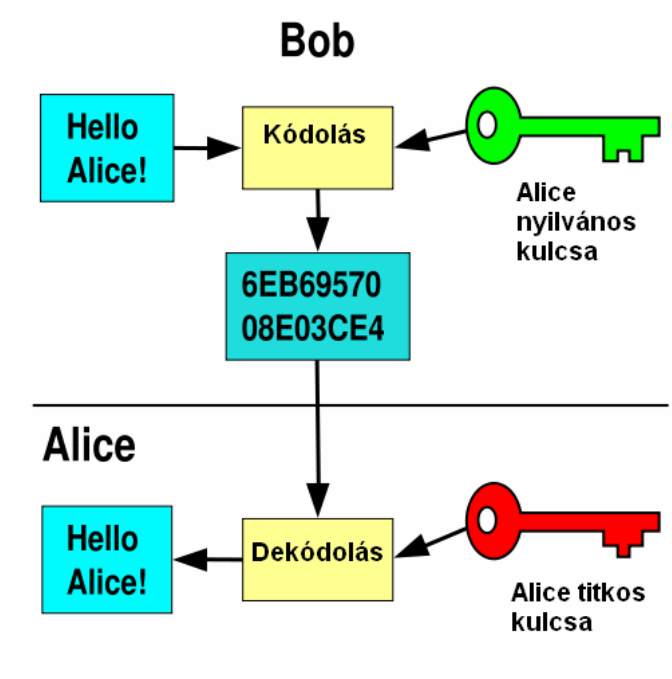
1. első lépésben egy x bizalmas üzenetet akar A küldeni B-nek. Ezt úgy teszi meg, hogy B nyilvános kulcsával titkosítja az üzenetet és elküldi neki. Üzenet= Enc_{PKB} (x).

2. második lépésben az x titkosított üzenetet B visszafejti saját (SK_B) kulcsával. Mivel ilyen kulcsa csak neki lehet, ezzel tulajdonképpen azonosítja magát.

3. harmadjára az x üzentre B válaszolhat A-nak. Elküld neki egy x' üzentet A nyilvános kulcsával titkosítva. Üzenet= Enc_{PKA} (x')

4. negyedik lépésben A megkapja a x' titkosított üzenetet és privát kulcsával (SK_A) dekódolja azt.

Egy ilyen üzenetváltással tulajdonképpen azonosítani lehet a kommunikációban résztvevő feleket. Az azonosítás a dekódolás folyamatában valósul meg. A felek privát kulcsukkal azonosítják magukat (SK_A, SK_B). Ha B válaszüzenete, azaz x' tartalmazza az A által küldött x üzenetet, akkor A biztos lehet abban, hogy olyan valakivel kommunikál, aki rendelkezik az SK_B privát kulccsal. Ugyanilyen módon lehet A-t is azonosítani.



4. ábra: Aszimmetrikus kulcsú titkosítás modellje

Az RSA algoritmus

A nyilvános kulcsú titkosítás legelterjedtebb fajtája. Az RSA algoritmust három részre bonthatjuk: kulcsválasztás, kódolás, dekódolás. A kulcsválasztás lépései:

- a kulcsválasztás során véletlenszerűen választunk 2 nagy prímszámot: p és q ($p \neq q$). Manapság legalább 500 bit méretű bináris számokkal dolgoznak.
- meghatározzuk p és q szorzatát, amit n-nel jelölünk. Ezután kiszámítjuk $\varphi(n)= (p-1)(q-1)$. Választunk egy e számot, ami (p-1), illetve (q-1) képest relatív prím, tehát legnagyobb közös osztója 1. Az előbbiből következik, hogy e $\varphi(n)$-hez is relatív prím.
- utolsó lépesben meghatározzuk az e inverzét modulo $\varphi(n)$, olyan d számot keresünk, amelyre $1 < d < \varphi(n)$ és $ed = 1 \pmod{\varphi(n)}$.

A kulcs nyilvános része PK=(n, e) egy mindenki által hozzáférhető nyilvántartásba kerül. A SK = d pedig a kulcs titkos része lesz.

Legyen A és B két kommunikáló fél. Ha A üzenetet akar küldeni B-nek, meg kell tudnia B nyilvános kulcsát (n_B, e_B). Az üzeneten egy kölcsönösen egyértelmű transzformációt hajt

végre, melyben nemnegatív egészek sorozatává alakítja, amelyek mindegyike kisebb, mint n_B. A blokkokon a titkosítást egyesével hatja végre. Az i-edik blokk: $C_i = (M_i^{eB}, \text{mod } n_B)$.

Visszafejtéskor B minden blokkra elvégzi: $M_i = (C_i^{dB}, \text{mod } n_B)$. Az így előálló sorozatra a titkosításban használt transzformáció inverzét alkalmazva kapjuk az üzenetet.

Digitális aláírás

A nyilvános kulcsú titkosítás egyik legfontosabb felhasználási területe a partnerazonosítás és a letagadhatatlanság, illetve az adatintegritás egyidejű biztosítására szolgáló digitális aláírás. A digitális aláírás alapgondolata, hogy a kommunikációban résztvevő feleket titkos kulcsuk alapján lehet azonosítani, hiszen ezt az információt máshonnan nem lehet kikövetkeztetni. Ez biztosítja, hogy egy titkos kulcs használata egy adott személyhez köthető.

A digitális aláírás menete:

1. az aláírandó dokumentumból egy hash függvény segítségével lenyomatot készítünk. Ezt a lenyomatot A aláírja titkos kulcsával (SK_A), majd elküldi a lenyomatot és a dokumentumot B-nek.

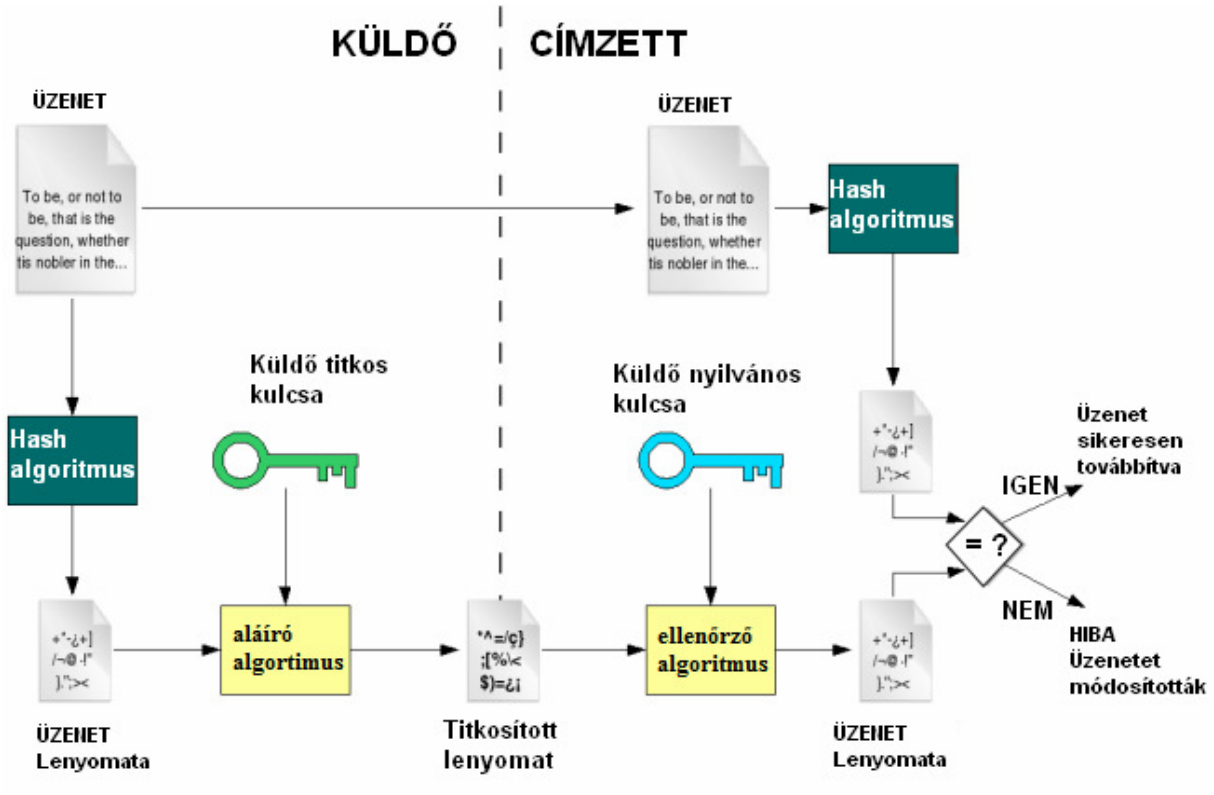
2. B visszafejti a lenyomatot A (PK_A) nyilvános kulcsával. Mivel a hash algoritmus nyilvános, ezért elkészíti a küldött dokumentum lenyomatát. A visszafejtett lenyomatnak és a dokumentum lenyomatának egyeznie kell.

Így bizonyítva van az üzenet és az aláírás integritása, mert egy megváltozott dokumentumhoz megváltozott lenyomat tartozna, és a módosult aláírás következtében a visszafejtett lenyomat sem egyezne a jó lenyomattal. A letagadhatatlanság is adott, hiszen privát kulccsal csak a megfelelő személy rendelkezhet.

Formálisan:

Digitális aláírás készítése: $\text{Sign}_{SKA}(m) = s$.

Aláírás ellenőrzése: $\text{Ver}_{PKA}(m, s) = 1$ (ha egyezik), 0 (ha nem).



5. ábra: Digitális aláírás modellje

Üzenethitelesítés

Üzenethitelesítést azért használunk, mert sok alkalmazás az üzenetek titkossága mellett az üzenetek integritásának védelmét követeli meg. Mivel a titkosítás önmagában nem biztosít integritást, ezért erre a célra külön algoritmust kell használnunk. Az üzenethitelesítés tehát a küldött üzenetek integritását és hitelességét biztosítja, hasonlóan a digitális aláíráshoz. Ezáltal ellenőrizhetjük a küldő személyazonosságát, illetve az üzenetben bekövetkezett véletlen vagy szándékos változásokat fel tudjuk ismerni. A leggyakoribb módszer az üzenethitelesítő kódok (Message Authentication Code) használata.

A MAC

Az üzenet azonosító kód tulajdonképpen egy ellenőrző összeg a feladó és az üzenet címzettje között. Segítségével alakul ki a biztonságos kommunikáció Ez egy szimmetrikus kulcsú hitelesítési módszer, aminek az értékének a kiszámítására, egy kulcsra van szükség. A kulcs titkos és csak azok tudják, akik az üzenetet váltják. Az azonosító kód értéke függ az üzenettől és az előbb említett közös kulcstól.

Egy üzenetváltás MAC használatával:

1. a küldő és a vevő rendelkezik egy közös K_{KV} kulccsal, amit semmilyen harmadik fél nem ismer.

2. a küldő az m üzenet elküldése előtt kiszámolja a hozzá tartozó kódot a közös kulcsuk segítségével, majd elküldi az üzenetet kiegészítve az előbb kiszámolt üzenet azonosító kóddal.

3. a vevő megkapja az üzenetet és a kódot. A vevő fogja a közös kulcsot (K_{KV}) és a küldő üzenetére alkalmazva kiszámolja az azonosító kódot. Ha ez megegyezik a küldő által mellékelt kóddal, akkor mindent rendben talál, biztos lehet a küldő fél személyazonosságában.

Ha az azonosító nem stimmel, akkor az üzenet integritása sérülhetett, amíg átjutott a csatornán, ami történhetett szándékos beavatkozás során, vagy egyszerűen nem a megfelelő titkos kulcs került alkalmazásra. Mivel az üzenet tartalma megváltozott, az ellenőrző kód is más lesz, ennek a következmény az lesz, hogy a címzett elutasítja az üzenetet.

A MAC kód kiszámítására többféle algoritmus létezik:

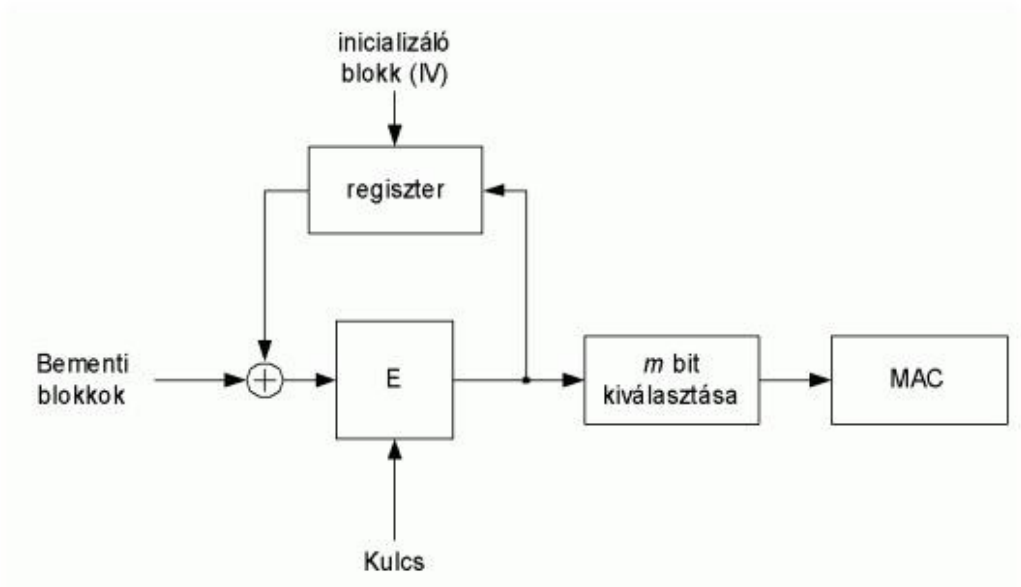
- szimmetrikus kulcsú blokk kódoló: a MAC előállítása során az üzenetet CBC módban rejtjelezzük egy blokkrejtjelezővel. Az üzenetet blokkokra vágjuk.
  A függvény így írható le:
  $X = X_1, X_2 \ldots X_n$
  $Y_1 = E_k (X_1)$
  $Y_i = E_k (X_i + Y_{i-1}), i = 1 \ldots n$
  $MAC_k(X) = Y_n,$
  ahol X az üzenet, X_i az üzenetblokkok, K a kulcs.

- lenyomatkészítő függvénnyel: mivel a MAC függvénynek szűkítő transzformációnak kell lennie, ezért hash függvényből konstruálhatunk MAC függvény, mégpedig úgy, hogy a kulcsot hozzáfűzzük az üzenethez és így készítünk lenyomatot. Attól függően, hogy a kulcsot az üzenet elejére vagy végére fűzzük, megkülönböztetünk titok prefix és titok suffix módszert. A hash

függvény tulajdonságainak köszönhetően egy tetszőleges méretű üzenetből fix hosszúságú lenyomatot képez. A lenyomatból nem lehet következtetni a kiindulási adatra, és szinte lehetetlen olyan másik dokumentumot készíteni, aminek a hash lenyomata megegyezne vele.
Formálisan:
$MAC_k = h\ (K|X)$,
ahol h a hash függvény, a K kulcs és az X üzenet pedig összefűzve van.

- kombinált módszerrel: hasonló a szimmetrikus kulcsú blokk kódolóhoz azt leszámítva, hogy E_k kódoló bemenetére az üzenetből képzett lenyomat blokkjai kerülnek az üzenet blokkjai helyett.
  Formálisan:
  $Z = h\ (K|X)$
  $Z = Z_1, Z_2 \ldots Z_n$
  $Y_1 = E_k\ (Z_1)$
  $Y_i = E_k\ (Z_i + Y_{i-1}),\ i = 1 \ldots n$
  $MAC_k(X) = Y_n$,
  ahol h a hash függvény, a K kulcs és az X üzenet összefűzve, Z az üzenet lenyomata, Z_i pedig az i-edik blokkról készült lenyomat.

- kombinált módszer kiválasztással: szintén blokk kódolót használunk CBC módban, de a MAC-et csak a számítás egy része fogja képezni.
  Nyilván ez lesz a legrövidebb (m hosszúságú) MAC a leírt módszerek közül, de nem szabad túl rövidre választani, mert azzal a sikeres támadás esélyét növeljük.



6. ábra: Kombinált módszer kiválasztással

Az SSL (Secure Socket Layer)

A mai rohanó világban az emberek egyre több időt próbálnak megspórolni azzal, hogy termékeket, szolgáltatásokat vesznek az interneten. Előnye nyilván az, hogy sokkal kényelmesebb a vásárlás, hátránya viszont a személyes kapcsolat hiánya. Nem tudjuk, hogy akitől vesszük az árut, tényleg az, akinek mondja magát; megbízható-e az internetes áruház; továbbá nem szeretnénk, ha illetéktelen személyek bizalmas információkat tudnának meg. Az SSL ebben nyújt nekünk segítséget. Az egyik legelterjedtebb protokoll, amelyet nyitott hálózatokon, például az interneten használnak. Először a Netscape Navigator böngészőben jelent meg, hogy a kliens és szerver gép közötti kapcsolatot tegye biztonságossá. Az SSL azonosítja a szervert azáltal, hogy egy digitális tanúsítvány meglétét követeli a szerver oldalon. A mai böngészők szinte mindegyike támogatja az SSL protokollt, de magunk is találkozhatunk ezzel, ha olyan oldalra tévedünk, ahol bizalmas információkat kell közölnünk. Az SSL kapcsolatot igénylő oldalak a megszokott http (Hypertext Transfer Protocol) helyett https (Secure http) betűkkel kezdődnek. A https biztosítja, hogy két számítógép biztonságosan kommunikálhasson az interneten keresztül. Az SSL a szállítási réteghez tartozik. A hibrid referenciamodellben a transzport réteg felett és az applikációs réteg alatt található. Az SSL 1.0 verziója egy tesztverzió volt, az első nyilvánosan használt verzió a 2.0 és manapság a 3.0 SSL alkalmazzák, illetve a az SSL 3.1 verzióját, ami nem más, mint a TLS.

Az SSL felépítése

Az SSL négy alprotokollból áll: Record protokoll, Handshake protokoll, Change-Cipher-Spec protokoll és Alert-protokoll. Ezek közül a Record protokoll helyezkedik el a transzportréteg legtetején, a többi három a Record protokoll és az alkalmazási réteg között van.

- Record protokoll: magasabb szintű protokollok (SSL Handshake, SSL Alert, HTTP) beágyazását teszi lehetővé, a közöttük lévő kommunikáció védelme a feladata. A Record protokoll integritás védelmet, visszajátszás elleni védelmet, titkosítás valósít meg nemcsak a protokollok között, hanem a szerver és kliens között is.

- Handshake protokoll: egyik feladata, hogy a kommunikációban résztvevő feleket azonosítsa. A szerver azonosítása mindig megtörténik, de a kliensazonosítás opcionális. További feladata, hogy a szerver és a kliens által is támogatott titkosítási algoritmust, a hozzá tartozó paramétereket egyeztesse a felek között. Nyilvános kulcsú titkosítást használ a közös kulcsok létrehozásához. Végül magát a titkosított SSL kapcsolatot hozza létre. Leggyakrabban használt algoritmusok: hash függvények: MD5, SHA; szimmetrikus titkosítás: AES, DES, RC4, Triple DES, digitális aláírás: RSA, DSA.

- Alert protokoll: a protokoll az SSL viszony alatt létrejövő problémákról értesít. Egy ilyen hibaüzenet áll magából az üzenetből és a hiba leírásából. Végzetes hiba (fatal message) esetén mindkét fél befejezi a kapcsolatot. A szabályos befejezés mindig a close_notify üzenettel ér véget, amely üzenetnek az elküldését bármely fél (kliens vagy szerver) kezdeményezheti. Viszont, ha bármely kapcsolat a close_notify üzenet nélkül ér véget az azt eredményezi, hogy az SSL viszonyt nem lehet folytatni.

- Change-Cipher-Spec protokoll: ez a protokoll egy üzenetből áll, amit rendszerint a Handshake protokoll végénél kerül küldésre. Az üzenetet mind a kliens mind a szerver elküldi, ezzel tájékoztatva a másik felet, hogy a megbeszélt titkosítási specifikációkat (algoritmus, kulcsok) kezdi el használni. Ez után, az üzenetet vevő a Handshake előtti vételi állapotról átáll az új specifikációkra. A CCS tehát egy változást idéz elő az SSL viszonyban anélkül, hogy a kapcsolatot újra kellene kezdeni. Két állapota lehet a CCS üzenetnek: read current, read pending.

Az SSL viszony (session) és a kapcsolat (connection)

A SSL protokoll használata során, két fél között több párhuzamos viszony is létrejöhet. Egy viszonyon belül pedig több kapcsolat lehetséges. A legtöbb implementáció azonban a párhuzamos viszonyokat nem támogatja, hanem az egy viszonyon belül több kapcsolat esetét alkalmazza.

SSL session

Az SSL viszonyt egyértelműen meghatározza az alábbi hat állapotváltozó:

- viszony azonosítója (session ID): egy 32 oktettből álló tetszőleges számsor, amit a szerver arra használ, hogy egy aktív viszonyt azonosítson; vagy egy olyan viszonyt, amit újra lehet aktiválni.

- a felek tanúsítványai. Az értéke =0, ha nincs tanúsítvány vagy a Fortezza algoritmus van használatban.

- tömörítő metódus: a felek megegyezhetnek adattömörítési algoritmusban. Egyelőre ezt nem használják, ezért üresen hagyott mezőt jelent.

- a titkosítás specifikációi: definiálja mely titkosítási algoritmust, illetve hash függvényt használjuk a támogatott algoritmusok és hash függvények közül.

- mestertitok (mastersecret): 48 oktett nagyságú, csak a kliens és a szerver ismeri. A mestertitok segítségével készül minden más titok (kapcsolatkulcsok), épp ezért az egész viszony alatt érvényes marad.

- végül a viszonyhoz tartozik a resumable jelzőbit, ami megmutatja, hogy a viszonyt lehet-e még új kapcsolatok nyitására használni. A jelzőbit értéke =1, ha lehet. A 0-ás bitérték azt jelenti, hogy valamely a viszonyhoz tartozó kapcsolatban végzetes hiba történt, ami automatikusan a kapcsolat bontását jelenti.

A titkosítás specifikációi

Az SSL kapcsolat alatt az úgynevezett cipher suite-t négy elem határozza meg:

- a titkosítás fajtája.

- a titkosításra használt algoritmus: ez lehet RC2, RC4, DES, AES, TDES illetve a Fortezza algoritmus. Továbbá az az opció is megvan, hogy semmilyen titkosítást nem használunk.

- hash függvényként választhatjuk: MD5 vagy inkább a manapság elterjedt SHA, vagy nem használjuk egyiket sem.

- a lenyomat mérete.

Az SSL kapcsolat

Hasonlóan az SSL viszonyhoz, egy kapcsolat állapotát is leírhatjuk változókkal, amelyek egy új kapcsolat létrehozásakor frissülnek:

- két véletlen szám egyenként 32 oktett méretűek, amelyeket a szerver (server_random) és a kliens (client_ random) egyenként hoz létre. A viszony létrejöttekor keletkeznek és minden új kapcsolat esetén.

- két titkos kulcs: server_MAC_write_secret és client_MAC_write_secret. Ezeket a kulcsokat az üzenet azonosító kód (MAC) kiszámításához, a hash függvényekben alkalmazzák. A MAC mérete így az alkalmazott hash függvénytől függ: MD5 esetén 16 oktett; SHA esetén 20 oktett.

- két kulcs az adatok szimmetrikus kódolásához. A kulcsok mérete a törvényi szabályozás, illetve a titkosító algoritmustól függ.

- továbbá két kezdőértéket megadó vektor a CBC módban történő szimmetrikus kódoláshoz. Lennie kell a szerver oldalon és a kliens oldalon is. Méretük szintén algoritmusfüggő.

- két sorozatszám (üzenet sorszáma), egy-egy a kliensnek és a szervernek 8 oktetten kódolva. Minden kapcsolathoz külön vannak nyilvántartva, és mindig eggyel növekedik, amikor az adott kapcsolatban egy üzenet elküldésre kerül.

A server_random, client_random és a sorozatszámok mind a visszajátszás (replay attack) ellen nyújtanak védelmet. Megakadályozzák a már korábban elküldött üzenetek ismételt elküldését. Minden kapcsolatnak megvan a saját kulcsa. Egy viszonyon belül azonban a kapcsolatok ugyanazt a mesterkulcsot és algoritmust használják.

Tanúsítványok (Certificates)

A nyilvános rejtjelező kulcsának tanúsítványa (Public Key Certificate) egy elektronikus dokumentum, ami digitális aláírást használ arra, hogy egy nyilvános kulcsot kapcsolatba tudjunk hozni egy személyazonossággal. Ez lehet egy név, egy szervezet neve vagy lakcím adatok és így tovább. A tanúsítványok rendszerében egy hierarchia van, ahogy az a SET protokoll leírásában megtalálható.

A tipikus információk egy digitális tanúsítványon:

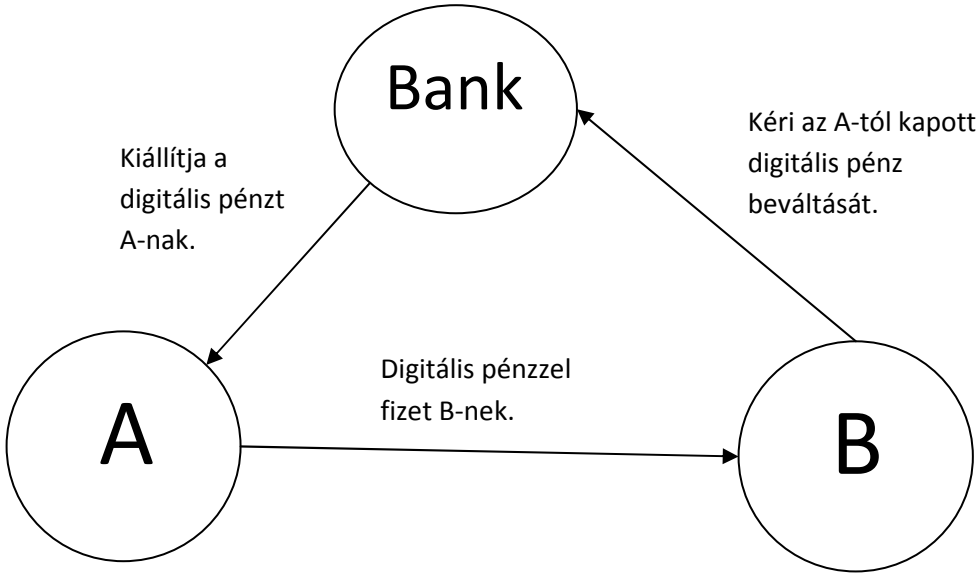
- Version: verziószám (1,2 vagy 3)
- Serial number: kibocsátó által adott egyedi sorszám
- Signature: milyen algoritmust használtak a tanúsítvány aláírásához
- Issuer: az a megbízható fél, aki aláírta a tanúsítványt
- Validity: kiadás dátuma és a lejárat dátuma
- Subject: a tanúsítvány jogos tulajdonosa
- SubjectPublicKeyInfo.subjectPublicKey: a tanúsítvány tulajdonosának nyilvános kulcsa
- IssuerUniqueID: a megbízható fél egyedi azonosítója
- SubjectUniqueID: a tanúsítvány tulajdonosának egyedi azonosítója, amit a kibocsátó féltől kapott

Fizetési rendszerek fajtái

Azt hogy hogyan, mivel, milyen módon akarjuk megvalósítani a fizetést magunk dönthetjük el. Napjainkban használatos módszerek:

Digitális pénz: DigiCash

A DigiCash[7] on-line, anonim fizetési rendszer. Az elektronikus készpénzt egy bináris szám jelenti. Ez a virtuális bankjegyünk egyedi azonosítója. A fizetés a sorszám elküldésével valósul meg. Persze a digitális pénz kicserélhetjük „igazi pénzre" miután a bank ellenőrizte annak hitelességét. A forgalomban lévő pénz sorszámát tárolni kell, hogy azt kétszer elkölteni ne lehessen. Hátránya ebből adódóan, hogy a sorszámok tárolása hosszabb használat esetén nehézkessé válhat. Előnye a többi fizetési rendszerhez képest azonban az, hogy a tranzakciók teljesen anonimen zajlanak.



7. ábra: Digitális pénzzel való fizetés modellje

A jól működő digicash rendszernek az alábbi kritériumoknak kell megfelelnie[10]:

- biztonság: megbízható titkosítás algoritmust kell használni, hogy a tranzakció során egyik fél se tudja megváltoztatni vagy reprodukálni az elektronikus tokent.
- hordozhatóság: a digitális pénz használata nem lehet helyhez kötött. A pénznek átutalhatónak kell lennie a számítógép-hálózaton, sőt más hordozható eszközökre is.

- oszthatóság: a digitális pénz címleteinek oszthatónak kell lennie, lehetőséget kell biztosítania a kibocsátónak, hogy pénzünket a legkisebb egységekre szét lehessen darabolni.

- off-line elérhetőség: nevéből adódóan egyik félnek sem kell on-line kapcsolatban lennie a tranzakció lebonyolításához. A vásárló bármikor utalhat a kereskedőnek, harmadik fél részvétele nélkül is.

- tartósság: korlátlan ideig lehessen fizetni vele, ne váljon értéktelenné a fel nem használt pénzünk.

- széleskörű használhatóság: ne csak egy adott területen lehessen használni digitális pénzünket. Ez nyilván a kibocsátó intézet hitelességének, megbízhatóságának ellenőrzését foglalja magába.

- kétoldalú rendszer: azaz ne csak a vásárló-kereskedő kapcsolat jöhessen létre, hanem két vásárló egymás között is tudjon digitális pénzfogalmat bonyolítani.

- felhasználóbarát: egyszerű legyen a fizetés és a pénzfogadás folyamata. A használóknak ne kelljen külön ismeretekkel (kriptográfiai) rendelkezniük.

Mikrofizetések

Az elektronikus fizetések egy olyan formája, amelyet nemcsak interneten keresztül alkalmazhatunk, hanem a mindennapi életben (pl. boltokban, közlekedési eszközökön) is jól működik. Általánosan elmondhatjuk, hogy az emberek többsége ma még többnyire készpénzzel szokott fizetni. Persze megjelenik a bankkártyás fizetés is, de az ilyen fajta fizetések költségei, van úgy, hogy túl sokba kerülnek. Itt tipikusan a kis összegű fizetésekre kell gondolni. A mikrofizetések ezt hivatottak kiküszöbölni. Kis összegű fizetéseket tesz lehetővé, így kikerülve azt az esetet, hogy maga a tranzakció díja drágább legyen, mint a termék, amiért fizettünk. Ha egy termék lehetővé teszi, hogy „feldarabolható” legyen például egy újság, akkor a vevő csak az őt érdeklő cikkért fizet, vagy ilyen egy népszerű albumról egy zeneszám. A mai digitális világban számos szolgáltatást lehet így darabokra osztani. Az ilyen fizetési rendszerrel a készpénzt akarják felváltani, helyette elektromos pénztárcákat

alkalmaznak smart kártyára vagy memória kártyára integrálva. Ahhoz, hogy egy ilyen rendszer jövedelmező legyen a tranzakció költségeit kell csökkenteni. Ezt úgy érik el, hogy off-line azonosítást alkalmaznak, csoportosítják a fizetéseket vagy csökkentik a számítási feladatok összetettségét a processzorokon.

PayWord

A PayWord egy mikrofizetési protokoll, amelyet 1996-ban Ronald L.Rivest és Adi Shamir dolgozott ki. Kredit alapú, off-line, azonosított rendszer, amely nyilvános kulcsú titkosítást használ. Fizetési eszközként zsetont, úgynevezett payword-öt használ. A protokollban résztvevő felek: a vásárló, a kereskedő, a bróker.

A protokoll lépései:

1. A vásárló (U) kapcsolatba lép a brókerrel egy biztonságos csatornán keresztül. Kiválaszt egy B brókert, amire B elküldi a C_U tanúsítványt számára. A tanúsítvány tartalma: $C_U=Sign_{SKB}(B, U, AU, PKU, E, IU)$. A tanúsítványt B aláírja privát kulcsával (SK_B). A tanúsítványban szerepel a bróker azonosítója (B), a vásárló azonosítója (U), vásárló címe (lehet IP cím vagy e-mail cím – AU), a vásárló nyilvános kulcsa (PK_U), tanúsítvány érvényességi ideje (E) és egyéb információk (kredit limit, tanúsítvány száma, vásárlás feltételei – IU). A tanúsítvány által a vásárló zsetonokat (payword) hoz létre a bróker jóváhagyásával. A kereskedő a C_U tanúsítvány dekódolásával (bróker nyilvános kulcsával) ellenőrzi annak tartalmát. A tanúsítvány biztosítja a kereskedőt, hogy az érvényességi időn belül kibocsátott zsetonokat B értéküknek megfelelően beváltja a kereskedő számára.

2. Fizetéshez a vásárló készít egy $\{w_1,\ldots,w_n\}$ zsetonokból álló láncot. Első lépésben választ egy w_n véletlen számot, ez lesz az n-dik zseton. Ezután egy H hash függvény (MD5 vagy SHA-1) segítségével előállítja a többi zsetont. A w_{n-1}előállítása: $w_{n-1}=H(w_n)$. A w_{n-2} előállításához másodszor is alkalmazzuk a H függvényt az előbb előállított hashértékre: $w_{n-2}=H(w_{n-1})$. A $(w_0, w_1,\ldots w_{n-1},w_n)$ lánc létrejön, miután n-szer alkalmaztuk a hashfüggvényt. A vásárló egy úgynevezett Commit üzenetet küld a kereskedőnek. Az üzenet tartalma: $M=Sign_{SKU}(V, CU, w_0, D, IM)$. Az üzenetet aláírja a vásárló privát kulcsával (SK_U). Az üzenet tartalma: a kereskedő azonosítója (V), a vásárló tanúsítványa (C_U), a hash lánc utolsó eleme (w_0), a beváltási határidő (D),

egyéb információ (lánc hossza, egy zseton (payword) értéke) – IM). Az üzenettel engedélyezi, hogy a bróker beváltsa a zsetonokat a kereskedő részére a beváltási határidőn belül. A kereskedőnek ellenőrizni kell, hogy a zsetonok értéke megegyezik-e a szolgáltatás darabjainak értékével, illetve ellenőrzi a beváltási határidőt. A C_U tanúsítványban található nyilvános kulccsal (PK_U) ellenőrzi a Commit üzenet aláírását (SK_U). Ha egy termék ára i darab zsetonból áll, akkor fizetés formálisan: $P=(w_i,i)$. A kereskedő ellenőrzi a w_i-t, hogy annak hash értéke megegyezik-e az előzőleg kapott zseton értékével, amíg $i > 1$. Ha $i = 1$ akkor a Commit-ban található w_0 értékkel hasonlítja össze. A kereskedő mindig csak az utolsónak kapott zsetont tárolja.

3. Beváltáskor a kereskedő elküldi a Commit üzenetet, illetve az utolsónak kapott zsetont (w_k,k) a brókernek, aki összehasonlítja a Commit-ban lévő w_0 értékkel (utolsó zseton hash értéke egyenlő-e w_0 értékével). Ha rendben találta, akkor leveszi a vásárló számlájáról a megfelelő (k*egy zseton értéke) összeget, ugyanennyit pedig a kereskedő számlájára helyez.

Előnye, hogy a Commit üzenetben a vásárló egy aláírással hitelesíti összes zsetonját. Természetesen a zsetonok előállítása időigényes is lehet: nagy darabszám esetén mindig meg kell hívni a hash függvényt. A kereskedőnek viszont elég egyszer meghívnia, és tranzakciónként egy aláírást kell ellenőriznie. Mindezt a kereskedő on-line teszi meg. A brókernek meg kell újítani a tanúsítványokat (havonta), ami egy digitális aláírást igényel. Meg kell hívnia k-szor a hash függvényt, igazolja minden egyes vásárló Commit üzenetét, amit azonban megtehet off-line módban is.

Elektronikus csekk

Az elektronikus csekkek használata főleg az olyan országokban elterjedt, ahol a hagyományos (papír alapú) csekkek is nagy népszerűségnek örvendenek (USA). A papír alapú csekkek széleskörű használata miatt, itt van igény egy kényelmesebb elektronikus csekk használatára.

Az elektronikus csekkek azonosítása alapvetően két kategóriába esik:

- a kedvezményezett bankja csak azokat az információkat küldi a kibocsátó banknak, amik alapján be tudják azonosítani a csekket.

- a kedvezményezett bankja a csekk mindkét oldaláról egy szkennelt képet küld.

Az első esetben a csekken lévő adatokat rögzítik, és elektronikus formában tárolják el. Ezáltal a csekkek elszámolásához tartozó költségek csökkenek, de a bankok közötti biztonságos kapcsolat igénye megnövekedik. Például Franciaországban a csekkhez tartozó információkat a CMC7 sor adja, ami alfanumerikus mágneses karakterekből áll (szabvány írja elő). Az USA-ban ennek a sornak a MICR (Magnetic Ink Character Recognition) felel meg.

A második esetben digitális képek készülnek a csekkről, amit elektronikusan továbbítanak a bankok. A biztonságos továbbításon felül az ilyen megoldások képtömörítési algoritmusokat is igényelnek.

Amerikában kifejlesztettek egy úgynevezett SafeCheck rendszert. Ennek a lényege, hogy a vásárló valós időben (a vásárlás helyszínén) tudja hitelesíteni a csekkjét. Természetesen ez speciális eszközt igényel. A pénztáros beüti a vásárlás összegét, közben az eszköz leolvassa a MICR adatokat. Ezeket aztán elküldi a vásárló bankjának, ami ellenőrzi a fizetőképességet és jóváhagyja a tranzakciót.

NetCheque

A NetCheque[9] eredetileg egy egyetemi rendszerben, erőforrásokhoz való hozzáférések menedzselésére (pl. memória menedzselés) találták ki. Aztán ezt a koncepciót az elektronikus csekkek használatával egy fizetési rendszerré bővítették. A NetCheque-ben egy kerberos[5] szerver végzi a felek azonosítását (kétféle azonosítást használ: Ticket és Authenticator), illetve az úgynevezett viszonykulcsok (session key) előállítása és szétosztása a feladata. Három viszonykulcs szükséges: a vevő és a bankja (B1) között: K_{CB}; a kereskedő és az ő bankja (B2) között: K_{MB}; a két bank között: K_{BB} (általában a kerberos szerver elkülönül a két banktól). A felek számlát nyitnak a saját bankjuknál, majd elektronikus csekkeket kapnak. A NetCheque az alábbi mezőket tartalmazza:

- az összeg megnevezése
- a pénznem
- csekk dátuma
- a vevő számlaszáma
- kedvezményezett neve

- számlatulajdonos digitális aláírása
- a kedvezményezett és a bankok jóváhagyása

Az utolsó két mezőt a vásárló bankja hitelesíti. A vevő kiállítja a csekket, ezzel tudja a kedvezményezett az adott összeget lehívni a vevő számlájáról. Mind a vevő, mind a kereskedő egy (session ticket) igazolvánnyal azonosítja magát a saját bankjának (T_{CB}: a vevő és a B1 bank; T_{MB}: a kereskedő és B2 bankja), illetve T_{BB}-vel a B2 azonosítja magát a B1 banknak.

Az igazolványok tartalma:

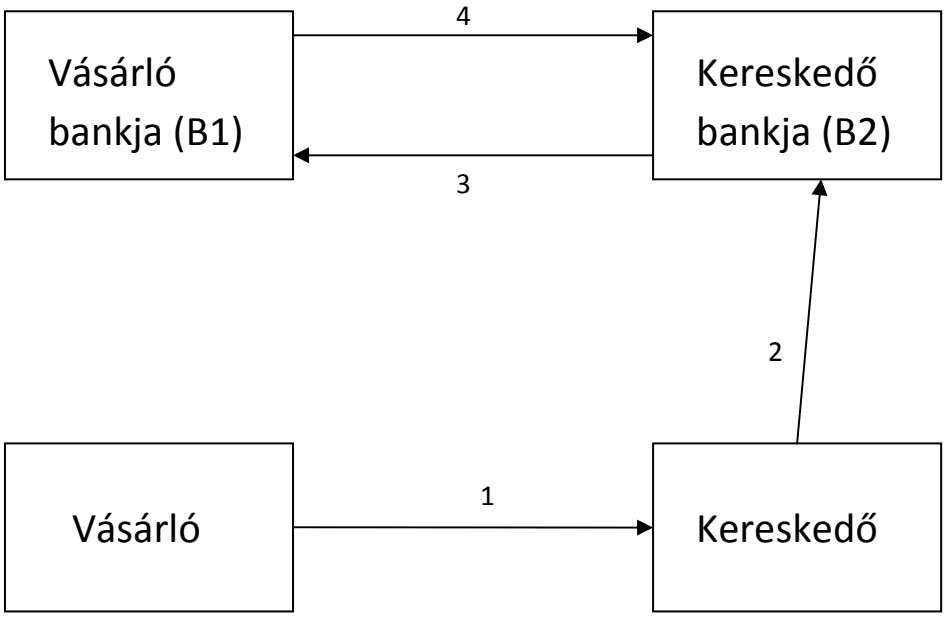
- A vásárló igazolványa: $T_{CB}=\{S, K_{SB1}(C, B1, K_{CB})\}$. Az S a kerberos szerver azonosítója, C a vásárló neve, B1 a bank címe és a K_{CB} viszonykulcs. A szerver azonosítóját kivéve a többi adat K_{SB1} kulccsal (a kerberos szerver és a vevő bankja között) titkosítva van.
- A kereskedő igazolványa: $T_{MB}=\{S, K_{SB2}(M, B2, K_{MB})\}$. M a kereskedő neve, B2 a bankjának címe, K_{MB} viszonykulcs, az S ugyanaz, mint az előbb. Itt az S-t kivéve a többi adat titkosítva a K_{SB2} kulccsal.
- $T_{BB}=\{S, K_{SB1}(B2, B1, K_{BB})\}$.

A fizetés folyamata:

1. A vásárló elkészíti a hitelesítőt ($AuthC= K_{CB}\{H(Check)\}$), amiben a vásárló azonosítója, a számlaszáma és a csekk lenyomata van, titkosítva az egész a K_{CB} viszonykulccsal. Ehhez hozzáfűzi az igazolványát (T_{CB}) és így küldi el a kereskedőnek a csekket: $Check \parallel AuthC \parallel T_{CB}$.
2. A kereskedő a T_{MB} igazolvánnyal biztonságos kapcsolatot hoz létre bankjával. A csekk jóváhagyásához elkészíti ő is a hitelesítőt ($AuthM= K_{MB}\{H(Check)\}$), majd a vevőtől kapott üzenethez hozzáfűzi saját hitelesítőjét, és úgy küldi a bankjába: $Check \parallel AuthC \parallel T_{CB} \parallel AuthM \parallel T_{MB}$.

3. A kereskedő bankja jóváhagyja a csekket az $AuthB=K_{BB}\{H(Check)\}$ hitelesítővel. Hozzáfűzi a T_{BB} igazolványt és küldi a vásárló bankjába: $Check \| AuthC \| T_{CB} \| AuthB \| T_{BB}$.

4. A vevő bankja végül ellenőrzi az adatokat: az igazolványokat és a hitelesítőket (a vásárló és a kereskedő bankja hiteles-e), majd a K_{CB} és K_{BB} kulcsokkal dekódolja a hitelesítőket, válaszol a kereskedő bankjának.



8. ábra:Netcheque működési modellje

Fizetési rendszerek osztályozása

	On-line rendszerek	Off-line rendszerek
Azonosított	First Virtual, CyberCash, iKP, NetBill, MiniPay	Danmont/Visa, CLIP, Mondex
Anonim	NetCash, e-cash	CAFE

E-mail alapú fizetési rendszer: Paypal

A PayPal[12] egy olyan elektronikus fizetési rendszer, ami e-mailen alapul. A vevő és az eladó között a PayPal a harmadik megbízható fél szerepét tölti be. Tipikus C2C rendszer. A vevő és eladó személyes információit csak PayPal tudja, tranzakciók során a résztvevő felek közül egyik sem láthatja a másik bizalmas adatait. Azért terjedt el, mert igen egyszerű a kezelése, ezért internetes aukciók során (eBay) előszeretettel használják. Ahhoz, hogy pénzt küldhessünk vagy fogadhassunk az alábbiak szükségesek:

- érvényes e-mail cím
- érvényes bankkártya vagy bankszámla

A banki adatokat a PayPal bizalmasan kezeli, SSL titkosítással védi. Az e-mail cím lesz az egyedi azonosító (PayPal ID). Ehhez tartozik továbbá egy jelszó is. A személyes adatok közül csak az e-mail és a szállítási cím látható az eladó számára. Pénzünket utalhatjuk bankszámláról (2 munkanapon belül történik) vagy közvetlenül PayPal számlánkról (ha ott tartunk pénzt). Lehetőség van visszautalni pénzünket a bankba is, ami ingyenes, de akár 8 munkanapot is igénybe vehet. A PayPal tehát közvetít a vevő és az eladó között, és ezért némi díjat számol fel. A fizetés ingyenes, de ha pénzt kapunk, akkor van egy tranzakciós díj ($0.3 USD), plusz a kapott összeg nagyságának megfelelően egy százalékot (1.9% – 2.9% között) számol fel.

Fizetés menete:

1. Miután kiválasztottuk a terméket, rákattintunk a „fizetés PayPal-el” gombra, ekkor az összeg levonásra kerül a PayPal-es számlánkról vagy arról a bankkártyáról kezdeményez fizetést, ami a PayPal számlához tartozik.
2. Ezután az eladó számlájára jóváírja az összeget (mínusz a tranzakciós díj), amit a vevő számlájáról levont.
3. Utolsó lépésben e-mailt küld a PayPal a vásárlónak és az eladónak is, amiben visszaigazolja a tranzakciót és a tranzakció összegét.

Kártya alapú fizetés (SET – Secure Electronic Transaction)

A biztonságos kártyás fizetést teszi lehetővé az interneten, amelyet a Visa és a MasterCard dolgozott ki. Az SSL-nél biztonságosabb, mert a kommunikáló feleket is azonosítja tanúsítványok segítségével. Továbbá biztosítja a tranzakcióban résztvevő adatok bizalmasságát, a tranzakcióban résztvevő adatok integritását. Ezen tulajdonságokat nyilvános kulcsú titkosítással, digitális borítékkal, kettős aláírással éri el.

Nyilvános kulcsú infrastruktúra

A SET protokoll résztvevői: a vásárló, a kereskedő és a banki interfész. Mind a három résztvevőnek van publikus és titkos kulcsa. A SET nyilvános kulcsú infrastruktúrára támaszkodik, amelyen belül a hierarchia megfigyelhető. A hierarchia csúcsán a gyökér hitelesítő (Root CA – Certificate Authority), alatta az egyes bankkártya márkák hitelesítői (Brand CA), ezek alatt a földrajzi hitelesítő szerverek (Geopolitical CA) végül alattuk a vásárlók, kereskedők, bankok.

Digitális boríték

A digitális boríték egy olyan technika, amely ötvözi a nyilvános kulcsú és a szimmetrikus kulcsú titkosítást. Először az üzenetet titkosítjuk szimmetrikus kulcsú titkosítással, majd a szimmetrikus kulcsot rejtjelezzük a címzett nyilvános kulcsával. Az így létrejött kulcs és üzenet lesz a digitális boríték. Előnye, hogy a nyilvános kulcsú titkosítást csak a szimmetrikus kulcsra alkalmazzuk (az egész üzenet helyett), ezáltal kiküszöbölve a nyilvános kulcsú titkosítás lassú tulajdonságát.

Kettős aláírás

A kettős aláírás célja, hogy A két dokumentumot (M1 és M2) tudjon egyszerre aláírni úgy, hogy M1-t csak B1, M2-t csak B2 tudja elolvasni, ahol B1 és B2 a címzettek. Ehhez A képezi a két üzenet hash értékét külön-külön, majd az ezekből képzett hash értéket írja alá. Formálisan: $Sign_{SKA}(h(h(M1) \| h(M2)))$.

B1-nek elküldi az $M1 \| h(M2) \| Sign_{SKA}(h(h(M1) \| h(M2)))$, így B1 megkapja a neki szánt üzenetet (M1),továbbá A aláírását is tudja ellenőrizni az M2 hash értékének segítségével, de a M2 üzenetet nem tudja elolvasni. B2 pedig: $M2 \| h(M1) \| Sign_{SKA}(h(h(M1) \| h(M2)))$

üzenetet kap, ahol hasonlóan az előzőhöz a neki szánt üzenetet el tudja olvasni, az aláírást tudja ellenőrizni, de az M1 üzenetet nem tudja elolvasni.

A SET protokoll lépései

1. a vásárló az interneten belép a kereskedő oldalára, majd ha ott eldöntötte, hogy a web-shopból mit akar megvenni, kattint a „vásárlás” gombra.

2. ezt követően a vásárló kitölti a szükséges adatokat és elküldi, hogy milyen típusú bankkártyát kíván a vásárláshoz használni.

3. a kereskedő fogadja az adatokat és hozzákapcsol egy tranzakció-azonosítót (TRID), amit digitálisan aláír. Az aláírt TRID-en kívül válaszában elküldi az aláírás-ellenőrző kulcsának tanúsítványát, a nyilvános rejtjelező kulcsának tanúsítványát és a banki interfész tanúsítványát.

4. Vásárlási kérelem: a vásárló szoftvere ezután előállítja a rendelési információkat (OI) és a fizetési utasításokat (PI). Mivel a PI bizalmas adatokat tartalmaz, ezért a kereskedő ezt nem láthatja, de PI-t és OI-t össze kell kapcsolni egy tranzakció-azonosítóval, hogy később is tudják mely vásárláshoz, mely fizetési utasítások tartoztak. Ezt a digitális boríték + kettős aláírás technikájával valósítja meg. A rendelési információkat ellenben el tudja olvasni.

5. Engedélyezés kérés: következő lépésben a kereskedő elküldi a banki interfésznek a vásárlótól kapott PI adatokat (amiket ő nem lát, ezért kéri majd a bank közbenjárását) a vásárló és saját tanúsítványával együtt. A banki interfész saját kulcsával kibontja digitális borítékot, ellenőrzi a tanúsítványokat. Leellenőrzi, hogy a kereskedő által használt TRID azonos-e a fizetési utasításban szereplővel. Az interfész elküldi PI-t a vásárló bankjának, amely engedélyezi a tranzakciót, ha a vevő fizetőképes.

6. Engedélyezés kérésre válasz: ha a bank engedélyezi a tranzakciót, akkor az interfész aláírja, majd digitális borítékban elküldi válaszát a kereskedőnek. Az üzenet tartalmazhat egy lehívási zsetont. Később a kereskedő kifizetésénél lesz jelentősége.

7. Vásárlási kérelemre adott válasz: a kereskedő egy nyugtázó üzenetet küld a vásárlónak, amit aláírt. A vásárló ezt megőrzi. A kereskedő elküldi a terméket vagy a szolgáltatást a vásárló rendelkezésére bocsátja.

8. A kereskedő végül egy lehívási kérelemmel kaphatja meg pénzét. Ez a kérelem tartalmazza a tranzakció adatait, TRID-et, fizetendő összeget. A kérelmet aláírja, digitális borítékban elküldi a banki interfésznek a korábbi lehívási zsetonnal együtt.

9. A banki interfész ellenőrzi a lehívási zsetont és a hozzá tartozó kérelmet. Ha mindent rendben talál a vásárló bankjánál kezdeményezi az összeg átutalását.

10. Végül nyugtát küld a kereskedőnek a banki interfész, amit ő megőriz a tranzakció lezajlásának igazolásához.

SSL alapú elektronikus fizetés lebonyolítása

A fizetés során a lépéseknek adott sorrendje van, nincsenek párhuzamos lépések és a rákövetkező lépés csak akkor kezdődhet, ha a megelőző már befejeződött. Hozzá kell tenni, hogy alábbiakban tárgyalt protokoll a Magyarországon használatos séma[8].

1. A Vásárló belép a Bolt honlapjára. Ha a honlapon regisztráció/bejelentkezés szükséges, akkor ennek elvégzése után a Vásárló hozzáférhet a termékek adataihoz, böngészhet a webshopban. Ha kiválasztotta a termék(ek)et, rákattint a „fizetés" opcióra. Bizonyos boltok rendelkeznek SSL kapcsolattal (lásd. SSL.), de ez opcionális.

2. A második lépésben a Bolt adatokat küld a Banknak. A vásárláshoz tartozó adatok: melyik termékből mennyit rendelt a vásárló; milyen színű, milyen méretű a termék… stb. egy adatbázisban tárolódnak. Ehhez az adatokhoz hozzárendelünk egy egyedi azonosítót (Transaction ID), amivel a tranzakciót egyértelműen azonosítani lehet. Az így létrejött tranzakció azonosító (TRID), a tranzakció összege és a felhasználó neve kerül a Bankhoz. A Bank a vásárláshoz tartozó adatokról nem kaphat információt, csupán az előbb említett adatokat kaphatja meg. Ha a Bolt nem regisztrálja a vevőket, akkor egy fiktív felhasználónevet küld. A Bank és a Bolt között TDES titkosítás valósul meg, ami egy korábban végrehajtott kulcscserét is jelent. Így történik meg a Bank-Bolt közötti azonosítás.

3. A bank nyugtázza a tranzakciókérést és visszajelez a boltnak. Ha a Bank sikertelen tranzakciót állapít meg, akkor a Vevő értesítést kap erről a Bolttól és a vásárlás befejeződik, vagy a Vevő újra indítja a tranzakciót. Sikeres tranzakciókérés esetén a Bolt a vevőt a tranzakció azonosítóval együtt a Bankhoz irányítja. A vevő-bolt kapcsolatot felváltja egy sok esetben biztonságosabb vevő-bank kapcsolat. Itt már kötelező az SSL kapcsolat. A Bank oldalán a biztonságos kapcsolat folytatódik. A Vevő tranzakciós azonosítója alapján a Bank kikeresi az ehhez tartozó adatokat: bolt neve, fizetendő összeg, egyéb információk. Rendszerint egy nyomtatványt kell kitölteni, ahol megadjuk a bankkártyánk számát, ha a fenti adatokat rendben találtuk. Ezek a személyes adatok semmiképp sem kerülhetnek a Bolthoz, vagy egyéb

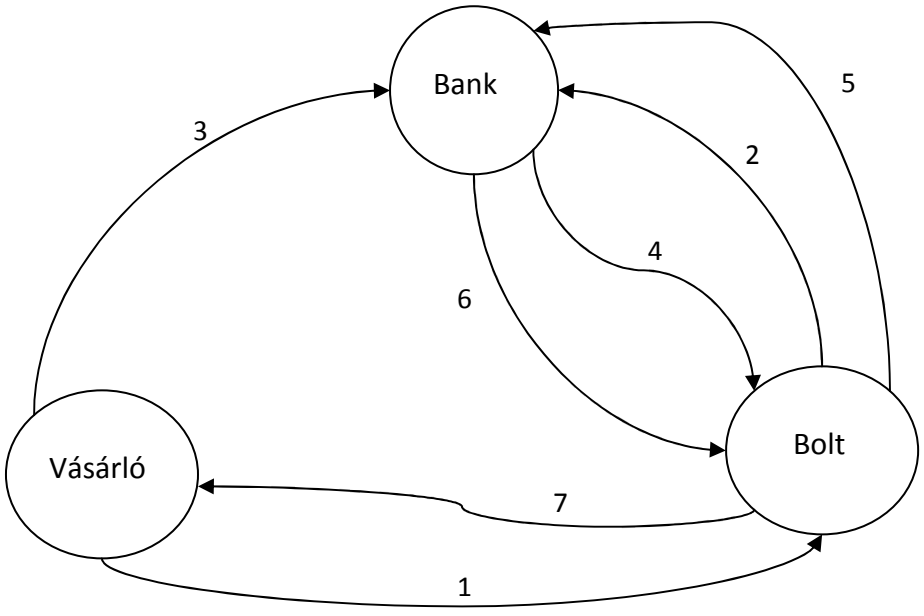
harmadik személyhez. Ha nem találtunk rendben valamit, akkor még kérhetjük a tranzakció megszakítását. A Vevő jóváhagyása után a Bank elvégzi az authorizálást.

4. Ezt követően a Bank visszaküldi a Boltba a Vevőt csatolva a tranzakciós azonosítóval, ami alapján a Bolt azonosítja, és kikeresi adatbázisából a tranzakcióhoz tartozó adatokat.

5. A Bolt kapcsolatba lép a Bankkal és elkéri a tranzakciós azonosítóhoz tartozó authorizációs eredményt. Amennyiben ezt a lépést nem teszi meg adott időn belül, úgy a Bank visszavonja a Vevő számláján a foglalást.

6. A Bank lezárja a tranzakciót miután közölte annak eredményét a Bolttal.

7. A Bolt pedig a Vevővel közli a tranzakció eredményét.



9. ábra: SSL alapú tranzakció lépései

Egy javasolt fizetési protokoll lépései

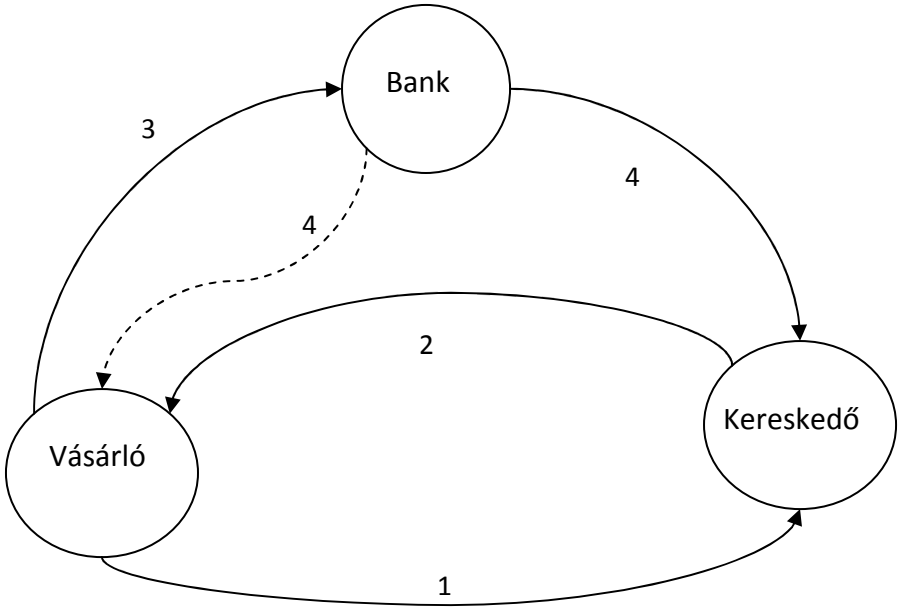
1. A vásárló megkeresi a kereskedő internetes boltját és kiválasztja a terméket. Az oldalon azonosítja magát egy felhasználónévvel és a hozzá tartozó jelszóval.

2. A kereskedő elküld neki egy C_K tanúsítványt, ami tartalmazza a kért termék árát (Ö), a rendelés azonosítóját (OID), a vásárló nevét (V), a kereskedő nevét és bankszámlaszámát (K), illetve a kereskedő aláírását. A kereskedő saját kulcsával (SK_K) aláírja a tanúsítványt, ezzel hitelesítve önmagát. Formálisan: $C_K = Sign_{SKK}(V, K, OID, Ö)$.

3. A vásárló megkapja C_K tanúsítványt. Ellenőrzi a tanúsítványon a kereskedő aláírását a hozzá tartozó nyilvános kulccsal (PK_K), ezáltal megbizonyosodik, hogy nem egy fiktív webshopról van szó, majd a vevő kiegészíti a tanúsítványt a bankkártya információkkal (B). A bankkártya információkat tikosítja a bank nyilvános kulcsával (PK_B). A tanúsítványt aláírja digitálisan saját titkos kulcsával (SK_V), majd elküldi bankjának. Az elküldött tanúsítvány: $C_V = Sign_{SKV}(C_K \| Enc_{PKB}(B))$.

4. A bank ellenőrzi a vevő aláírását, annak nyilvános kulcsával, így hitelesítve őt. A banknak szánt adatokat (bankkártya száma) saját titkos kulcsával dekódolja (SK_B). A C_K tanúsítványt szintén ellenőrzi a kereskedő nyilvános kulcsával (PK_K), majd a vásárló számlájáról átutal a C_K-ban lévő összeget a kereskedő számlájára. Ha a vevő rendelkezik a számláján a kért összeggel, akkor a bank végrehajtja a tranzakciót. Ezt követően visszajelez a kereskedőnek. Küld számára egy tanúsítványt (C_B). A tanúsítvány tartalmazza a kereskedő nevét, számlaszáma (K), a tranzakció azonosítóját (OID), továbbá a tranzakció összegét (Ö). A tanúsítványt aláírja saját titkos kulcsával (SK_B): $C_B = Sign_{SKB}(K, OID, Ö)$. Opcionálisan értesítheti a vevőt is, hogy a kért összeget levonták a számlájáról.

5. Utolsó lépésben, ami nem tartozik már szervesen a protokollhoz, a kereskedő megkapja a bank tanúsítványát. Ellenőrzi a bank aláírását, majd megőrzi a tanúsítványt későbbi reklamáció esetére.



10. ábra: Javasolt protokoll lépései

A protokoll jellemzése

Ahogy az előbbi leírásból kiderül a protokoll működése tanúsítványokon alapszik, azaz nyilvános kulcsú infrastruktúrára támaszkodik akárcsak a SET protokoll. A tanúsítványokat ezért mind a kereskedőnek, mind a vevőnek meg kell újítani bizonyos időközönként. Természetesen ez a kritérium a bankra is vonatkozik.

A másik fő összetevője a tanúsítványok mellett a digitális aláírások. Reklamáció esetén ezek segítségével lehet eldönteni a vitát, mivel bárki ellenőrizheti az aláírást, de egyben letagadhatatlanságot is biztosít az alábbi esetekben:

- ha a kereskedő nem küldi a terméket: a vevő saját tanúsítványa (C_V) segítségével, a bankon keresztül bizonyítani tudja, hogy a kereskedő próbál csalni, hiszen a C_V tartalmazza a kereskedő tanúsítványát is (C_K), amit ő digitálisan aláírt.

- ha a vevő fizet, de a bank próbálna csalni: amíg a kereskedő nem kapja meg a bank tanúsítványát (C_B), addig nem küldi a terméket. Ellenben, ha megkapta a C_B-t, a pénzt viszont nem, akkor a bank csalhat, amit viszont a C_B-n lévő digitális aláírás (SK_B) segítségével a kereskedő bizonyítani tud bárkinek.

- ha a vevő nem fizet: ezt a bank könnyen le tudja ellenőrizni, a C_V tanúsítvány ugyanis tartalmazza a vevő digitális aláírását, ezzel hitelesítve a tranzakciót.

A két protokoll összehasonlítása (SSL alapú – Javasolt Protokoll)

A protokollok összehasonlításához az alábbi kriptográfiai szempontokat használhatjuk fel:

- azonosítás az SSL alapú fizetésnél: a bank azonosítja a kereskedőt a második lépésben szimmetrikus kulcsú titkosítással (TDES). A vásárló nem azonosítja a kereskedőt közvetlenül, hanem bízik a bankban. A bankot majd a harmadik lépésben (SSL kapcsolatban) a tanúsítványa alapján tudja azonosítani a vevő. A vásárlót a bank nem azonosítja az SSL kapcsolatban, csupán kártyainformációit ellenőrzi, ezért egy eltulajdonított kártya esetén, akárki kezdeményezheti a tranzakciót.

- azonosítás a javasolt protokollnál: a kereskedőt az előző protokollhoz képest nem csak a bank azonosítja, hiszen a második lépésben először a vevő győződik meg a kereskedő hitelességéről. Itt ellenőrzi a kereskedőtől kapott C_k tanúsítványt. Ezután nyugodt szívvel aláírja a tanúsítványt, amivel aztán a bank tudja azonosítani a vevőt, ami a kártyaeltulajdonítás ellen védelmet jelent. A bank szintén azonosítja a kereskedőt C_K segítségével. Végül a kereskedő azonosítja a bankot C_B tanúsítvány segítségével.

- integritás biztosítása: SSL kapcsolatnál a bank és a vevő között, a vevő adatainak integritásának biztosítása a MAC segítségével. A kereskedő adatainak integritása TDES alkalmazásával.

- integritás a javasolt protokollnál: az adatok integritását mind a vevő, mind a kereskedői oldalon a digitális aláírás biztosítja.

- bizalmasság SSL kapcsolatnál: az általános elvárásokat (lásd. bizalmasság) a protokoll megvalósítja. A TRID segítségével a bankok nem jutnak információhoz a terméket illetően, megvalósul a belső támadás elleni védelem. Az SSL kapcsolat a vevő és a bank között biztosítja, hogy a kereskedő vagy egyéb illetéktelen személy a vevő banki információihoz ne jusson hozzá.

- bizalmasság a javasolt protokollnál: hasonlóan a TRID-hez itt az OID biztosítja, hogy a bank a termékhez kapcsolódó információkhoz ne jusson hozzá. A külső támadás elleni védelem pedig azáltal van biztosítva, hogy a kereskedőhöz eleve nem jutnak el a vevő banki információ, hanem azok nyilvános kulcsú titkosítással egyből a bankhoz jutnak a vevőtől.

- rendelkezésre állás: mindkét esetben a banknak elérhetőnek kell lennie, hiszen amíg nem kapja meg a kereskedő a visszaigazolást a tranzakcióról, addig a vevő sem kapja meg termékét.

- hatékonyságuknál meg lehet említeni, hogy habár a javasolt protokoll használ lassabban működő nyilvános kulcsú titkosítást, ezt csak kevés biten (banki információkon) kell elvégezni.

További különbségek a két protokoll között:

- a javasolt protokoll rövidebb, mint az SSL alapú. Abban is különbség van, hogy egy sikeres tranzakció lebonyolításához - a javasolt protokoll esetében - a vevőnek sokkal több dolga akad, míg az SSL alapú változatnál inkább a kereskedő végzi a lépések nagy részét. Röviden a SSL alapúnál a kereskedő-bank viszony a meghatározó, azonban a javasoltnál inkább dominál a vevő-bank viszony.

- a bank SSL tanúsítvány szükséges a SSL alapúnál, míg a javasoltnál a banknak ez nem szükséges.

- ehelyett a javasolt protokollhoz szükséges vásárlói tanúsítvány, hozzá egy kliensprogram, amivel a vevő digitális aláírást készít. Ennek megfelelően a kereskedőnek és a banknak is rendelkeznie kell ezekkel, habár ezek csak vevői oldalon jelentenek némi kellemetlenséget (megvenni, megtanulni a kliensprogram használatát).

A kifejezetten kártyás fizetésekre kidolgozott SET protokollal összehasonlítva, azt tapasztaljuk, hogy biztonságukat tekintve mindkettő teljesíti a követelményeket. A kliensprogram itt is szükséges, akárcsak a javasolt protokollnál, azonban a SET protokoll is

több lépésből áll a javasolt protokollnál, és ahogy azt a táblázat is mutatni fogja, jóval költségesebb is.

	SSL alapú	Javasolt protokoll	SET
Digitális aláírás	-	3	8
Tanúsítvány ellenőrzés	1 vagy 2	4	8
Szimmetrikus kulcsú titkosítás	4	-	6
Nyilvános kulcsú titkosítás	-	1	6
Kulcscsere	1 vagy 2	-	-

11. ábra: Felhasznált kriptográfiai eszközök a két protokoll és a SET esetén

Összefoglalás

A dolgozatból megismerhettük, hogyan biztosítják a manapság használatos elektronikus fizetési rendszerek a megismert kriptográfiai eszközök segítségével a biztonságos kereskedelmet. Ezután a napjainkban elterjedt és még nem annyira elterjedt különböző fizetési rendszereket ismerhettük meg aszerint, hogy mi a működésük alapja:

- digitális pénz alapú – ami a hasonló a hagyományos pénzhez, csak itt a bankók egy-egy egyedi sorszámok, amit majd aztán készpénzre vált nekünk a bank.
- digitális csekk alapú – ami a hagyományos csekket hivatott felváltani az olyan országokban, ahol az ilyen módú fizetés széles körben használt pl. USA.
- mikrofizetési rendszerek – az internet elterjedésével jelent meg, abból az elgondolásból, hogy miért fizetne az ember az egész termékért, ha annak csak egy részére van szüksége.

- kártya alapú – elektronikus pénztárca megjelenése a kártyákon, amelyek a hagyományos pénztárcát váltották fel újratölthető vagy csak egyszer használatos változatban (gondoljunk csak a telefonkártyára)

- e-mail alapú – igen népszerű manapság az emberek között a PayPal, mert egyszerű és biztonságos kereskedelmet valósít meg egy megbízható harmadik félen keresztül.

- SSL alapú fizetés – manapság a legelterjedtebb fizetési módszer biztonsága és egyszerű megvalósítása miatt.

A javasolt protokollt ismertetése után összehasonlítottuk a már létező SSL alapú protokollal, majd megvizsgáltuk annak előnyeit és hátrányait. Végül összefoglaltuk ezeket, és levontuk a következtetést, miszerint a javasolt protokoll lehet, hogy drágább, de kevesebb lépésből áll, biztonságosabb a sok digitális aláírás miatt, de ugyanakkor kliensprogram használatát igényli. Ha azonban a SET protokollhoz viszonyítjuk, akkor annál olcsóbb és kevesebb lépésből áll, ennek ellenére biztosítja a szükséges azonosításokat, ezáltal kielégítve az alapvető kriptográfia szempontokat. Így sikerült egy olyan protokollt alkotni, amely a kettő között helyezkedik el.

Irodalomjegyzék

Könyvek

1. Buttyán Levente, Vajda István: Kriptográfia és alkalmazásai, Typotex, Budapest, 2005.

2. Ködmön József: Kriptográfia, az informatikai biztonság alapjai, ComputerBooks Könyvkiadó, Budapest, 1999.

3. L. Jean Camp: Trust and Risk in Internet Commerce, MIT Press, 2000.

4. Mostafa Hashem Sherif: Protocols for Secure Electronic Commerce, CRC Press, Boca Raton, FL, USA, 2nd edition, 2003.

Cikkgyűjtemény

5. G. Steiner, C. Neuman, JI Schiller, Kerberos: An authentication service for open network systems, Proc. USENIX Winter Conf, 1988, pp.191-202.

6. N. Asokan, P.A. Janson, M. Steiner, M. Waidner: The state of the art in electronic payment systems, Computer 30 (9), 1997, pp. 28–35.

7. S. Godin: Presenting Digital Cash. Sams, Indianapolis, 1996, p.99-106, 226-229.

8. CIB Bank Zrt.: SSL Alapú Kártyatranzakciók az Interneten, Internetes kártyaelfogadás szolgáltatás technikai dokumentációja.

Internetes források

9. B. Clifford Neuman and Gennady Medvinsky: Requirements for Network Payment: The NetCheque Perspective In Proceedings of IEEE COMPCON'95, 1995. http://clifford.neuman.name/papers/pdf/9503_netcheque-neuman-medvinsky-compcon95.pdf

10. Jon W. Matonis: Digital cash & Monetary Freedom, 1995. http://libertarian.co.uk/lapubs/econn/econn063.pdf

11. National Bureau of Standards, Data Encryption Standard, FIPS-Pub.46. National Bureau of Standards, U.S. Department of Commerce, Washington D.C., January 1977. http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf

12. PayPal: http://netforbeginners.about.com/od/ebay101/ss/paypal101.htm, http://www.dummies.com/how-to/content/finding-out-how-paypal-works.html

13. Ronald L. Rivest and Adi Shamir: PayWord and MicroMint: Two Simple Micropayment Schemes, 2001 http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.70.9003&rep=rep1&type=pdf

14. SSL: http://www.pierobon.org/ssl/ch2/sslosi.htm

Ábrajegyzék