Kriptográfiai protokollok formális vizsgálata a CSN logikai rendszer bővítésével

doktori (Ph.D.) értekezés

Takács Péter

Témavezető: Prof. Dr. Pethő Attila

Debreceni Egyetem
Természettudományi Tudományterületi Doktori Tanács
Informatikai Tudományok Doktori Iskola

Debrecen, 2009

Ezen értekezést a Debreceni Egyetem Természettudományi Tudományterületi Doktori Tanács Informatikai Tudományok Doktori Iskola Digitális kommunikáció programja keretében készítettem a Debreceni Egyetem doktori (PhD) fokozatának elnyerése céljából.
Debrecen, 2009. október 15.

.....
Takács Péter
jelölt

Tanúsítom, hogy Takács Péter doktorjelölt 2004 - 2009 között az Informatikai Tudományok Doktori Iskola Digitális kommunikáció programjának keretében irányításommal végezte munkáját. Az értekezésben foglalt eredményekhez a jelölt önálló alkotó tevékenységével meghatározóan hozzájárult. Az értekezés elfogadását javasolom.

Debrecen, 2009. október 15.

.....
Prof. Dr. Pethő Attila
témavezető

Tartalomjegyzék

1. fejezet

Bevezetés

A vezeték nélküli hálózatok és a mobil számítástechnika alapjaiban változtatta meg a felhasználók informatikai lehetőségeit. Ez a folyamat ma is tart, kiegészülve a beágyazott rendszerek még jelentősebb előretörésével. Az új technológiai szemléletmód angolul a *pervasive computing* (*mindent átható számítástechnika*) és az *ubiquitous computing* (*ubicomp*, *mindenhol jelenlévő számítástechnika*) nevet kapta, magyarul pedig kezd elterjedni a *rejtett számítástechnika* elnevezés.

A nevek mögötti tartalom korábbra, egészen az 1990-es évekre vezethető vissza, amikor M. Weiser, a Xerox cég vezető fejlesztője dolgozott ki olyan termék-prototípusokat, amelyek egy új megközelítését tükrözték a számítástechnikai eszközöknek. [154] Az elgondolás legfőbb eleme az, hogy *,,az információ technológiának úgy kellene beleívódnia az emberek hétköznapjaiba, ahogyan azt ma az írás teszi ...”* [28]

Az elképzelés szükséges technikai elemei (olcsó, kis fogyasztású, mobil eszközök; alkalmas hálózati összeköttetés lehetősége; osztott működést támogató szoftver elemek) csak napjainkra váltak megfelelő mértékben elérhetővé. A mai fejlettebb eszközök szinte mind tartalmaznak számítástechnikai támogatást. Sok esetben nem tudatosul bennünk, hogy a használt eszközök megnövekedett hatékonysága, újabb képességei a háttérben működő *rejtett számítástechnikai* megoldásoknak köszönhetők. Ténylege-

sen fedve maradnak a megoldás részletei, számunkra a megfelelő felhasználás elsajátítása a feladat.

A szakemberek számára viszont újabb és újabb kihívásokat és megoldásra váró problémákat jelentenek ezek az új eszközök. Biztosítani kell a felhasználók által már megszokott vagy elvárt funkciók megvalósítását. Így van ez az eszközök kommunikációjánál is. Alapvető feladat a hibamentes adatcsere, a megkívánt helyzetekben a védett és titkosított kommunikáció. Ennek megvalósítása vezeték nélküli, mobil eszközök között viszont egészen más megoldásokat és tevékenységeket követel akár a felhasználóktól, akár a fejlesztőktől. Újabb algoritmusokat és protokollokat kell kidolgozni és elsajátítani az elérni kívánt célok teljesítéséhez.

Értekezésünk témája kapcsolódik a fenti irányvonalhoz, a kriptográfiai protokollok formális ellenőrzésének oldaláról közelíti meg a területet. A bemutatott munka két nagy részre osztható.

Az első egy speciális kriptográfiai feladat, az időfeloldó (time-release) kriptográfiát tárgyalja. Ebben a részben először a kriptográfiai protokollok világába nyújtunk betekintést, majd ismertetjük magát az időfeloldó titkosítás problémakörét, az ahhoz kapcsolódó eddigi eredményeket. Ezt követően vizsgáljuk az M. Kudo és A. Mathuria által 1999-ben közzétett megoldást. [90] Ennek egyik újdonsága a formális eszközök használata, a T. Coffey, P. Saidha (és később T. Newe) nevéhez köthető CSN logikai rendszer alkalmazása. [48][122] Munkánk során átdolgoztuk és pontosítottuk a CSN logikai rendszert és további kiegészítéseket tettünk a Kudo-Mathuria protokollal kapcsolatban. A módosított protokollokat a CSN-logika segítségével vizsgáltuk első megközelítésben. [145] Ezt követően bemutatjuk ugyanennek a problémának az AVISPA fél-automatikus protokoll ellenőrző rendszerrel történő vizsgálatát is. [147] A kétféle megközelítés eredményei összhangban vannak egymással.

Munkánk második nagy fejezete további kiegészítéseket fogalmaz meg a CSN-logikával kapcsolatban. 2005-ben F.-L.Wong és F. Stajano vetette fel azt az igényt, hogy a többcsatornás kriptográfiai protokollok esetén szükséges lenne egy olyan logikai rendszerre, amely alkalmas ezen terület protokolljainak vizsgálatára. [157] Sikerült a CSN-logikát úgy bővítenünk,

hogy az alkalmassá vált a kitűzött feladat megoldására. [146] [148] [150] [149] Alkalmazásképpen bemutatjuk a MANA protokollcsalád első három tagjának vizsgálatát az új rendszer segítségével.

A dolgozat utolsó fejezetében a vizsgálati eljárás (modális logikai eszközök) más területeken történő használhatóságát elemezzük. A protokollszerű megközelítés számos gyakorlati kutatási ágban felbukkan. Itt megemlíthetjük a számítástechnika, a kommunikáció, a kriptográfia, az orvostudomány, a gazdaság egyes részeit. Habár ezek az alkalmazási területek igen távol eshetnek egymástól, az eredmények értékelése és felhasználása nagyon eltérő lehet, elmondható, hogy a különböző tudományterületek a matematika eszközeit használva hasonló eljárásokkal, hasonló keretek között képesek vizsgálni az adott területen kialakított protokollokat. Ennek eredményeképpen várható, hogy az igen eltérő tudományterületek kapcsolódási pontokat találnak, átvehetik egymás tapasztalatait, az eddigiektől eltérő szempontból vizsgálhatják kutatási feladataikat.

2. fejezet

Kriptográfiai alapok

Ebben a fejezetben bemutatjuk azokat a kriptográfiai alapfogalmakat, amelyek szükségesek a továbbiak értelmezéséhez.[1]

2.1. Alapfogalmak

A kriptográfia legalapvetőbb feladata egy üzenet eljuttatása a **küldő** féltől (*sender*) a **fogadó** félig (*receiver*) úgy, hogy az **üzenet** (*message*) tartalmát ne ismerhesse meg más. Az üzenet általában érthető szöveg (*nyílt szöveg, plaintext, cleartext*) - bár mindig figyelembe kell venni a kódolás és az adattömörítés technikai megoldásait is. Az átalakított üzenetet **titkosított üzenet**nek (*chiphertext*) nevezzük, az üzenet átalakítása a **titkosítás** (*encryption*). A fogadó fél a titkosított üzenetet visszaalakítja az eredeti üzenetté. Ez a folyamat a titkosított üzenet **visszafejtés**e (*decryption*).

A titkosítás és visszafejtés, az üzenetek titkos kezelése, továbbítása a **kriptográfia** (*cryptography*) feladata. Azokat a tevékenységeket, amelyek

[1] A fejezet kialakítása során legfőbb forrásul Schneier 1996-ban megjelent könyvét [133] használtuk, de számos más mű ad magyar [45][84] vagy idegen nyelven [38][110] hasonló összefoglalást a témakörben. Ahol szükségesnek láttuk, szerepel az angol és más magyar terminológia is kurzív kiemeléssel. Ez sokszor hasznos a rövidítések, jelölések feloldásában.

a titkosított üzenetek megfejtését, feltörését kísérlik meg, a **kriptoanalízis** (*cryptanalysis*) körébe soroljuk. A két területet összefoglalóan egy tudományágnak tekintjük és **kriptológia** (*cryptology*) néven hivatkozunk rá.

Az adatok **titokban tartás**án (*secrecy*, **bizalmasság** megőrzése - *confidentiality*) kívül a kriptológiának más feladatai is vannak. Például a fogadó fél számára biztosítani kell azt, hogy

- meggyőződhessen az üzenet eredetéről, a küldő fél személyéről - a támadó ne tudja megszemélyesíteni a küldő felet - **hitelesség**, hitelesítés (*authentication*),
- meggyőződhessen az üzenet sértetlenségéről, arról, hogy a feladó által ténylegesen elküldött üzenet jutott el hozzá, az üzenetet nem módosította senki - integritás, **teljesség** (*integrity*),
- a küldő fél ne tudja letagadni az elküldött üzenetet - **letagadhatatlanság** (*nonrepudation*).

A kriptográfia **algoritmus**okat (*cryptographic algorithm, cipher*) és **protokoll**okat (*cryptographic protocol*) használ mindezek megvalósítására. Az algoritmusok olyan függvények, amelyek a titkosítás és a visszafejtés során kerülnek felhasználásra. A protokollok lépések sorozatát foglalják egybe, amelyek segítségével két vagy több partner megvalósítja a kitűzött feladatokat. A két építőelem kölcsönösen kiegészíti egymást. A protokollok magukba építik az algoritmusokat, az algoritmusok pedig támaszkodnak a protokollok által kialakított kapcsolódási pontokra.

2.1.1. Kriptográfiai algoritmusok

Általánosan elfogadott megközelítés az, hogy egy kriptográfiai algoritmust akkor tekintenek igazán megbízhatónak, ha azt nyilvánosságra hozzák, az érdeklődők számára megismerhetővé teszik. Ezek az algoritmusok **kulcs**okat (paramétereket) használnak működésük során, amit viszont titokban tartanak az alkalmazók, ezzel valósítva meg a kívánt biztonsági elemeket.

A titkosítást és visszafejtést a következő módon jelölhetjük:

$$E_{kT}(m) = c \quad D_{kV}(c) = m \quad , vagyis \quad D_{kV}(E_{kT}(m)) = m$$

Itt E a titkosítást végző függvényt, D a visszafejtést végző függvényt, m az üzenetet, c a titkosított üzenetet, kT a titkosító kulcsot, kV a visszafejtő kulcsot jelöli.

Két alapvető, kulcs-alapú algoritmust használhatunk: szimmetrikus kulcsú (*symmetric-key*) és nyilvános kulcsú (*public-key*) algoritmusokat.

Szimmetrikus kulcsú algoritmusok. A szimmetrikus kulcsú algoritmusokat szokták hagyományos (*conventional*), *titkos kulcsú*, vagy *egykulcsos* algoritmusoknak is nevezni. Ezek olyan algoritmusok, ahol a titkosító és a visszafejtő kulcs könnyen kiszámítható egymásból. A legtöbb esetben a kétféle kulcs azonos. Az algoritmus használatakor a küldő és a fogadó félnek meg kell egyeznie a használt kulcsban.[2] A kommunikáció védettsége tehát ebben az esetben a kulcsok minőségén és védelmén múlik. Az előző jelöléseket használva ez a séma a következő módon írható le (k a közös kulcsot jelöli):

$$E_{kT}(m) = c,\ D_{kV}(c) = m,\ kT = kV = k,\ vagyis\ D_k(E_k(m)) = m\,.$$

A szimmetrikus kulcsú titkosító algoritmusokat két nagy csoportra szokás bontani. Az első a **folyam-titkosítók** (*stream ciphers, stream algorithms*) köre, amelyek a nyílt szöveg bit-jein, néha byte-jain hajtanak végre műveleteket. A második a **blokk-titkosítók** (*block ciphers*) köre, amelyek az üzenet meghatározott csoportjain (blokkok) hajtanak végre számításokat. A modern algoritmusok blokkmérete általában 64, 128, 192 vagy 256 bit. Szimmetrikus kulcsú titkosítási rendszerre példa lehet a DES (*Data Encryption Standard*), az IDEA (*International Data Encryption Algorithm*), és az AES (*Advanced Encryption Standard*). További részletek tekintetében a már említett szakirodalmakra hivatkozunk.

Nyilvános kulcsú algoritmusok. A nyilvános kulcsú algoritmusokat szokták *aszimmetrikus* (*asymmetric algorithm*) is nevezni. Ezekben az algoritmusokban a titkosító és a visszafejtő kulcs nem egyezik meg, a két kulcs

[2] A kulcsegyeztetés sok esetben egy védett csatornán történő kommunikációs folyamat végeredményeképpen jön létre. A dolgozat második felében ilyen, több csatornát alkalmazó kriptográfiai protokollokat vizsgálunk.

nem - vagy csak igen nagy számítási időben - származtatható egymásból. A modell működésének lényege, hogy a küldő fél a fogadó fél **nyilvános kulcs**át (*public-key*) használja a titkosítás során. Ez a kulcs szabadon elérhető bárki számára. A fogadó fél a titkosított üzenetet saját **titkos kulcs**ával (*secret-key*) fejti vissza. A nyilvános és titkos kulcs összetartozó párt alkot, a titkos kulcsot a felhasználónak védenie kell.

$$E_{k_{pB}}(m) = c,\ D_{k_{sB}}(c) = m,\ k_{pB} \neq k_{sB},\ vagyis\ D_{k_{sB}}(E_{k_{pB}}(m)) = m\,.$$

Itt E a titkosító-, D a visszafejtő függvényt, m az üzenetet, c a titkosított üzenetet, k_{pB} fogadó B fél nyilvános kulcsa, k_{sB} a fogadó B fél titkos kulcsa. Aszimmetrikus kulcsú titkosítási rendszerre példa lehet az RSA (R. Rivest, A. Shamir, L. Adleman nevéből - 1977), az ElGamal (T. ElGamal nevéből - 1984), az ECC (*Elliptic Curve Cryptography*). További részletek tekintetében szintén a már említett szakirodalmakra hivatkozunk.

Jelölések. A dolgozatban többféle jelölésrendszert alkalmazunk a protokollok leírására. Ennek egyik oka a szakirodalomban kialakult hagyományok követése, a másik az alkalmazott logikai rendszer (CSN-logika) viszonylag összetett leírási módja.

A **protokollok leírása** során hagyománnyá vált a résztvevőket keresztnevekkel jelölni. Az elnevezések kötődnek a szereplők protokollban játszott szerepéhez, a résztvevők protokollbeli tulajdonságaihoz. Az angol nyelvű szakirodalom az úgynevezett **Alice-Bob jelölésrendszert** használja: [84]

Név	Jelölés	Leírás
Alice, Bob	A, B	A kommunikáló felek általános jelölése.
Carol, Dave	C, D	A három, vagy ennél több szereplő esetén a kommunikáló felek jelölése. Két szereplő esetén előfordul az, hogy a lehallgató felet Carol képviseli.
Eve	E	Passzív támadó: lehallgató (*passive attacker*). Nem avatkozik be a protokollba, figyeli az üzenetváltásokat.

Mallory	M	Aktív támadó (*active attacker*). Beavatkozik a protokollba. Az üzeneteket elfoghatja, módosíthatja, a szereplőket megszemélyesíthet, stb.
Trent	T	Abszolút megbízható fél (*absolute trusted third party agent*). Sokszor szervereket jelöl.
Walter	W	Felügyelő (*warden*). Szerepe a résztvevő partnerek segítése a protokoll végrehajtásában.
Peggy	P	Bizonyító (*prover*). Egy vagy több állítás teljesülését bizonyítja, igazolja a protokoll végrehajtása során.
Victor	V	Verifikáló (*verifier*). Ellenőrizni tudja Peggy bizonyító eljárását.

A dolgozatban magyar neveket használunk a következőképp: Alice = Aliz; Bob = Botond; Carol = Cecil; Dave = Dávid; Eve = Evelin; Mallory = Márton; Trent = Tamás; Walter = Valter; Peggy = Petra; Victor = Viktor.

Egy általános üzenetküldés sémája a következő:

$$X \to Y : m$$

Itt X, Y - a résztvevő partnereket jelöli. A partnerek a felsorolt szereplők lehetnek. A nyíl jelzi az üzenetküldés irányát (X küld üzenetet Y-nak). A kettőspont után az elküldött m üzenet szerepel.[3] Egy titkosított üzenetet az $E_k(m)$ forma jelöl, ahol E jelenti magát a titkosító algoritmust, k a titkosító kulcs, m a nyílt szöveg. A $D_k(m)$ a visszafejtés hasonló jelentésű formája. D a visszafejtő algoritmus, k a visszafejtő kulcs, m a visszafejtendő titkosított üzenet. Ritkábban alkalmazott az $S_k(m)$ és $V_k(m)$ jelölés, ami az üzenetek digitális aláírását és annak ellenőrzését jelentik (lásd később). A konkatenációt, összefűzést az egymás után írt, vesszővel elválasztott üzenetdarabok jelölik.

Egy példa mindezekre: $A \to B : \{A, B, E_k(A, m)\}$. Ennek jelentése: Az

[3] m gyakran összetett üzenetet jelöl, amelynek részeit vesszővel elválasztva, kapcsos zárójelek között szoktak felsorolni.

A-val jelölt felhasználó (Aliz) küld üzenetet a B-vel jelölt felhasználónak (Botond). Az üzenet első része Aliz egyedi azonosítója (neve), második része Botond egyedi azonosítója (neve). Az üzent harmadik darabja egy titkosított üzenetdarab, amely a k kulcs felhasználásával titkosítja az A és m üzenetdarabokat, ahol A Aliz azonosítója, m pedig maga az üzenet.

A titkosítás és a visszafejtés más jelölései:

$\{m\}_{k_{AB}}$ $\{m\}_k$	szimmetrikus kulcsú algoritmus, titkosítás és visszafejtés - m az üzenet, k_{AB} (vagy k) az A és B fél közös kulcsa.
$\{m\}_{k_A}$ $\{m\}_{k_{pA}}$	aszimmetrikus kulcsú algoritmus, titkosítás és/vagy digitális aláírás ellenőrzése - m az üzenet, k_A (vagy k_{pA}) az A fél nyilvános kulcsa.
$\{m\}_{k_A^{-1}}$ $\{m\}_{k_{sA}}$	aszimmetrikus kulcsú algoritmus, visszafejtés és/vagy digitális aláírás - m az üzenet, k_A^{-1} (vagy k_{sA}) az A fél titkos kulcsa.

A CSN-logika jelölési rendszere a bemutatott jelölési módok kevert változatának fogható fel, részletesen a 3.2.3. fejezetben kerül ismertetésre.

Kriptoanalízis - Algoritmusok támadása. A kriptoanalízis a következő támadási módszereket alkalmazza:

- Kriptoszöveg alapú támadás (*ciphertext-only attack*). Az analizáló számára néhány olyan üzenet titkosított alakja ismert, amelyeket ugyanazzal a titkosítási algoritmussal titkosítottak. A feladat minél több nyílt szöveg visszaállítása, valamint a kulcs előállítása.
- Ismert nyílt szöveg alapú támadás (*known-plaintext attack*). Az analizáló számára nemcsak titkosított üzenetek ismertek, hanem ezek nyílt szövegű párja is. A feladat a kulcs előállítása, valamint újabb titkosított üzenetek létrehozása a már ismert kulccsal.

- Választott nyílt szöveg alapú támadás (*chosen-plaintext attack*). Az analizáló számára nemcsak nyílt és titkosított üzenet-párok ismertek, hanem a folyamat során el tudja érni, hogy az általa választott nyílt szöveg titkosított megfelelőjéhez is hozzájusson. A feladat szintén a kulcs és újabb titkosított üzenetek előállítása.
- Módosított választott nyílt szöveg alapú támadás (*adaptive-chosen-plaintext attack*). Az előző támadás módosított változata. Az analizáló módosíthatja a nyílt szöveget, amelynek ismét megkapja a titkosított változatát.
- Választott kriptoszöveg alapú támadás (*chosen-ciphertext attack*). Az analizáló az általa választott titkosított szövegek nyílt párját érheti el.
- Választott kulcs alapú támadás (*chosen-key attack*). Az analizáló a különböző kulcsok közötti kapcsolatokat ismeri.
- Kulcs megszerzésén alapuló támadás (*rubber-hose cryptanalysis, purchase-key attack*). Az analizáló valamilyen (nem a fenti felsorolt) módon megszerzi a kulcsot.

A kriptográfiai algoritmusok analizálásának többféle eredménye lehetséges, a feltörésnek különböző típusai vannak:

- Teljes feltörés (*total break*). Az analízis megtalálja a kulcsot; minden üzenet fejthetővé válik.
- Globális megfejtés (*global deduction*). Az analízis olyan algoritmust talál, aminek segítségével a kulcs ismeretének hiányában is tudja fejteni az üzeneteket.
- Lokális megfejtés (*instance (or local) deduction*). Az analízis megtalálja az elfogott titkosított üzenetek megfejtését.
- Információszerzés (*information deduction*). Az analízis a kulcsról, vagy a nyílt szöveggel kapcsolatban szerez információkat. Ez lehet a kulcs néhány bitjének megismerése, a nyílt szöveg szerkezetének feltárása, stb.

Egy algoritmust **feltétlen biztonságos algoritmus**nak (*unconditional secure algorithm, feltétel nélkül biztonságos algoritmus*) tekintünk, ha az

analizáló a rendelkezésére bocsátott tetszőleges mennyiségű kriptoszöveg segítségével sem tudja az algoritmust feltörni. Eddig egyedül a véletlen átkulcsolás algoritmusáról (*Vernam-féle algoritmus, One-Time Pad*) bizonyított a feltétlen biztonságosság. Minden más algoritmus törhető kriptoszöveg alapú támadással. Ez minden lehetséges kulcs kipróbálását jelenti - **teljes kimerítő támadás** (*brute-force attack, nyers erő támadás, primitív próbálgatás módszere*).

Az algoritmusok biztonsága szoros kapcsolatba hozható a bonyolultságelmélettel. Egy **algoritmust kiszámítható biztonságú**nak (*computationally secure, strong secure*) tekintünk, ha a rendelkezésünkre álló eszközök segítségével bizonyos határon (idő vagy adatmennyiség) belül nem tudjuk feltörni.[4] Mindezekről bővebben a 3.1. fejezetben lesz szó.

2.1.2. Kriptográfiai protokollok

Egy protokoll lépésekbe rendezett tevékenységek sorozata két, vagy több partner között, annak céljából, hogy egy adott problémát megoldjanak. A lépések sorozata azt jelenti, hogy egy protokollnak minden esetben van kezdeti és befejező lépése. Egy lépést teljes egészében végre kell hajtani; amíg egy lépés nem fejeződött be, a következő lépés végrehajtása nem kezdődhet el. Egy protokollhoz mindenképpen legalább két szereplő szükséges, egy szereplő által végrehajtott lépéssorozatot nem tekintünk protokollnak. A szereplők lehetnek számítógépek, az azokon működő alkalmazások is.[5]

Egy protokoll a következő kritériumoknak kell megfeleljen:

- A protokoll résztvevői mind ismerik a protokoll lépéseit.

[4] Az adat-komplexitás (*data complexity*) a töréshez szükséges input-adatmennyiséget; a feldolgozási-komplexitás (*processing complexity*) töréshez szükséges számítási kapacitást, időt; a tárolási komplexitás (*storage complexity*) a támadáshoz szükséges számítógép memória mennyiségét méri.

[5] Például SOA ('Service-Oriented Architecture' - Szolgáltatás-orientált architektúra) rendszerekben a szoftver komponensek protokollkapcsolatokat építenek ki.

- A résztvevők elfogadják és teljes mértékben végrehajtják a protokoll-lépéseket. A protokoll-lépések általában vagy számítási lépések egy résztvevő által, vagy üzenetküldés és fogadás a résztvevők között.
- Egy protokoll egyértelmű, minden lépés egyértelműen definiált, nem félreérthető. A protokoll-lépések általában lineárisan követik egymást, az elágazások esetén mindig egyértelmű döntési eljárást kell tartalmaznia a protokollnak a következő lépés kiválasztásához.
- Egy protokoll teljes kell legyen, vagyis minden lehetséges helyzetre legyen megfelelő protokoll-lépés.

A már felsorolt néhány alapfeladaton túl (titkosítás, hitelesítés, üzenetek teljessége és letagadhatatlansága) a kriptográfiai protokollok olyan összetett feladatok megoldását célozzák meg, mint a digitális pénz, az elektronikus banki rendszerek, az elektronikus választási rendszerek, stb.

Egyes protokoll-csoportok kiemelhetők különleges tulajdonságaik alapján:
- **Döntőbírós protokollok** (*arbitrated protocols*). A protokollban szereplő A és B felek egy harmadik, abszolút megbízható T döntőbíró (*arbitrator*) segítségével valósítják meg a protokoll célkitűzéseit.
- **Ítélkező protokollok** (*adjudicated protocols*). A döntőbírós protokollok egy olyan módosított változata, ahol a protokoll két részprotokollra osztható. Az első rész egy nem döntőbírós protokoll, a résztvevők minden alkalommal ezt a protokollrészt igyekeznek használni. A másik részprotokoll egy döntőbírós protokoll, amely akkor kerül alkalmazásra, ha az első részprotokoll végrehajtása során a felek között vita alakul ki. Ilyen esetben a döntőbírót már *ítélkező*nek (*adjudicator*) nevezzük.
- **Önműködő protokollok** (*self-enforcing protocols*). Az ilyen protokollok önmaguk garantálják a helyes működést. Bármely fél csalni próbál, a protokoll lehetővé teszi azt, hogy a másik fél azonnal érzékelje a visszásságot és megszakíthassa a protokoll végrehajtását.

Protokollok ellenei támadások.
A protokollok elleni támadásoknak két nagy csoportja van: az egyik a

passzív támadás (*passive attack*), a másik pedig az **aktív támadás** (*active attack*).

Passzív támadás esetén a támadó fél nem avatkozik be a protokoll működésébe, de igyekszik minden üzenetet lehallgatni, minden információt megszerezni, hogy analizálni tudja azokat. Ez a fajta protokoll-támadás legszorosabban a kriptoszöveg alapú támadáshoz kapcsolható. Nehéz észrevenni.

Aktív támadás esetén a támadó fél aktívan avatkozik be a protokoll lépéseibe. Ebben az esetben lehetősége nyílik a támadónak megszemélyesítenie a protokoll egy szereplőjét, egy üzenetet kicserélni egy másikkal, új üzenetet generálni, törölni egy üzenetet, módosítani egy üzenet tartalmát, egy kommunikációs csatornát felfüggeszteni, hozzájutni tárolt adatokhoz. A támadás módja mindig a protokolltól és a kommunikációs hálózattól függ. A támadás célja ekkor már nem csak információszerzés, hanem lehet például információ korrumpálása, bizalmatlanság keltése a felek között, erőforrások jogtalan felhasználása.

Aktív vagy passzív támadó nem csak külső fél lehet. Sok esetben legális protokoll szereplő (egyéni felhasználó, rendszer-adminisztrátor) dönt úgy, hogy kihasználja az alkalmazott protokoll gyengeségeit. A protokoll legális résztvevőit, ha nem követik a protokoll előírásait, csalóknak (*cheater*) nevezzük. A **passzív csaló** (*passive cheater*) követi a protokoll-lépéseket, de több információt igyekszik megszerezni, mint amit a protokoll biztosítana számára. Az **aktív csaló** (*active cheater*) megbontja a protokoll-folyamatot.

2.1.3. Üzenetküldés titkosítva

Az előző részek ismereteire támaszkodva már többféle módon is meg tudunk valósítani két fél között titkosított kommunikációt. Az alapesetek a következők.

Kommunikáció szimmetrikus titkosítással. A protokoll lépései:

1. Aliz és Botond megállapodik az alkalmazott szimmetrikus kulcsú kriptorendszerben.

2. Aliz generál egy k kulcsot, és eljuttatja azt Botondnak. Az üzenetváltás titkos kell legyen. $A \rightarrow B : k$
3. Aliz titkosítja az m nyílt szöveget a k kulcs felhasználásával. $E_k(m)$
4. Aliz elküldi a titkosított szöveget Botondnak. $A \rightarrow B : E_k(m)$
5. Botond a k kulcs segítségével visszafejti a titkosított szöveget, megkapja Aliz m üzenetét. $D_k(E_k(m)) = m$

Kommunikáció nyilvános kulcsú titkosítással. A protokoll lépései:

1. Aliz és Botond megállapodik az alkalmazott nyilvános kulcsú kriptorendszerben.
2. Aliz hozzájut Botond nyilvános k_{pB} kulcsához. Ez megtörténhet úgy, hogy Aliz eléri a kulcsot egy nyilvános adatbázisból: $T \rightarrow A : k_{pB}$; vagy egyszerűen Botond küldi el neki: $B \rightarrow A : k_{pB}$
3. Aliz k_{pB} felhasználásával titkosítja a nyílt szöveget. $E_{k_{pB}}(m)$
4. Aliz elküldi a titkosított szöveget Botondnak. $A \rightarrow B : E_{k_{pB}}(m)$
5. Botond a saját k_{sB} titkos kulcsával visszafejti a titkosított szöveget, megkapja Aliz üzenetét. $D_{k_{sB}}(E_{k_{pB}}(m)) = m$

Hibrid kriptorendszerek. A protokoll lépései:

1. Aliz és Botond megállapodik az alkalmazott hibrid kriptorendszerben.
2. Aliz hozzájut Botond nyilvános k_{pB} kulcsához. Ez megtörténhet úgy, hogy Aliz eléri a kulcsot egy nyilvános adatbázisból: $T \rightarrow A : k_{pB}$, vagy egyszerűen Botond küldi el neki: $B \rightarrow A : k_{pB}$
3. Aliz generál egy k kapcsolatkulcsot, és titkosítja Botond nyilvános kulcsával. $E_{k_{pB}}(k)$
4. Aliz elküldi Botondnak a titkosított kapcsolatkulcsot. $A \rightarrow B : E_{k_{pB}}(k)$
5. Botond visszafejti a titkosított kapcsolatkulcsot, saját titkos kulcsát használva, megkapja k-t. $D_{k_{sB}}(E_{k_{pB}}(k)) = k$
6. Aliz titkosítja az m üzenetet a k kapcsolatkulccsal. $E_k(m)$
7. Aliz elküldi a titkosított üzenetet Botondnak. $A \rightarrow B : E_k(m)$
8. Botond visszafejti az üzenetet a k kulccsal, megkapja Aliz m üzenetét. $D_k(E_k(m)) = m$

2.1.4. Egyéb protokollépítő elemek

Egyirányú függvények. Az egyirányú függvények fogalma központi jelentőségű a kriptográfiában. Egy f egyirányú függvény (*one-way function*) egy olyan függvénykapcsolat, amelyben az $x \rightarrow f(x)$ hozzárendelés lineáris időben kiszámítható (könnyű kiszámíthatóság), míg az $f(x) \rightarrow x$ meghatározása exponenciális, vagy nagy kitevőjű polinomiális időben lehetséges (nehéz kiszámíthatóság).[6] Az egyirányú függvények felhasználása általában összehasonlítások során, változások vizsgálatakor kerül előtérbe.

Csapóajtó egyirányú függvény (*trapdoor one-way function*) olyan speciális egyirányú függvény, amelynél a $f(x) \rightarrow x$ érték meghatározása egy kiegészítő információ megadásával hatékonnyá tehető.

Speciális egyirányú függvények a **hash függvények** (*one-way hash function, kriptográfiai ellenőrző összeg, üzenet kivonat*). Ezek a függvények a különböző hosszúságú x értékekhez ugyanolyan hosszúságú $f(x)$ értéket rendelnek. Azokban az esetekben, amikor a hash függvény egy kulcsot is használ az $f(x)$ kiszámítása során **kulcsolt hash függvény**ről (keyed hash function), vagy üzenethitelesítési kódról (*MAC - Message Authentication Code, DAC - Data Authentication Code, adathitelesítési kód*) beszélünk. Ennek eredménye az, hogy az egyirányú hash függvényre alapozott összehasonlítások csak a kulcs ismeretében végezhetők el. [45][113] Itt kell megemlítenünk a Debreceni Egyetem Informatika Karán kidolgozott új egyirányú függvény-osztályt, amely mind az elmélet, mind a gyakorlat területén jelentős eredménynek számít. [25][26][27]

Digitális aláírás. A digitális aláírás célja a hagyományos aláírás előnyös tulajdonságait átültetni az elektronikus kommunikációs folyamatokba. Követelményként sorolják fel a hitelességet, a hamisíthatatlanságot, az újrafelhasználhatóság elkerülhetőségét, az aláírt dokumentum integritását, az aláírás letagadhatatlanságát. A hagyományos dokumentumkezelés az aláíráson kívül különböző technikákat (oldalak számozása, pecsét, dokumentumrészek, felsorolások folyamatos számozása, stb.) használ mindezek megvalósítására, de így is lehetséges a követelmények megsértése, kijátszá-

[6] Az itt szereplő bonyolultságelméleti fogalmak pontos meghatározása később, az 3.1.2. fejezetben szerepel.

sa. A számítógépek világában talán még nehezebb egyes követelmények teljesítése az állományok korlátlan másolhatósága és könnyű módosíthatósága miatt. A következő protokollok megoldják a problémakör egyes feladatait.

Dokumentum digitális aláírása szimmetrikus kriptorendszer és döntőbíró segítségvel. Aliz digitálisan aláírt üzenetet kíván küldeni Botondnak. A protokoll megvalósítása érdekében segítségül hívja Tamást, az abszolút megbízható felet. Aliz szimmetrikus kriptográfiai rendszert használ a feladat végrehajtásához. k_{AT} Aliz és Tamás közötti, k_{BT} Botond és Tamás közötti, többször felhasználható titkos kulcs, amelyek még a protokoll megkezdése előtt lettek generálva és szétosztva a megfelelő résztvevők között. Az aláíró protokoll lépései a következők:

1. Aliz titkosítja Botondnak szánt m üzenetét a k_{AT} kulccsal és elküldi azt Tamásnak: $A \rightarrow T : E_{k_{AT}}(m)$
2. Tamás visszafejti az üzenetet a k_{AT} kulccsal: $D_{k_{AT}}(E_{k_{AT}}(m)) = m$. Ezután Tamás kiegészíti az m üzenetet egy C_{TA} igazolással, hogy azt Aliztól kapta, majd titkosítja ezt a Botonddal közös k_{BT} kulccsal: $E_{k_{BT}}(C_{TA}, m)$
3. Tamás elküldi a titkosított üzenetet Botondnak: $T \rightarrow B : E_{k_{BT}}(C_{TA}, m)$
4. Botond visszafejti a kapott üzenetet a k_{BT} kulccsal: $D_{k_{BT}}(E_{k_{BT}}(C_{TA}, m)) = C_{TA}, m$

Belátható, hogy a korábban felsorolt követelményeknek eleget tesz a protokoll és az általa létrehozott digitális aláírás konstrukció. Abban az esetben, ha egy harmadik fél számára kell bemutatni a digitális aláírást, Tamás segítségét kell használni újra a következő lépésekkel:

1. Botond az m üzenetet és a Tamástól kapott C_{TA} igazolást titkosítja saját k_{BT} kulcsával, és elküldi az eredményt Tamásnak: $B \rightarrow T : E_{k_{BT}}(C_{TA}, m)$

2. Tamás visszafejti az üzenetcsomagot a nála lévő k_{BT} kulccsal: $D_{k_{BT}}(E_{k_{BT}}(C_{TA}, m)) = C_{TA}, m$, majd ellenőrzi adatbázisában a C_{TA} igazolást, azt, hogy az eredeti üzenet Aliztól származik.
3. Tamás újra titkosítja az üzenetcsomagot a Cecillel közös titkos k_{CT} kulccsal, majd elküldi az eredményt Cecilnek: $T \rightarrow C : E_{k_{CT}}(C_{TA}, m)$
4. Cecil visszafejti az üzenetcsomagot a nála lévő k_{CT} kulccsal: $D_{k_{CT}}(E_{k_{CT}}(C_{TA}, m)) = C_{TA}, m$, így hozzáfér mind az üzenethez, mind Tamás igazolásához.

Mindkét protokoll működése T munkáján alapul, aki lényegében közjegyzői feladatokat lát el a rendszerben. Az üzenetek és a hozzá tartozó aláírások, igazolások T rendszerében vannak tárolva, ha az sérül, akkor a kialakított digitális aláírási rendszer összeomlik.

Digitális aláírás nyilvános kulcsú kriptográfiával. Vannak olyan nyilvános kulcsú titkosító algoritmusok, amelyek alkalmasak digitális aláírás létrehozására. Ezekben az esetekben a titkos-nyilvános kulcspár egyértelműen azonosítja a tulajdonost. Néhány algoritmus esetén mind a nyilvános, mind a titkos kulcs alkalmas a titkosítás elvégzésére (például az RSA ilyen). Ekkor ha a dokumentumot a titkos kulccsal titkosítjuk, egy digitális aláírást hozunk létre. A protokoll lépései a következők:

1. Aliz titkosítja - és így alá is írja - a dokumentumot a saját k_{sA} titkos kulcsával, majd elküldi az az aláírt dokumentumot Botondnak: $A \rightarrow B : E_{k_{sA}}(m)$
2. Botond visszafejti a titkosítást Aliz k_{pA} nyilvános kulcsával, így ellenőrizve a dokumentum aláírását: $D_{k_{pA}}(E_{k_{sA}}(m)) = m$

A protokoll működése során nem kell igénybe venni egy harmadik fél szolgáltatásait, a létrehozott digitális struktúra mégis kielégíti a felsorolt követelményeket.

Azoknál az algoritmusoknál, amelyek nem rendelkeznek hasonló szimmetrikus tulajdonságokkal, külön algoritmusrészeket kell használnunk a digitális aláírás létrehozására. Számos olyan algoritmus is létezik, amelyek

digitális aláírás létrehozását támogatják, de nem alkalmasak titkosítási feladatok ellátására. Néha alkalmazzák az $S_{k_{sA}}$ (*signature*), illetve a $V_{k_{pA}}$ (*verification*) jelölést is.

Időpecsét és egyirányú hash függvény alkalmazása. A digitális aláírás újbóli felhasználását teszik lehetetlenné a dokumentum mellé csatolt dátum és időadatok, melyeket együtt titkosítanak (együtt írnak alá) a dokumentummal. Ez lehetővé teszi a dokumentumok aláírásának időbeli kötését is. A titkos kulccsal való kódolás miatt a digitális aláírás nem másolható, hiszen a dokumentum megváltoztatása esetén ennek kódolt értéke is változna. Az időpecsét azt biztosítja, hogy a dokumentum az időpecsétben szereplő időpontban már létezett, a dokumentum létrehozásának ideje így behatárolt. Egyes cégek, szervezetek az időpecsét hitelességét növelő egyedi időhöz kötött adatokat, kulcsokat szolgáltatnak az Interneten.

Vannak olyan megoldások is, amelyek egyirányú hash függvényeket alkalmaznak nagyobb méretű dokumentumok egyszerűbb és gyorsabb kezelése érdekében. Ekkor általában a dokumentum aláírása helyett a dokumentum egyirányú hash függvényértékét - amely már rögzített hosszúságú - írják alá. A dokumentumot és a hozzá tartozó aláírt hash értéket együtt kell ebben az esetben kezelni. A megoldás legnagyobb gyakorlati előnye a gyorsasága, így széles körben alkalmazzák a lassabb aszimmetrikus kódolási algoritmusok kiváltására.

További építőelemek. Az előbbi feladatoknak számos más variánsa és megoldási módja lehet. Nem ejtettünk szót a véletlenszámok és az álvéletlenszámok világáról, amelyek gyakorlatban és elméletben is igen szoros kapcsolatba hozhatók a protokollokkal. Szintén lehetne szólni a többszereplős digitális aláírásokról, a digitális aláírás és a titkosítás összekapcsolásáról, stb. Szerencsére a már idézett szakirodalomi források bő tárházai ezeknek az itt nem szereplő részeknek.

2.2. Alap-protokollok

A következő fejezetek egyszerű, alapvető protokollokat mutatnak be. A tárgyalás fő irányvonala a már felsorolt kriptográfiai feladatok (bizalmasság, hitelesség, teljesség, letagadhatatlanság) mentén történik, de nyilvánvaló, hogy nem maradhatunk csak ebben a körben, mivel egy ténylegesen üzemelő rendszerben a számítástechnikai és informatikai feladatok összefonódnak a biztonsági és más feladatokkal. Igyekeztünk a problémák összetettsége szerint rendezni a protokollokat.

2.2.1. Kulcscsere protokollok

A titkosító rendszereket úgy próbálják felépíteni, hogy a kommunikáció folyamán minden összetartozó kommunikációs rész egyedi kulcsot használjon. Egy kulcs érvényessége így időben behatárolt, amivel a támadások hatékonyságát lehet csökkenteni. Ezt a kulcsot általában kapcsolatkulcsnak (*session key, egyedi kulcs*) nevezik. A következő protokollok a kapcsolatkulcs szétosztását teszik lehetővé. Legtöbb esetben a protokoll feltételezi egy Kulcsszétosztó Szerver (*Key Distribution Server, KDC* [7]) működését, amelyet abszolút megbízható harmadik félnek tekintenek a protokoll résztvevői. A kulcsoknak a protokollok elindítása előtt a szerveren kell lenniük. Maguk a protokollok nem foglalkoznak azzal, hogy ezek a kulcsok hogyan kerülnek a szerverre.[8]

Kulcscsere szimmetrikus-kulcsú titkosítással.
A protokoll célja közös kapcsolatkulcs kialakítása a kommunikálni kívánó felek között. A protokoll feltételezi, hogy Aliznak és Botondnak létezik titkos kulcsa (mindegyiküknek külön-külön) a KDC-vel való védett kommunikáció megoldására. A protokoll lépései:

[7] Napjainkban a KDC elnevezést sokszor felváltja a CA - 'Certificate Authority' terminológia, amely az általánosabb, digitális hitelesítési bizonylatok kiadását végző felet jelöli.

[8] A szakirodalom ezt az előkészítő fázist *kriptográfiai inicializáló fázis*nak nevezi.

1. Aliz megkeresi Tamást és egy kapcsolatkulcsot kér tőle a Botonddal való kommunikáció titkosítására.
2. Tamás generál egy véletlen kapcsolatkulcsot. Két példányban titkosítja ezt: az egyik példányt Aliz kulcsával, a másikat Botond kulcsával. A titkosítás után Tamás mindkét példányt elküldi Aliznak.
3. Aliz visszafejti a saját példányát és hozzájut a kapcsolatkulcshoz.
4. Aliz elküldi Botondnak a megmaradt példányt.
5. Botond visszafejti az üzenetet, megkapja a kapcsolatkulcsot.
6. Aliz és Botond a közös kapcsolatkulcsot használva üzenetet váltanak.

Amennyiben a KDC kompromittálódik, úgy az egész védelmi rendszer összeomlik, a támadó minden múltbeli (általa rögzített) és jövőbeli üzenetet megismerhet.

Kulcscsere nyilvános kulcsú titkosítással. A protokoll célja közös kapcsolatkulcs kialakítása a kommunikálni kívánó felek között. A protokoll lényegében megegyezik a hibrid kriptorendszereknél megismert protokollal. A protokoll a nyilvános kulcsú titkosítást alkalmazza a közös kulcs kialakítására. A protokoll lépései:

1. Aliz megkapja Botond nyilvános kulcsát a KDC-től.
2. Aliz generál egy véletlen kapcsolatkulcsot, titkosítja ezt Botond nyilvános kulcsával, és elküldi a titkosított kulcsot Botondnak.
3. Botond, miután visszafejtette Aliz üzenetét saját titkos kulcsával, rendelkezik a kommunikációhoz szükséges kapcsolatkulccsal.
4. A két fél az egymással történő kommunikáció során használhatja a közös kapcsolatkulcsot.

A protokoll támadható az úgynevezett **beékelődő támadás**sal (*Man-in-the-Middle attack, MiM*). Márton, mint aktív támadó, megszemélyesítheti Botondot, amikor Alizzal vált üzenetet, és imitálhatja Alizt, amikor Botonddal cserél üzenetet. A támadás lényegében a nyilvános kulcsok el-

fogására és cseréjére épül.[9] A beékelődő támadás lépései:

1. Aliz átküldi Botondnak saját nyilvános kulcsát. Márton elfogja Aliz üzenetét, és elküldi Botondnak saját nyilvános kulcsát.
2. Botond elküldi Aliznak saját nyilvános kulcsát. Márton elfogja Botond üzenetét, és elküldi Aliznak a saját nyilvános kulcsát.
3. Amikor Aliz üzenetet küld Botondnak, úgy gondolja, hogy azt Botond nyilvános kulcsával teszi. Márton elfogja Aliz üzenetét, és visszafejti azt a saját titkos kulcsával. Ezután Márton titkosítja a visszafejtett üzenetet Botond tényleges nyilvános kulcsával, majd elküldi azt Botondnak.
4. Amikor Botond küld üzenetet, Márton az előző lépéshez hasonlóan képes a megszemélyesítésre: Márton elfogja Botond üzenetét, visszafejti azt. Ezután Márton titkosítja a visszafejtett üzenetet Aliz tényleges nyilvános kulcsával, majd elküldi azt Aliznak.

Az Interlock protokoll. Ezt a protokollt R. Rivest és A. Shamir találta ki 1984-ben, a beékelődő támadás megakadályozására. A protokoll lépései:

1. Aliz elküldi Botondnak saját nyilvános kulcsát.
2. Botond elküldi Aliznak saját nyilvános kulcsát.
3. Aliz titkosítja üzenetét Botond nyilvános kulcsával. Átküldi a titkosított üzenet felét Botondnak.
4. Botond titkosítja üzenetét Aliz nyilvános kulcsával. Elküldi a titkosított üzenet felét Aliznak.
5. Aliz elküldi a megmaradt üzenetrészt Botondnak.
6. Botond összeilleszti az Aliztól kapott két üzenetrészt, és visszafejti a saját titkos kulcsával.
7. Botond elküldi a megmaradt üzenetrészt Aliznak.
8. Aliz összeilleszti a Botondtól kapott két üzenetrészt, és visszafejti saját titkos kulcsával.

[9] Az eset ugyanaz, mint amikor KDC kompromittálódik, Márton kicseréli a szerveren tárolt nyilvános kulcsokat saját nyilvános kulcsára.

A protokoll leglényegesebb pontja az, hogy a félbevágott üzenetrészek Márton által nem használhatók.

Kulcscsere digitális aláírással. A digitális aláírás szintén alkalmas beékelődő támadás megakadályozására. A protokollban Tamás aláírja mind Aliz, mind Botond nyilvános kulcsát. Az aláírt kulcsok tartalmazzák a tulajdonos aláírt tanúsítványát. Amikor Aliz és Botond megkapják a kulcsot, ellenőrzik Tamás aláírását, aminek révén majdnem biztosak lehetnek abban, hogy az adott nyilvános kulcs a megjelölt személyhez tartozik.

A támadó Márton ebben a protokollban nem tudja megszemélyesíteni a résztvevő feleket, ahogy azt korábban láttuk. Viszont Márton egyik lehetséges támadási módja a Kulcsszétosztó Szerver (KDC) támadása, Tamás titkos kulcsának megszerzése. Amennyiben ez sikerül, úgy a támadó képes a rendszert kompromittálni.

Kulcs- és üzenetátvitel. Aliz és Botond számára nem szükséges teljesen lezárni a kulcscsere protokollt az üzenetváltást megelőzően. A következő protokollban Aliz úgy küld üzenetet Botondnak, hogy előtte nem történt kulcscsere. A protokoll a hibrid kriptorendszerek tulajdonságait használja ki. A protokoll lépései:

1. Aliz generál egy véletlen k kapcsolatkulcsot, és titkosítja az m üzenetet ezzel a kulccsal: $E_k(m)$.
2. Aliz megszerzi Botond k_{pB} nyilvános kulcsát.
3. Aliz titkosítja a k kulcsot Botond nyilvános kulcsával: $E_{k_{pB}}(k)$.
4. Aliz elküldi a titkosított üzenetet és a titkosított kapcsolatkulcsot Botondnak. A beékelődő támadás még hatékonyabb elkerülése céljából Aliz alá is írhatja k_{sA} titkos kulcsával ezt az üzenetet: $E_{k_{sA}}(E_{k_{pB}}(k), E_k(m))$.
5. Botond ellenőrzi Aliz aláírását Aliz nyilvános k_{pA} kulcsával: $D_{k_{pA}}(E_{k_{pB}}(k), E_k(m))$. Valamint visszafejti a k kapcsolatkulcsot a saját titkos k_{sB} kulcsával: $D_{k_{sB}}(E_{k_{pB}}(k)) = k$.

6. Botond visszafejti a k kulcs alkalmazásával Alice üzenetét: $D_k(m) = m$.

Itt is meg kell jegyeznünk, hogy a digitális aláírás alkalmazásakor (4. lépés) a gyakorlatban nem az egész üzenetet, hanem annak kivonatát írják alá. Az ilyen esetekben viszont a dokumentumot és a kivonatra vonatkozó aláírást együtt kell kezelni.

Kulcs- és üzenetszórás. A kulcs- és üzenetszórás (*key- and message broadcast*) jelentése lényegében az, hogy a kulcsot és/vagy az üzenetet nem egy partnerhez, hanem többhöz továbbítjuk egyazon protokoll keretében, ugyanabban az időben. A következő protokollban Aliz titkosított üzenetet küld ezen a módon Botond, Cecil és Dávid számára. A protokoll az előbbi protokoll bővített változata. A protokoll lépései:

1. Aliz létrehoz egy véletlen k kapcsolatkulcsot, és titkosítja az m üzenetet a kulcs felhasználásával: $E_k(m)$.
2. Aliz megszerzi Botond, Cecil és Dávid nyilvános kulcsát. Ezek sorban: k_{pB}, k_{pC}, k_{pD}
3. Aliz titkosítja a k kulcsot sorban a nyilvános kulcsokkal: $E_{k_{pB}}(k)$, $E_{k_{pC}}(k)$, $E_{k_{pD}}(k)$
4. Aliz elküldi a titkosított üzenetet és a titkosított kulcsokat a többiek felé. Ezeket az üzeneteket bárki foghatja a hálózaton.
5. A megfelelő titkos kulcsokkal (k_{sB}, k_{sC}, k_{sD}) csak Botond, Cecil és Dávid képes visszafejteni a k kulcsot: $D_{k_{sB}}(k) = D_{k_{sC}}(k) = D_{k_{sD}}(k) = k$.
6. A k kulccsal Botond, Cecil, Dávid vissza tudja fejteni a titkosított üzenetet: $D_k(m) = m$.

2.2.2. Hitelesítés, partnerazonosítás

A hitelesítés, felhasználó azonosítás (*authentication*) számos rendszer használatakor felmerül. Az egyik leggyakrabban használt módszer ennek

a feladatnak a megoldására a jelszó használata. A felhasználó a jelszó megadásával igazolja, hogy ismeri azt a titokdarabot, amelyet az ellenőrző fél is ismer - közvetlen, vagy közvetett módon. A következő protokollok ennek a módszernek a különböző variánsait mutatják be.[10]

Hitelesítés egyirányú függvénnyel. R. Needham és M. Guy ismerte fel azt, hogy egyirányú függvények alkalmazása esetén magát a jelszót nem kell az ellenőrző félnek tárolnia. [133] A tárolt információ az egyirányú függvény függvényértéke lehet, ami számos előnyt rejt magában. A protokoll lépései:

1. Aliz elküldi jelszavát az ellenőrző félnek.
2. Az ellenőrző fél kiszámítja az alkalmazott egyirányú függvény függvényértékét a megadott jelszóra.
3. Az ellenőrző fél összehasonlítja a kapott függvényértéket az általa tárolt értékkel.

A jelszavakhoz tartozó tárolt függvényértékek eltulajdonítása az egyirányú függvények kedvező tulajdonságai ellenére sem minden esetben feleslegesek a támadó számára. Amennyiben megfelelő számítási kapacitás áll a támadó fél rendelkezésére, úgy képes lehet támadást intézni a jelszavas védelmet alkalmazó rendszerek ellen is. Számos közlemény elemez olyan eseteket, amikor különböző támadási stratégiákkal sikerült eredményesen jelszavakat visszaállítani. Az egyik eshetőség a **szótáralapú támadás** (*dictionary attack*), amelynek a lényege az, hogy egy megfelelően kialakított, értelmes szavakat, neveket tartalmazó listát próbálnak ki az egyirányú függvény feltörésére.

A támadások megnehezítésére sok módszert lehet alkalmazni. Lehetőség van véletlen bitsorozatokkal kiegészíteni a jelszót, növelve az ellenőrizendő jelszó-sztring hosszát (*salt*), használhatók jelmondatok, tokenek, stb. [161]

[10] A jelszavas védelmen kívül más lehetőségek is vannak a felhasználók azonosítására. Részletesen foglalkozik ezzel a témával például egészségügyi körben Ködmön József és Bodnár Károly. [85] Napjainkban a biometrikus azonosítás is egyre nagyobb szerepet játszik ezen a terülten. [144][6]

Hitelesítés nyilvános kulcsú kriptográfiával. Számítógépes rendszereknél felvetődik a jelszavak ellopásának sokféle módozata. Az egyik cél lehet a jelszavak eltulajdonítása még az egyirányú függvény alkalmazása előtt. A támadónak lehetősége adódhat olyan program telepítésére, amely figyeli és tárolja a rendszerbe belépő felhasználó billentyűzetleütéseit[11], szervereken nem lehetetlen a memóriatartalom és a kommunikációs portok figyelése, stb. Egyféle megoldás lehet a nyilvános kulcsú titkosítás technikájának alkalmazása. [12] A protokoll feltételezi, hogy az ellenőrző szerepét betöltő szerver számítógép tárolja a felhasználók nyilvános kulcsait. A protokoll lépései:

1. A szerver egy véletlen m sztringet küld Aliznak.
2. Aliz titkosítja (aláírja) az m sztringet a titkos k_{sA} kulcsával: $E_{k_{sA}}(m)$, majd visszaküldi azt a szervernek, megadva a nevét is.
3. A szerver megkeresi Aliz nyilvános k_{pA} kulcsát a tárolt listában, és visszafejti a titkosított üzenetet (ellenőrzi az aláírást): $D_{k_{pA}}(E_{k_{sA}}(m)) = m$.
4. Amennyiben a visszafejtett sztring megegyezik a szerver által Aliznak elküldött sztringgel, a szerver engedélyezi Aliz számára a rendszerbe történő belépést.

Kölcsönös hitelesítés az Interlock protokollal. Tételezzük fel, hogy Aliz és Botond kölcsönösen hitelesíteni akarják egymást. Mindkettőjük rendelkezik egy olyan jelszóval, amelyet a másik fél is ismer. P_A Aliz jelszava, P_B Botond jelszava. k_{pA}, k_{pB}, k_{sA}, k_{sB} a megfelelő nyilvános és titkos kulcsokat jelölik. A protokoll lépései:

1. Aliz és Botond eléri egymás nyilvános kulcsát: k_{pA}, k_{pB}

[11] Ez akár távolabbról, közvetlen telepített program nélkül is kivitelezhető. Az ilyen támadások a rendszerek által kibocsátott elektromágneses sugárzást figyelik. [91][153]

[12] Egy új megoldás A. Pethő, A. Huszti és J. Folláth által kidolgozott aszimmetrikus rendszer, amely USB meghajtót alkalmaz hardver-kulcs tárolására. [61]

2. Aliz titkosítja a P_A jelszót Botond nyilvános kulcsával $E_{k_{pB}}(P_A)$, és elküldi azt Botondnak.
3. Botond titkosítja a P_B jelszót Aliz nyilvános kulcsával $E_{k_{pA}}(P_B)$, és elküldi azt Aliznak.
4. Aliz visszafejti a kapott üzenetet: $D_{k_{sA}}(E_{k_{pA}}(P_B)) = P_B$, és ellenőrzi annak helyességét.
5. Botond visszafejti a kapott üzenetet: $D_{k_{sB}}(E_{k_{pB}}(P_A)) = P_A$, és ellenőrzi annak helyességét.

A protokoll ellen a következő beékelődéses támadás építhető fel:

1. Aliz és Botond nyilvánosságra hozzák nyilvános kulcsaikat, és kezdeményezik a partner nyilvános kulcsának elérését. Márton elfogja ezeket az üzeneteket, kicseréli Botond nyilvános kulcsát a sajátjára, és elküldi azt Aliznak. Hasonló módon kicseréli Aliz nyilvános kulcsát a sajátjára, és elküldi azt Botondnak.
2. Aliz titkosítja a P_A jelszót a nála lévő, általa Botond nyilvános kulcsának tekintett kulccsal, és ezt elküldi Botondnak. Márton elfogja ezt az üzenetet, visszafejti P_A-t privát kulcsával, újra titkosítja Botond nyilvános kulcsával, és elküldi a titkosított kulcsot neki.
3. Hasonlóan, Botond titkosítja a P_B jelszót a nála lévő, általa Aliz nyilvános kulcsának tekintett kulccsal, és ezt elküldi Aliznak. Márton elfogja ezt az üzenetet is, visszafejti P_B-t privát kulcsával, újra titkosítja Aliz nyilvános kulcsával, és elküldi a titkosított kulcsot neki.
4. Aliz visszafejti a P_B jelszót, és ellenőrzi, hogy az korrekt-e.
5. Botond visszafejti a P_A jelszót, és ellenőrzi, hogy az korrekt-e.

A protokoll lefolyása során Aliz és Botond semmi különbséget nem tapasztal a támadás során. Márton viszont ezzel a beékelő támadással megtudja mind a P_A, mind a P_B jelszavakat.

1989-ben D. Davies és W. Price javasolta az Interlock protokoll felhasználását hitelesítésre, de 1994-ben S. Bellovin és M. Merritt sikeres támadást írt le ez ellen. [24] A protokollt 1996-ban C. Ellison javította. [58]

2.2.3. Hitelesítés és kulcscsere

Ebben a fejezetben néhány olyan protokollt mutatunk be, amelyek kombinálják a hitelesítést és a kulcsok cseréjét. A kiindulási feltételek a következők: Aliz és Botond egy számítógépes hálózat két távoli felhasználója, akik titkosított kommunikációt akarnak egymással lebonyolítani. A feladat annak biztosítása, hogy közös titkos kulcsot tudjanak használni, és kölcsönösen hitelesítsék is egymást. A protokollok többsége feltételezi, hogy Tamás, a megbízható harmadik fél különböző titkos kulcsokat osztott ki a protokoll megkezdése előtt minden résztvevőnek (k_{AT}, k_{BT}). A leírás során i egy index-szám, k véletlen kapcsolatkulcs, l életidő, t_A, t_B időpecsétek, és r_A, r_B véletlenszámok (*nonce*).

A Wide-Mouth Frog protokoll. Egyszerű szimmetrikus kapcsolatkulcsok kezelésére szolgáló protokoll, amely megbízható harmadik felet (Tamás) alkalmaz. Aliz és Botond rendelkezik a megfelelő titkos kulcsokkal Tamás felé. A protokollban ezek a titkos kulcsok a kapcsolatkulcsok szétosztásában játszanak szerepet. A protokoll lépései:

1. Aliz összefűz egy időpecsétet, Botond azonosítóját (nevét) és egy véletlen kapcsolatkulcsot, majd mindezt titkosítja a Tamással közös kulccsal. Ezután azonosítóját (nevét) és a titkosított üzenetdarabot elküldi Tamásnak: $A \rightarrow T : A, E_{k_{AT}}(t_A, B, k)$
2. Tamás visszafejti az Aliztól kapott üzenetet. Ezután összefűz egy új időpecsétet, Aliz azonosítóját (nevét) és a véletlen kapcsolatkulcsot. Titkosítja mindezt a Botonddal közös kulccsal. Tamás a következő üzenetet küldi ezután Botondnak: $T \rightarrow B : E_{k_{BT}}(t_B, A, k)$.

A protokoll egyik gyengesége az, hogy Alizra bízza a kapcsolatkulcs generálását. Egy véletlen kapcsolatkulcs létrehozása során számos tényezőt figyelembe kell venni. Egy gyengén megkonstruált kulcs nagyobb valószínűséggel támadható, így érdemes támogatást nyújtani a kulcsgeneráláshoz.

A Yahalom protokoll. A protokoll feltételezi, hogy Aliz és Botond (külön-külön) titkos kulccsal rendelkezik Tamás felé. A protokoll végre-

hajtása után Aliz és Botond is megbízhat abban, hogy a kijelölt féllel kommunikál, nem egy harmadik féllel. A protokoll újdonsága az, hogy először Botond lép kapcsolatba Tamással, aki csak Aliz felé küld üzenetet. A protokoll lépései:

1. Aliz összefűzi azonosítóját (nevét) és egy véletlen r_A számot, és ezt elküldi Botondnak: $A \to B : A, r_A$
2. Botond összefűzi Aliz azonosítóját, az Aliz által küldött véletlen számot, a saját maga által generált véletlen számot. Mindezt titkosítja a Tamással közös titkos kulccsal. Tamás felé a következő üzenetet küldi: $B \to T : B, E_{k_{BT}}(A, r_A, r_B)$.
3. Tamás két üzenetet generál. Az első tartalmazza Botond azonosítóját (nevét), egy véletlen kapcsolatkulcsot, az Aliz által generált véletlen számot, és a Botond által generált véletlen számot. Mindezt titkosítja az Alizzal közös kulccsal. A második üzenet tartalmazza Aliz azonosítóját (nevét), a véletlen kapcsolatkulcsot. Ezt az üzenetet a Botonddal közös titkos kulccsal titkosítja. Tamás mindkét üzenetet Aliz felé továbbítja: $T \to A : E_{k_{AT}}(B, k, r_A, r_B), E_{k_{BT}}(A, k)$.
4. Aliz visszafejti az első üzenetet, kibontja k-t, és ellenőrzi, hogy az üzenetből kibontott r_A egyezik-e az általa az első lépésben generált véletlen számmal. Aliz ezután két üzenetet küld Botondnak. Az első a Tamástól kapott üzenet Botond kulcsával titkosított része, a második pedig az r_B véletlen szám titkosítása a k kapcsolatkulccsal: $A \to B : E_{k_{BT}}(A, k), E_k(r_B)$.
5. Botond visszafejti a kapott üzenetrészt az általa ismert titkos kulccsal, kibontja a k kapcsolatkulcsot. Ezután ellenőrzi, hogy a második üzenetrész visszafejtésével kapott r_B véletlen szám egyezik-e azzal a számmal, amit a második lépésben generált.

A Needham-Schroeder protokoll. A protokollt R. Needham és M. Scroeder közölte 1978-ban. [117] A protokoll lépései:

1. Aliz üzenetet küld Tamásnak, közölve azonosítóját (nevét), Botond azonosítóját (nevét), és egy véletlen számot: $A \to T : A, B, r_A$

2. Tamás generál egy véletlen k kapcsolatkulcsot, majd titkosít a Botonddal közös titkos kulccsal egy olyan üzenetet, amely tartalmazza a véletlen k kapcsolatkulcsot, Aliz azonosítóját (nevét). Ezután titkosítja az Alizzal közös titkos kulccsal az Aliztól kapott r_A véletlen számot, Botond azonosítóját (nevét), a k kulcsot és az előbbiekben titkosított üzenetet. Végül üzenetet küld Aliznak a következőképpen: $T \rightarrow A : E_{k_{AT}}(r_A, B, k, E_{k_{BT}}(k, A))$
3. Aliz visszafejti az üzenetet, és kinyeri a k kapcsolatkulcsot. Ellenőrzi az üzenetben lévő r_A és az általa generált, az első lépésben elküldött véletlen érték egyezését. Ezután elküldi Botondnak a megkapott titkosított üzenetrészt: $A \rightarrow B : E_{k_{BT}}(k, A)$
4. Botond visszafejti az üzenetet, és kinyeri a k kapcsolatkulcsot. Ezután generál egy véletlen r_B értéket, majd ezt a k kulccsal titkosítva elküldi Aliznak: $B \rightarrow A : E_k(r_B)$
5. Aliz visszafejti a k kulcs segítségével az üzenet tartalmát, kiszámítja $r_B - 1$ értékét, és titkosítja ezt a k kulccsal. Botondnak a következő módon válaszol: $A \rightarrow B : E_k(r_B - 1)$
6. Botond visszafejti az üzenetet a k kulcs felhasználásával, és ellenőrzi az elküldött $r_B - 1$ érték egyezését.

Az r_A, r_B és $r_B - 1$ számok használata az **ismétléses támadás** (*replay attack, visszajátszásos támadás*) megelőzését szolgálja. Egy ilyen támadás során Márton rögzíti a kommunikáció üzeneteit és később a protokoll támadására használja ezeket. A protokoll erősebb verziója időpecsétet használ ennek a támadásnak az elkerülésére. [65][53] A Tamás és Aliz közötti kulcs sérülése még veszélyesebb, Márton még akkor is képes a protokoll támadására, ha Aliz megváltoztatja a kulcsát. 1987-ben R. Needham és M. Schroeder módosította az eredeti protokollt mindezen hibák javítására. Az új protokoll az ugyanabban a folyóiratban, ugyanabban a számban megjelenő Otway-Rees protokollal egyezik meg.

Otway-Rees protokoll. A protokoll szimmetrikus kulcsú titkosítást használ a hitelesítés és a kulcscsere megoldására. A protokoll lépései:

1. Aliz generál egy üzenetet a Tamással közös titkos kulccsal titkosítva, amely tartalmaz egy i index számot, Aliz azonosítóját (nevét), Botond azonosítóját (nevét), és egy véletlen r_A számot. Botondnak ezt a titkosított üzenetet és a hozzá fűzött index számot, a neveket küldi el: $A \rightarrow B : i, A, B, E_{k_{AT}}(r_A, i, A, B)$
2. Botond egy új r_B véletlen számot generál, majd létrehoz egy üzenetet Tamással közös titkos kulccsal titkosítva, amely tartalmazza az r_B véletlen számot, az i index számot, Aliz és Botond nevét. Botond ezután a következő formájú üzenetet küldi Tamásnak:
$B \rightarrow T : i, A, B, E_{k_{AT}}(r_A, i, A, B), E_{k_{BT}}(r_B, i, A, B)$
3. Tamás generál egy k kapcsolatkulcsot, és a következő üzenetet küldi el Botondnak: $T \rightarrow B : i, E_{k_{AT}}(r_A, k), E_{k_{BT}}(r_B, k)$. Itt a titkosító kulcsok Tamás és Aliz, valamint Tamás és Botond közös kulcsai.
4. Botond elküldi Aliznak a neki szánt részt az index számmal:
$B \rightarrow A : i, E_{k_{AT}}(r_A, k)$
5. Aliz visszafejti az üzenetet, kinyeri a k kulcsot és az r_A véletlen számot, amit összehasonlít az általa generált és tárolt számmal.

A Kerberos protokoll. A Kerberos protokoll a Needham-Schroeder protokoll egy változata. Az 5-ös verzióban Aliz és Botond rendelkezik titkos kulccsal Tamás felé, a protokoll során Aliz egy kapcsolatkulcsot akar létrehozni a Botonddal való kommunikációhoz. A protokoll lépései:

1. Aliz üzenetet küld Tamásnak az azonosítók (nevek) megadásával:
$A \rightarrow B : A, B$
2. Tamás két titkosított üzenetdarabot hoz létre. Az üzenetdarabok a megfelelő szereplővel közös titkos kulccsal vannak titkosítva és tartalmazzák a t időpecsétet, egy l élettartamot jelölő számot, egy k kapcsolatkulcsot és a megfelelő személyi azonosítókat (neveket). Mindkét üzenetdarabot Aliznak küldi el Tamás:
$T \rightarrow A : E_{k_{AT}}(t, l, k, B), E_{k_{BT}}(t, l, k, A)$

3. Aliz létrehozza a k kapcsolatkulccsal titkosított üzenetdarabot, amely tartalmazza az azonosítóját (nevét) és a t időpecsétet. Ezután Aliz a következő üzenetet küldi Botondnak:
   $A \rightarrow B : E_k(A, t), E_{k_{BT}}(t, l, k, A)$
4. Botond válaszul a $t + 1$ számot küldi vissza Aliznak a k kulccsal titkosítva: $B \rightarrow A : E_k(t + 1)$

A protokoll érzékeny az időre, a résztvevők óráját a támadások elkerülésére szinkronizálni kell.

A Neumann-Stubblebine protokoll. A Kerberos protokoll javítására alakította ki 1992-ben A. Kehne, J. Shönwälder és H. Langendörfer a következő protokollt, ami szintén érzékeny a szinkronizálásra. [82] Egy protokoll időérzékenységét **SRA-támadás** (*suppress-replay attack*) végrehajtására lehet használni. Ez a támadás a szinkronizálatlan üzenetküldést, a késleltetés és a visszajátszás lehetőségeit aknázza ki. A protokoll javított verziója, amely megkísérli az előző támadás elleni védekezést, 1993-ban született meg. [119] A protokoll a Yahalom protokoll erősített változata. A protokoll lépései:

1. Aliz a saját azonosítóját (nevét) és egy véletlen r_A számot küld el Botondnak: $A \rightarrow B : A, r_A$
2. Botond Aliz azonosítóját (nevét), a kapott r_A véletlen számot és egy t_B időpecsétet összefűz és titkosít a Tamással közös titkos kulccsal. Tamásnak a következő üzenetet küldi (B Botond azonosítója (neve), r_B egy Botond által generált véletlen szám):
   $B \rightarrow T : B, r_B, E_{k_{BT}}(A, r_A, t_B)$
3. Tamás generál egy véletlen k kapcsolatkulcsot. Ezután két üzenetet hoz létre titkosítva Alizzal és Botonddal közös titkos kulccsal. Aliznek a következő üzenetet küldi:
   $T \rightarrow A : E_{k_{AT}}(B, r_A, k, t_B), E_{k_{BT}}(B, k, t_B), r_B$

4. Aliz visszafejti az üzenetet, ami az ő kulcsával van titkosítva, kinyeri a k kapcsolatkulcsot. Ellenőrzi, hogy az üzenetben lévő r_A érték megegyezik-e az általa generált véletlen számmal, amit az első lépésben küldött el Botondnak. Ezután Aliz két üzenetrészt küld Botondnak. Az első a Tamástól kapott üzenetrész, a második pedig az r_B véletlen szám titkosítása a k kapcsolatkulccsal:
   $A \rightarrow B : E_{k_{BT}}(A, k, t_B), E_k(r_B)$
5. Botond visszafejti az üzenetdarabot a saját titkos kulcsával, kinyeri a k kapcsolatkulcsot. Ellenőrzi, hogy a t_B és r_B értékek azonosak-e az általa a második lépésben generált és nála tárolt megfelelő értékekkel.

A megfelelő véletlen számok és az időpecsét egyezése esetén Aliz és Botond biztosak lehetnek egymás azonosításában és a közös kapcsolatkulcsban. A protokoll működése során nem szükséges a partnerek idő-szinkronizációja, mivel az időpecsét csak Botond órájához kötődik.

A protokoll egy további érdekessége, hogy Aliz a Tamástól kapott üzenetet Botond azonosítására időkorlát nélkül használhatja fel. A protokoll befejezése és a kapcsolat lezárása után Aliz és Botond újra azonosíthatja a másik felet egy háromlépéses kiegészítő protokoll segítségével, ahol nem kell Tamást segítségül hívniuk. Az új véletlen számok az ismétléses támadás kivédését szolgálják. A kiegészítő protokoll lépései:

1. Aliz elküldi Botondnak az előző protokoll harmadik lépésében Tamástól kapott üzenetet egy új r'_A véletlen számmal összefűzve:
   $A \rightarrow B : E_{k_{BT}}(A, k, t_B), r'_A$
2. Botond Aliznek egy új r'_B véletlen számot és a k kapcsolatkulccsal titkosítva Aliz új véletlen számát küldi vissza: $B \rightarrow A : r'_B, E_k(r'_A)$
3. Aliz Botondnak visszaküldi az r'_B új véletlenszámot a k kulccsal titkosítva: $A \rightarrow B : E_k(r'_B)$

A Denning-Sacco protokoll. A protokollt D. E. Denning és G. M. Sacco dolgozta ki 1981-ben. A protokoll nyilvános kulcsú kriptográfiai eszközöket használ, Tamás tárolja a résztvevők nyilvános kulcsait. A protokoll lépései:

1. Aliz üzenetet küld Tamásnak a saját és Botond azonosítójával (nevek): $A \rightarrow T : A, B$
2. Tamás titkos k_{sT} kulcsával aláírva elküldi Aliznak Botond nyilvános k_{pB} kulcsát, valamint elküldi szintén k_{sT} titkos kulcsával aláírva Aliz saját nyilvános k_{pA} kulcsát: $T \rightarrow A : E_{k_{sT}}(B, k_{pB}), E_{k_{sT}}(A, k_{pA})$
3. Aliz elküld Botondnak egy véletlen k kapcsolatkulcsot és egy t_A időpecsétet, aláírva saját titkos kulcsával és titkosítva Botond nyilvános kulcsával, csatolja a nyilvános kulcsok Tamás által aláírt darabjait is: $A \rightarrow B : E_{k_{pB}}(E_{k_{sA}}(k, t_A)), E_{k_{sT}}(B, k_{pB}), E_{k_{sT}}(A, k_{pA})$
4. Botond visszafejti Aliz üzenetét saját titkos kulcsával és ellenőrzi Aliz aláírását Aliz nyilvános kulcsával, valamint megvizsgálja az időpecsét érvényességét is.

A protokoll befejezésekor Aliz és Botond kezében van a további védett kommunikációt lehetővé tevő k kapcsolatkulcs.

Érdekes és elgondolkodtató tulajdonsága a protokollnak az, hogy a protokoll lezárása után Botond kiadhatja magát Aliznak. [2] A protokollt kihasználó lépések a következők:

1. Botond elküldi saját és Cecil nevét Tamásnak: $B \rightarrow T : B, C$
1. Tamás aláírva elküldi mind Botond, mind Cecil nyilvános kulcsát Botondnak: $T \rightarrow B : E_{k_{sT}}(B, k_{pB}), E_{k_{sT}}(C, k_{pC})$
1. Botond elküldi Cecilnek az aláírt kapcsolatkulcsot és időpecsétet - amit előzetesen kapott Aliztől -, titkosítva Cecil nyilvános kulcsával; hozzáfűzve a két partner aláírt kulcsait: $B \rightarrow C : E_{k_{pC}}(E_{k_{sA}}(k, t_A)), E_{k_{sT}}(A, k_{pA}), E_{k_{sT}}(C, k_{pC})$
1. Cecil visszafejti Aliz üzenetét, ellenőrzi Aliz aláírását, ellenőrzi, hogy az időpecsét valid.

A protokoll lezárásakor Cecil úgy hiszi, hogy Alizzel kommunikál. Botond képes megtéveszteni Cecilt, vagy bárki mást az időpecsét lejártáig.
A protokollhibát a következőképp lehet megakadályozni: cseréljük ki az eredeti protokoll harmadik lépését a következőre:
$A \rightarrow B : E_{k_{pB}}(E_{k_{sA}}(A, B, k, t_A)), E_{k_{sT}}(A, k_{pA}), E_{k_{sT}}(B, k_{pB})$.

Látható, hogy a lépés a szereplők nevét csatolja az Aliz által aláírt üzenetdarabhoz. Botond ekkor már nem tudja felhasználni Aliz régebbi üzeneteit. A javításokat 1997-ben tette közzé G. Lowe. [97]

2.3. További protokollok, protokollfeladatok

Ebben a fejezetben tovább folytatjuk a protokollokkal kapcsolatos alapok ismertetését. Az előzőektől eltérően - ahol részletesen bemutatásra kerültek egyes protokollok -, a következőkben a fő hangsúly a protokollok által megoldandó feladatok pontos megfogalmazásán lesz. Az itt szereplő problémakörök a gyakorlati alkalmazások kialakításakor már összetett funkciók megvalósítását jelentik, amelyekben sok esetben nem is egy, hanem protokollok egy halmaza képes lefedni az igényeket.

Titokvágás - *Secret splitting.* A feladat egy titokban tartandó üzenet n részre vágása és szétosztása a partnerek között úgy, hogy a különálló darabokból ne lehessen következtetni az üzenet egészére, n-nél kevesebb rész ne tegye lehetővé a titok felfedését. Ez azzal a lehetőséggel is jár, hogy egyetlen darab elvesztése magával vonja a titok végleges elvesztését.

A probléma megoldása két és több szereplő esetén is lehetséges, egyik mód az XOR (kizáró vagy) logikai művelet alkalmazása.

Titokmegosztás - *Secret sharing.* A kiinduló probléma a következő: a titokban tartandó üzenetet darabokra kell vágnunk úgy, hogy a darabokat kapó n partner közül legalább k ($k \leq n$) félnek együtt kelljen működnie a titok visszanyerésére. Feltétel az, hogy $k-1$ közreműködő fél ne tudja megszerezni a titkot, de bármely k fél együttese lehetővé tegye a titok felfedését. A problémakör másik neve küszöbséma (*threshold scheme*), a titokrészeket árnyéknak (*shadow*) nevezik. Számos más verziója lehetséges a problémának, például a résztvevő feleket különböző szintű csoportokba lehet osztani, az egyes csoportokból más és más számú résztvevőnek kell közreműködnie a teljes titok visszaállításához.

Időpecsét szolgáltatások - *Timestamping services.* Sok esetben szükség van arra, hogy egy dokumentumról bizonyítani tudjuk, hogy egy meghatározott időpontban már létezett. Ilyen eset lehet egy szerzői joggal, egy szabadalommal kapcsolatos probléma. Digitális dokumentumok esetén a már említett módokon (2.1.4. fejezet) lehet megoldani a kérdést.

Kulcs letétbe helyezése - *Key escrow.* A feladat olyan kriptorendszer magalkotása, amely megtartja az egyén titoktartási jogát, ugyanakkor lehetővé teszi azt, hogy bírósági határozat esetén a hatóságok betekinthessenek ezekbe az egyéni titkokba. A probléma felmerül vállalati környezetben is, amikor már célszerű különválasztani a titkosításra és a digitális aláírásra szánt felhasználást, külön kulcsokkal kezelni ezeket.

Vak aláírás - *Blind signature.* A digitális aláírás létrehozásakor az aláíró ismeri az aláírt dokumentumot, annak tartalmát. Egyes esetekben szükség lehet olyan protokollra, amely biztosítja a dokumentum védelmét, lehetővé teszi azt, hogy az aláíró ne ismerhesse az általa aláírt dokumentum tartalmát - ez a vak aláírás.

Azonosítón alapuló titkosítás - *Identity-based cryptography.* Egy érdekes, és a gyakorlatban egyre nagyobb szerepet játszó lehetőség az azonosítón alapuló titkosítás (*Identity Based Encryption, IBE*). Ez a nyilvános kulcsú titkosításra épülő forma egy olyan megoldás, amikor a küldő félnek nem szükséges a partner nyilvános kulcsának ismerete, elegendő a fogadó fél egy megfelelő adatának (például e-mail cím) tudása. A megoldás egyik jellemzője, hogy erősen épít egy megbízható szerver szolgáltatásaira, és az, hogy a kulcsok képzése és az üzenetküldés sorrendje megváltozik.[13]

Elektronikus választási rendszer - *Secure elections.* A választásokkor többféle biztonsági követelménynek kell eleget tenniük a rendszereknek.

[13]Az azonosítón alapuló titkosításnak számos alkalmazási lehetősége van. A DE-EK Egészségügyi Informatika Tanszék munkatársaival egy kutatási projekt keretében vizsgáltuk az e-learning rendszerekben történő felhasználás módozatait. [89][52]

Biztosítani kell a választók anonimitását, a csalás elkerülését, vizsgálni kell a jogosultságot a választásra, garantálni kell, hogy egy személy egyszer szavazhasson, stb. Ennek elektronikus formában történő megvalósítása szintén nagy feladat a rendszerek tervezőinek. [79]

Digitális pénz - *Digital cash.* A digitális pénz problémaköre megoldandó kérdést vet fel. Az alkalmazott protokolloknak és titkosítási eljárásoknak garantálniuk kell az újrafelhasználhatóság megakadályozását, biztosítani kell a felhasználó anonimitását, a kétpartneres (harmadik fél nélküli) pénzcserét, stb. [128]

Összefoglaló

A bemutattokon kívül még számos más protokollt lehetne felsorakoztatni az egyes feladatok megoldására. Az Interneten protokolltárakat találunk, ahol különböző szempontrendszer szerint gyűjtik és osztályozzák a protokollokat. [137][10]

Az eddig bemutatott protokollok sokrétűek. Szerepelnek olyanok, amelyek titkos, nyilvános vagy éppen hibrid kriptorendszereket használtak. A támadások közül látunk olyat, amely a partnerek közé beékelődött aktív, vagy passzív támadót mutatott be. Szerepelt protokoll-lépéseket ismétlő támadó is. Több esetben módosított, javított protokollokat is bemutattunk. A szereplők számát tekintve két-, három-, vagy többszereplős protokollok fordultak elő.

Látható, hogy a protokollok világa összetett, a protokollok érzékenyek, a paraméterek helyes megválasztása alapvető kérdés, a lépések olyan apró rejtett hibákat tartalmazhatnak, amelyeket sok esetben igen nehéz észrevenni. Szükség van tehát a protokollok alapvető és átfogó vizsgálatára. A következő fejezetekben az kerül bemutatásra, hogy ezek a vizsgálatok milyen eszközökkel hajthatók végre.

3. fejezet

A kriptográfiai protokollok vizsgálati eszközei

Napjainkra a kriptográfiai kutatásoknak két jól elkülöníthető ága fejlődött ki. Az egyik a *formális* kutatások iránya, a másik a *számításelméleti* megközelítés. Az első ág a formális eszközök (modális logika, processz algebra, stb.) elemeit használja, a másik pedig a valószínűségszámítás és a komplexitáselmélet matematikáját emeli be a kriptográfiába. [5] Mindkét ágnak megvan a maga szerepe; más és más nézőpontból, ugyanakkor egymást kiegészítve vizsgálják a problémákat. Mindkét irány a matematika és a számítástudomány kapcsolódó ágaiból származtatható, és épít az ott elért eredményekre. A két nézőpont képviselői részben elhatárolódnak egymástól, de a különálló eredmények egységes szemléletbe ötvözése már megkezdődött. [3]

A következőkben tekintsük át röviden a kétféle megközelítés alapvető elemeit, az általuk közvetített vizsgálati módszertant. A pontosabb, tételszerű eredményeket a hivatkozott szakirodalomban találhatjuk meg.

3.1. Számításelméleti megközelítés

Ebben a fejezetben első lépésben a számításelmélet alapfogalmait tekintjük át röviden. Az ezt követő részben a kriptográfiai algoritmusokkal kapcsolatos eredményeket mutatunk be. A harmadik rész a számításelmélet és a kriptográfiai protokollok kapcsolatát elemzi. A fejezet legfőbb forrásául C. H. Papadimitriou [125] műve szolgált.

3.1.1. Számításelméleti alapfogalmak

A számításelmélet (bonyolultságelmélet) a matematika azon ága, amely azzal foglalkozik, hogy hogyan és milyen hatékonysággal lehet egyes problémákat megoldani. Az elmélet alapvető vonása az, hogy a számítási problémákat matematikai objektumokként kezeli. A vizsgált problémákkal kapcsolatban általában kétféle kérdéscsoport merül fe. Az úgynevezett *eldöntési problémák* (Létezik a problémának megoldása? Létezik algoritmus a megoldás előállítására, megtalálására? stb.) esetén a várt eredmény az *igen* vagy a *nem* válasz. Az *optimalizálási problémák, vagy függvényproblémák* esetén a válasz összetettebb, a megoldások különböző tulajdonságai is előtérbe kerülnek a vizsgálat során. A két problémakör természetesen egymással összefügg: *Bármely optimalizálási problémát átalakíthatunk egy vele nagyjából ekvivalens eldöntési problémává.* [125]

A bonyolultságelméleti vizsgálatok során a kutatók számítási modelleket használnak, amelyekben rögzítik a modell elemeit, a számítás során alkalmazható operációk körét és azok viszonylagos költségeit (időigény, tárigény). A legkorábbi modellek az 1930-as években születtek - K. Gödel, A. M. Turing, E. Post, S. C. Kleene, A. Chuch nyomán. Közülük A. M. Turing megközelítését (determinisztikus, nemdeterminisztikus, egyszalagos - egyszavas, többszalagos - többszavas, stb. Turing-gépek) használjuk napjainkban leggyakrabban. Ennek oka az, hogy egyszerű leírni a különböző Turing-gépekre vonatkozó szabályokat, a gépek jól modellezik a vizsgálandó problémákat, könnyen kialakíthatók az elméletben használt fogalmak, bonyolultsági osztályok.

Definíció 3.1.1.1 (Egyszalagos Turing-gép) *egy M Turing-gép az $M = (K, \Sigma, \delta, s)$ formális leírással adható meg, ahol K a gép állapotainak véges halmaza (utasítások), $s \in K$ a gép kezdőállapota, Σ a gép ábécéje (betűk véges halmaza), δ átmenetfüggvény: $K \times \Sigma \to (K \cup \{h, "igen", "nem"\}) \times \Sigma \times \{\leftarrow, \rightarrow, -\}$. K és Σ diszjunktak; Σ tartalmazza a $\sqcup$ (üres) és $\rhd$ (kezdet) szimbólumokat; h (megállási állapot), "igen" (elfogadó állapot), "nem" (elutasító állpot), $\leftarrow$ (lépés balra), $\rightarrow$ (lépés jobbra), $-$ (helyben maradni) szimbólumok nem elemei a $K \cup \Sigma$ halmaznak.*

Az egyik alapvető fogalomcsoport egy nyelv eldöntése és elfogadása.

Definíció 3.1.1.2 (Turing-gép eldönt egy nyelvet) *Legyen Σ betűk egy véges halmaza. Legyen $L \subseteq (\Sigma - \{\sqcup, \rhd\})^*$ egy nyelv, szavak halmaza ($\sqcup$ és $\rhd$ a Turing gép* üres *és* kezdet *jele) . Legyen M olyan Turing-gép, amely minden x bemenő szóra M(x)="igen", ha $x \in L$ és M(x)="nem", ha $x \notin L$ válaszokat adja. Ekkor azt mondjuk, hogy M eldönti az L nyelvet.*

Definíció 3.1.1.3 (Rekurzív, eldönthető nyelv) *Egy L nyelv rekurzív, vagy eldönthető, ha létezik olyan M Turing-gép, amely eldönti.*

Definíció 3.1.1.4 (Turing-gép elfogad egy nyelvet) *Az M Turing-gép felismeri, vagy elfogadja az L nyelvet, ha bármely $x \in (\Sigma - \{\sqcup, \rhd\})^*$ bemenő szóra $M(x) =$ "igen", ha $x \in L$; és M nem áll meg, azaz $M(x) = \nearrow$, ha $x \notin L$.*

Definíció 3.1.1.5 (Rekurzívan felsorolható nyelv) *Egy L nyelv rekurzívan felsorolható, ha létezik olyan M Turing -gép, amely elfogadja L-et.*

Definíció 3.1.1.6 (Turing-gép kiszámít egy f függvényt) *Egy M Turing-gép kiszámítja az $f : (\Sigma - \{\sqcup, \rhd\})^* \to \Sigma^*$ függvényt, ha minden $x \in (\Sigma - \{\sqcup, \rhd\})^*$ szóra $M(x) = f(x)$. Az f függvény rekurzív függvény, ha létezik M Turing-gép, amely kiszámítja.*

Értelmezhetők az egyszavas Turing-gépek $f(n)$ számítási sorozatai, amelyeket a bemeneti szó n hosszához lehet kapcsolni. A Turing-gépek pon-

tosabb tár- és időigény fogalmának megalkotásához szükséges a többszalagos Turing-gépek bevezetése, amelyek az egyszalagos gépek általánosításai. Bizonyítható, hogy a többszalagos Turing-gépek szimulálhatók egyszalagos gépekkel. Az elmélet legfontosabb ilyen irányú eredményei összefoglalva azt mondják ki, hogy nincs a Turing-gépeknek olyan bővítése, amely növelné az általuk eldöntött, vagy felismert nyelvek halmazát, vagy számítási sebességüket a polinomiális mértéknél jobban növelné.

Fontos elméleti eredmény a Turing-gépekkel kapcsolatban a Church-tézis, amely szerint bármely algoritmus megvalósítható Turing-géppel. Pontosabban: *„Az algoritmusok és időigényük matematikai eszközökkel való modellezésére tett bármely ésszerű kísérlet szükségképpen olyan modellhez és hozzá tartozó időigény-fogalomhoz vezet, amely polinomiálisan ekvivalens a Turing-géppel."* [125] Ugyanakkor a Turing-gép konstrukciójára alapozva könnyen modellezhetjük a ma használt számítógépeket, azok működését - közvetlen hozzáférésű gépek (*Random Access Machine - RAM*). Egy RAM-gép rendelkezik regiszterekkel, akkumulátorral, programszámlálóval, utasításkészlettel, címzési módokkal, stb. - hasonlóan a napjainkban használt számítógépekhez. [95]

Az előző modellezési körből kilépést jelentenek a nemdeterminisztikus Turing-gépek.

Definíció 3.1.1.7 (Nemdeterminisztikus Turing-gép) *Egy N nemdeterminisztikus Turing-gép egy $N = (K, \Sigma, \triangle, s)$ fomális leírással adható meg, ahol K, s, Σ megegyezik a determinisztikus Turing-gép hasonlóan jelölt fogalmaival. A $\triangle$ átmenetfüggvényt egy reláció írja le ($\triangle \subseteq (K \times \Sigma) \times [(K \cup \{h, "igen", "nem"\}) \times \Sigma \times \{\leftarrow, \rightarrow, -\}]$), amely szerint minden állapothoz egynél több következő lépés is tartozhat.*

A determinsztikus gépekhez hasonlóan a nemdeterminisztikus esetben is értelmezhető az, hogy az N gép eldönti az L nyelvet; a döntési struktúra számítási sorozatainak $f(n)$ hossza is vizsgálható (n a bemenet hossza) a determinsztikus esethez hasonlóan.

A számításelmélet leginkább vizsgált kérdései a bonyolultsági osztályokkal kapcsolatosak, amelyek az $f(n)$ függvény tulajdonságai alapján csopor-

tosítják a problémákat. Ezek az osztályok a matematikai objektumoknak tekintett problémák hasonló tulajdonságú körét foglalják magukba.

Egy bonyolultsági osztályt a következő paraméterek határoznak meg:

- **Számítási modell**. A leggyakrabban alkalmazott a többszalagos Turing-gépeken alapuló modell.
- **Számítási mód**. A legfontosabbak: determinisztikus és nemdeterminisztikus számítási mód.
- **Erőforrás**. A legfontosabbak: idő és tár.
- **Korlát**. Egy olyan f függvény (megengedett bonyolultsági függvény), amely pontosan megfogalmazza az erőforrás korlátozását.

Mindezek alapján egy **bonyolultsági osztály** azon nyelvek összessége, amelyek eldönthetők a választott módban működő M Turing-géppel úgy, hogy M minden $|x| = n$ hosszúságú bemeneten legfeljebb $f(n)$ erőforrás-egységet használ fel.

Az alapvető bonyolultsági osztályok a következők (f a megengedett bonyolultsági függvényt jelöli):

	deteminisztikus mód	nemdeterminisztikus mód
erőforrás: idő	TIME(f)	NTIME(f)
erőforrás: tár	SPACE(f)	NSPACE(f)

Látható, hogy a kutatás elsősorban az input méretének függvényében tanulmányozza a problémákat. Az egységes megközelítés a $O()$ jelölést (Landau-féle, vagy Bachmann-Landau jelölés) alkalmazza, amely a problémák aszimptotikus viselkedésének összehasonlítását teszi lehetővé - aszimptotikusan felső becslést adunk egy konstans-szorzótól eltekintve a korlátozott erőforrásra. A pontosabb leírás szerint legyenek s és t két $N \to M$ függvény (N és M a valós számok halmaza). $s(n) = O(t(n))$ azt fejezi ki, hogy léteznek olyan c és n_0 pozitív egész számok, hogy minden $n \geq n_0$ egész számra $s(n) \leq ct(n)$.[1]

[1] Szemléletesen: $s(n)$ legfeljebb olyan gyorsan nő, mint $t(n)$.

A vizsgálatok során f nem csak egy függvényt szokott képviselni, hanem függvények egy parametrizált családját. Az így kialakított bonyolultsági osztály a paraméterhalmaz által meghatározott függvények és az általuk értelmezett bonyolultsági osztályok egyesítését jelenti. A következőkben összefoglaljuk az ilyen módon értelmezett legfontosabb bonyolultsági osztályokat (n az input méretét jelöli, $j, k \in \mathbb{N}$):

- $\mathbf{P} = \bigcup_{j>0} TIME(n^j)$.
  Ez az osztály minden olyan döntési problémát tartalmaz, amely megoldható determinisztikus Turing-géppel, polinomiális számítási idő alatt. Ez a Cobhan-tézis szerint azt jelenti, hogy a probléma gyakorlatban kezelhető.
- $\mathbf{NP} = \bigcup_{j>0} NTIME(n^j)$.
  Nem determinisztikus Turing-géppel polinomiális időben eldönthető problémák osztálya. Tudjuk, hogy $P \subseteq NP$.
- $\mathbf{PSPACE} = \bigcup_{j>0} SPACE(n^j)$.
  Azon eldöntési problémák osztálya, amelyek megoldhatók determinisztikus Turing-géppel polinomiális mennyiségű tárat használva. A rendelkezésre álló idő nincs korlátozva.
- $\mathbf{EXP} = EXPTIME = \bigcup_{j>0} TIME(2^{n^j})$.
  Exponenciális időbonyolultságú problémák osztálya. Olyan problémák osztálya, amelyek determinisztikus Turing-géppel oldhatók meg $O(2^{n^k})$ időkorláttal.
- $\mathbf{L} = \bigcup_{j>0} SPACE(\log n)$.
  Logaritmikus mennyiségű tárkapacitás felhasználásával, determinisztikus Turing-géppel megoldható problémák osztálya.
- $\mathbf{NL} = \bigcup_{j>0} NSPACE(\log n)$.
  Logaritmikus mennyiségű tárkapacitás felhasználásával, nem determinisztikus Turing-géppel megoldható problémák osztálya.

Egyes bonyolultsági osztályok között tartalmazási relációk igazolhatók. Ezek közül néhány: $L \subseteq NL \subseteq P \subseteq NP \subseteq PSPACE \subseteq EXP$.

A számításelmélet (és a mai matematika) egyik legfontosabbnak ítélt problémája a P és NP osztályok közötti tényleges kapcsolat feltárása. [2] Ma még nem tudjuk, hogy a $NP \subseteq P$ kapcsolat teljesül-e. Amennyiben fennáll ez a reláció, úgy $P = NP$, ami számos, akár gyakorlati következményt is maga után von. Az elmélet ezeken a bonyolultsági osztályokon kívül még számos más osztályt is értelmez (például [163]-ben 488 bonyolultsági osztályt sorolnak fel).

Az egyik ilyen csoport a bonyolultsági osztályok komplementer osztályai. Tetszőleges C bonyolultsági osztály esetén coC jelöli az $\{\overline{L} : L \in C\}$ nyelvosztályt. Így értelmezhetők a coP, $coNP$, stb. osztályok.

Az elmélet egy másik irányzata a visszavezethetőség fogalmán keresztül ragadja meg a problémák nehézségének összehasonlítását. Két nyelv esetén definiálni lehet a (polinom idejű, determinisztikus Turing-géppel értelmezett) visszavezethetőséget, ami egy reláció a nyelvek között. Egy C bonyolultsági osztályon belül egy L nyelv $C - teljes$, ha minden C-beli nyelv visszavezethető L-re. Könnyen igazolható, hogy az eddig bemutatott osztályok egy része (P, NP, $coNP$, $PSPACE$, L, NL, EXP osztályok) zárt a visszavezetésre nézve. Érdemes olyan új osztályokat kialakítani, amelyekre visszavezethetők az adott osztály problémái - ezek lesznek a teljes osztályok ($P - teljes$, $NP - teljes$, $PSPACE - teljes$, stb. osztályok).

Újabb osztályokat képezhetünk a függvényprobléma fogalomból kiindulva. A számításelmélet eldöntési problémái igen/nem választ adhatnak a kérdésekre, a függvényproblémák megoldásai viszont ennél részletesebb válaszokkal szolgálnak. A megközelítés alapja az, hogy az eldöntési problémák és a függvényproblémák közötti kapcsolatok relációk segítségével formalizálhatók. Ilyen módon értelmezhető például a P és NP-beli nyelvekhez tartozó FP és FNP függvényproblémák osztálya. Igazolható, hogy ez a két osztály zárt a visszavezetésre nézve; valamint $FP = FNP$ akkor és csak akkor teljesül, ha $P = NP$.

[2] A P-NP probléma annak a hét milleniumi problémának az egyike, amelyet a Clay Mathematics Institute tűzött ki 2000-ben. A probléma eredeti megfogalmazói S. Cook és L. Levin, akik egymástól függetlenül ismerték fel a probléma fontosságát 1971-ben. [80]

3.1.2. Kiszámíthatóság és titkosítás

Az előző fejezetben röviden áttekintett osztályozási rendszer bemutatásának célja a kriptográfiai algoritmusok vizsgálatának előkészítése bonyolultságelméleti szempontból. A második fejezetben részletesebben tárgyaltuk a kriptográfia alapfogalmai között azt a modellt, amely leírja a titkos- és a nyilvános kulcsú kommunikációt. Ebben a fejezetben részben az ott bevezetett jelöléseket alkalmazzuk.

A nyilvános kulcsú titkosítás feltörésének nehézsége azon alapul, hogy milyen nehéz kiszámítani az m-et $E_{k_{pB}}(m)$-ből.[3] Azt tudjuk, hogy a feltételek teljesülése esetén m legfeljebb polinomiálisan hosszabb $E_{k_{pB}}(m)$-nál, így egy nyilvános kulcsú kriptográfiai rendszer FNP-beli probléma kell legyen. Jelenleg nem ismert, hogy FN és FNP kapcsolata milyen. Ebből és az előzőekből következik, hogy biztonságos nyilvános kulcsú kriptográfiai rendszer csak akkor létezhet, ha $P \neq NP$. Azonban $P \neq NP$ szükséges, de nem elégséges feltétel biztonságos nyilvános kulcsú kriptográfiai rendszer létezésére. A kérdés eldöntéséhez meg kell újra vizsgálnunk a már korábban definiált egyirányú függvény (2.1.4. fejezet) fogalmát.

Definíció 3.1.2.1 (Egyirányú függvény) *Az f szavakat szavakba képező függvényt egyirányú függvénynek nevezzük, ha*
- f injektív,
- $f(x)$ legfeljebb polinomiálisan hosszabb, vagy rövidebb x-nél,
- $f \in FP$, vagyis f polinom időben kiszámítható,
- $f^{-1} \notin FP$, vagyis az inverz függvény nem számítható ki polinom időben.

Amennyiben f^{-1} nem FP-ben van, akkor FNP osztályba kell essen. Az egyirányú függvényekhez kapcsolódó bonyolultsági osztály egy új osztály, amelyet UP-vel jelölünk. Ez az egyértelmű, polinom időkorlátos, nemdeterminisztikus Turing-géppel felismerhető nyelvek osztálya. Az UP osztállyal kapcsolatban kimutatható, hogy $P \subseteq UP \subseteq NP$. Igazolható az is, hogy $UP = P$ akkor és csak akkor teljesül, ha nem létezik egyirányú függvény.

[3] Az eddigi jelölések szerint m az üzenet, $E()$ a titkosító függvény, k_{pB} a fogadó B fél nyilvános kulcsa.

Látható, hogy abban az esetben, ha $P = NP$, nincsenek egyirányú függvények. Akkor viszont, ha $P \neq NP$ igaz, még az egyirányú függvények létezésének problémája nem oldódik meg.

Mindezek mellé még azt is fel kell sorolnunk, hogy a hagyományosnak nevezhető bonyolultsági fogalmak az algoritmusok leghosszabb futását kísérlik meg becsülni. Egy algoritmus viselkedésének kriptográfiai szemléletű tárgyalása közelebb áll az úgynevezett átlagos eset bonyolultságának vizsgálatához. Ennek a területnek a vizsgálata az 1970-es évek közepén indult. Pontosabb eredményeket [31]-ben találunk.

3.1.3. Kiszámíthatóság és kriptográfiai protokollok

A kriptográfiai protokollok is vizsgálhatók a számításelmélet eszközeivel. C. H. Papadimitriou szerint egy protokoll *„ ... egymással kölcsönhatásban levő számítási folyamatok olyan rendszere, amelyek tetszőlegesen bonyolult módon osztják meg bemenő és kimenő adataikat. Továbbá a számítások egy részének előírás szerint könnyűnek, más részének pedig nehéznek kell lennie."* [125]

A két tudományterület kapcsolatára jó példa a kriptográfiában használt interaktív bizonyítási rendszer és a nulla ismeretű bizonyítás, amelyek kapcsolódnak a számításelmélet NP osztályához.

Egy **interaktív bizonyítási rendszer** (*interactive proof system, interaktív protokoll*) egy absztrakt gép, amely két partner közötti üzenetváltásokat modellez. A rendszer résztvevői - Aliz az ellenőrző A fél és Botond a bizonyító B fél - üzeneteket váltanak annak eldöntése céljából, hogy egy megadott, mindkét fél által ismert és inputként értelmezett x karaktersorozat egy L nyelvhez tartozik, vagy sem. A bizonyító exponenciális algoritmust képes működtetni, míg az ellenőrző polinom idejű randomizált algoritmust. A kommunikáció utolsó üzenete igen/nem, amivel B jelzi A-nak, hogy jóváhagyja, vagy elutasítja a bemenetet.

Az interaktív bizonyítási rendszerek által eldöntött nyelvek osztályát IP-vel jelöljük. Bizonyítható, hogy $NP \subseteq IP$. Szintén belátható, hogy $IP = PSPACE$, ami kapcsolatot tár fel az interaktív bizonyítási rendszerek (interaktív protokollok) és a determinisztikus tárkorlátos gépek között.

Egy **nulla ismeretű bizonyítás** (*zero-knowledge proof, átlátszó bizonyítás*) olyan interaktív protokoll, amely a lefutáskor meggyőzi B-t, hogy A nagy valószínűséggel meg tud oldani egy problémát, és mindezt a protokoll anélkül éri el, hogy magáról a megoldás módjáról bármit elárulna B-nek. Ezzel kapcsolatban belátható, hogy az összes NP-beli problémának létezik nulla ismeretű bizonyítása. [69]

Mindezen eredmények azt mutatják, hogy a protokollok vizsgálata fontos szerepet játszik a bonyolultságelméletben (mint példák és problémakörök forrásai), a bonyolultságelmélet eredményei pedig hatással vannak protokollok vizsgálatára, amely gyakorlatban is használható eredményeket szolgáltathat.

3.2. Formális megközelítés

Az informatikában a formális módszerek olyan matematikai megalapozottságú technikák, amelyeket hardver- és szoftver-rendszerek fejlesztésére és ellenőrzésére alakítottak ki. A formális módszerek lehetővé teszik a minőségi követelmények pontos megfogalmazását, a kialakított rendszerek szisztematikus vizsgálatát, a megbízhatóság és hibatűrő képesség növelését, a matematikai precizitás előnyeinek bevonásával. [41] Ugyanakkor a formális vizsgálat a viszonylag magas költségek miatt főleg azokban az esetekben kerül alkalmazásra, amikor a megbízhatóság és a biztonság elsőrendűen fontos. A módszertan szélesebb körű alkalmazása iránti igény az informatikai rendszerek térhódításával egyre többször merül fel. A minőséggel kapcsolatos elvárások növekedése szintén a matematikai alapú helyességbizonyítás felé irányítja a figyelmet. [34][77][109][16]

A formális megközelítésnek számos bírálója akad. Sokszor hátrányként említik a magas költségeket, a leírás igen speciális jellegét (amelyet magasan képzett szakemberek képesek hatékonyan alkalmazni). Az absztrakt modellek használhatóvá tétele érdekében sokszor alkalmaznak természetes (emberi) nyelvet és informálisabb stílust - az olvashatóbb és könnyebben érthető

levezetés, indoklás kedvéért. Ez viszont alapja olyan bírálatoknak, hogy a formális specifikációban megkövetelt szigorúság veszti éppen el értelmét - az emberi nyelv félreérthetősége, a finom hibák elkerülhetőségének lehetetlensége erősődik meg.

A banki és pénzügyi szektorban megjelenő - formálisan tervezett, ellenőrzött - alkalmazások példája mutatja, hogy a fejlődés iránya a matematikai helyességbizonyítás elterjedése, egyre szélesebb körű alkalmazása. Hasonló tendenciák jelentkeznek napjainkban például az egészségügy területén is - orvosi, gyógyító protokollok, iránymutatások formális ellenőrzésének alakjában. Ezzel az irányzattal külön fejezetben foglalkozunk.

3.2.1. Kriptográfiai protokollok formális vizsgálata

A továbbiakban a formális ág fejlődését vizsgáljuk a kriptográfiai protokollok körében. A történeti vonatkozásokat is bemutató források közül C. A. Meadows [106][107][108]; Buttyán L. [42] közleményeit emelhetjük ki. B. Schneier [133] könyvének 3.3. fejezetében; G. Bella [17] könyve a 2. fejezetben ; P. Ryan, S. Schneider, M. Goldsmith, G. Lowe, A. W. Roscoe pedig [132] könyvük 9. részében foglalkoznak a témával. Az automatizált vizsgálatok legújabb eredményeit S. Mödersheim, L. Viganò, D. von Oheimb előadásvázlata [114] mutatja be.

Formális módszereket régóta használnak a kommunikációs protokollok vizsgálatára. [139] Az 1970-es évek végén, az 1980-as évek elején ezek a módszerek feltűntek a kriptográfiai protokollok ellenőrzése terén is. R. Needham és M. Schroeder már 1978-ban felvetette a protokollok vizsgálatának szükségességét.[4] A legelsőnek emlegetett érdemi megközelítés 1982-ben jelent meg (D. Dolev, S. Even, R. Karp [56]), majd egy év múlva D. Dolev és A. Yao tette közzé azt a közlemény [57], amely máig is hatással van a terület fejlődésére. Ez a munka lerakta az absztrakt állapotgép-modell alapjait, amely egy külön irányzatot képvisel a kriptográfiai protokollok formális analízisében.

[4] *"The need for techniques to verify the correctness of such protocols is great, and we encourage those interested in such problems to consider this area."* [118]

Sajnos ezek a törekvések akkoriban nem keltették fel a kutatók szélesebb körének érdeklődését, így egészen az 1990-es évek elejéig ezen a terülten nem történt lényeges változás. Az 1990-es évek elejére értek be azok a folyamatok, amelyek a tényleges áttörést hozták. Napjainkban legtöbbször az M. Burrows, M. Abadi, R. Needham által kialakított logikai rendszerre (BAN) [40] hivatkoznak, de ugyanakkor több hasonló közlemény jelent meg egy időben (P. V. Rangan 1988, L. Moser 1989, P. Bieber CKT5 1990 [29], P. Syverson 1990, R. Yahalom, B. Klein, T. Beth 1993 [160]). [106][42] A BAN rendszert egyszerűsége és könnyebb használhatósága népszerűbbé tette a többi rendszerrel szemben. Ezekkel az eredményekkel a modális logikán alapuló vizsgálatok felzárkóztak az absztrakt állapotgépekkel történő modellezés mellé.

Azóta sok kutató foglalkozott a kriptográfiai protokollok vizsgálatával, elemzésével. Amennyiben strukturáltan, csoportosítva akarjuk áttekinteni a fejlődést, azt többféle módon is megtehetjük.

•

Egy megközelítése a csoportosításnak a **tervezési és ellenőrzési fázisok szerinti besorolás**. [42] A három fő fázis a specifikáció, a konstrukció és a validáció. Ezek jellegzetességei és az itt felrajzolható főbb erővonalak a következők.

A. Specifikáció. A kriptográfiai protokollok tervezése azzal kezdődik, hogy meg kell határozni azokat a célokat és követelményeket, amelyeket a protokollnak teljesítenie kell. Ez azt is jelenti, hogy rögzíteni kell azt is, hogy a protokoll mikor tekinthető pontosnak, korrektnek. Kezdetekben a legfőbb követelménynek a tikosságot tekintették az alkotók. Később került előtérbe az azonosítás köre, amely már nem a titkosításhoz kapcsolódik közvetlenül. 1992-ben W. Diffie, P. van Oorschot és M. Wiener közleményükben pontosan rögzített elvárásokat fogalmaztak meg a hitelesítési protokollok helyességi kritériumaira. [54] T. Woo, S. Lam temporális logikát használva definiálta a hitelesítési protokollok követelményeit. [159] 1993-ban P. Syverson az Abadi-Tuttle-logikát bővítette temporális elemekkel hasonló céllal. [140] M. Bellare és P. Rogaway véletlen Turing gépekre

épülő modellt vezettek be 1994-ben. [21]

Egy másik megközelítés szerint a követelményeket a protokollokat leíró nyelvben lehet rögzíteni. A T. Woo és S. Lam által kidolgozott eseménysorokra alapozott követelményekhez hasonló megközelítést alkalmazott P. Syverson és C. Meadows az NRL Protocol Analyser megalkotásakor. [142]

B. Konstrukció. A protokollok kidolgozása, tényleges megalkotása során is alkalmazhatók formális módszerek. Ezek az eljárások jelentős költségcsökkentő lehetőséggel bírnának, hiszen alkalmazásukkal csökkenthető az utólagosan, a protokoll elkészülte után észlelt hibák száma. Azonban az ilyen alkalmazások száma még nem jelentős.

L. Gong, P. Syverson szigorú protokoll-tervezési irányelveket javasoltak 1995-ben. [72] Felvetésük egyik fontos pontja az, hogy az újabb és újabb észlelt hibák esetén nem alkalmazhatók a véget nem érő javítási folyamatok, hiszen szisztematikus és szigorú fejlesztési elvek és gyakorlat nélkül az újabb verziók ismét újabb hibákat eredményezhetnek. Bevezették az önellenőrző protokollok fogalmát. Ezt A. Keromytis és J. Smith általánosította 1996-ban. [83]

N. Heintze és J. Tygar 1996-ben egy moduláris fejlesztési megközelítését javasolta a protokollok fejlesztésének. [74] Ez a módszer a napjainkban erősödő, a különböző protokollok összekapcsolását irányzó vizsgálatok alapjának tekinthető.

Egy más megközelítését adta a protokollok fejlesztésének C. Meadows 1992-ben. [105] A fejlesztés során szinteket javasolt, amely a legfelsőbb, absztrakt elméleti szintről lebontva halad a részletezett specifikációig, vagy magát a protokoll lépéseket alakítja ki. Ehhez hasonló megközelítést alkalmazott és fejlesztett tovább J. Alves-Foss és T. Soule 1997-ben [8], valamint L. Buttyán, S. Staamann és U. Wilhelm 1998-ban [44]. További eredmény a protokollok formális megközelítésű konstrukciójával kapcsolatban M. Ababdi és R. Needham [2] közleménye 1994-ben. Ez utóbbi modellt Syverson bírálta 1996-ban [141] bemutatva a módszer korlátait.

C. Validáció. A kutatók legtöbb eredményt a már létező protokollok formális vizsgálata, validácója során érték el. C. Meadows 4 típust különített el közleményeiben, amelyek részletezve a következők.

C1. Általános célú specifikáló nyelvek és ellenőrző eszközök használata. A megközelítés alapja, hogy a kriptográfiai protokollok más elosztott rendszerhez, programhoz hasonlóan működnek, így hasonlóan kell a helyességbizonyítást is kezelni. Főbb eredmények címszavakban:

1989	R. Kemmerer - az Ina Jo bővített első-rendű predikátum kalkulusra épülő formális specifikációs nyelv használata.
1989	V. Varadharajan - absztrakt állapotgépek, állapot-diagrammok használata protokollok vizsgálatára.
1992	B. Nieh, S. Tavares - színezett Petri hálók alkalmazása.
1995	A. Roscoe, G. Lowe - CSP (Commnuicating Sequential Processes) modell és FDR (Failure Divergence's Refinement) ellenőrző eszköz alkalmazása. [131][96]
1995	E. Snekkenes - HOL (Higher Order Logic). [136]
1996	Convince: a HOL-ra épülő protokollvizsgáló rendszer. [37]
1997	Isabelle: HOL-ra épülő rendszer. [18][17][81]

Az 1990-es évek közepe utáni fejlődés bemutatására a következő osztályozási rendszer keretei között térünk ki.

C2. Szakértői rendszerek alkalmazása. A megközelítés alapja olyan rendszerek fejlesztése, amelyekben a protokoll tervezője különböző forgatókönyveket generál, majd megvizsgálja azok lefutását. A legtöbb ilyen rendszer alapja az absztrakt állapotgépek elmélete. Főbb eredmények:

1987	J. Millen, S. Clark, S. Freedman - Interrogator
1992	D.Longley, S. Rigby - Longley and Rigby Tool
1994	C. Meadows - NRL Protocol Analyzer

Az 1990-es évek közepe utáni fejlődés bemutatására szintén a következő osztályozási rendszer keretei között térünk ki.

C3. Modális logikai rendszerek használata. Az általunk alkalmazott vizsgálati módszer ebbe az osztályba sorolható, ezért a következő részben külön foglalkozunk ennek az osztálynak a fejlődésével és jellemzőivel.

C4. Algebrai alapú megközelítés. Az első hasonló alapú megközelítés szintén a már említett Dolev-Yao munkában [57] fedezhető fel. A támadó a modell szerint teljes körű felügyeletet gyakorol a kommunikációs hálózatra. A rendszer felfogható egy olyan gépnek, amellyel a támadó szavakat ír át rögzített szabályok szerint. Az algebrai megközelítés akkor fogja fel a protokollt biztonságosnak, ha a támadó nem képes bizonyos szavakat előállítani rendszerében. Ezt a modellt lehet a protokollok széles körére alkalmazni, így a kutatási ág egyik fontos feladata a vizsgálati kör bővíthetősége. Ennek során M. Merritt 1983-ban általánosította a Dolev-Yao modellt. Merritt munkáját M-J. Toussaint fejlesztette tovább. [152]

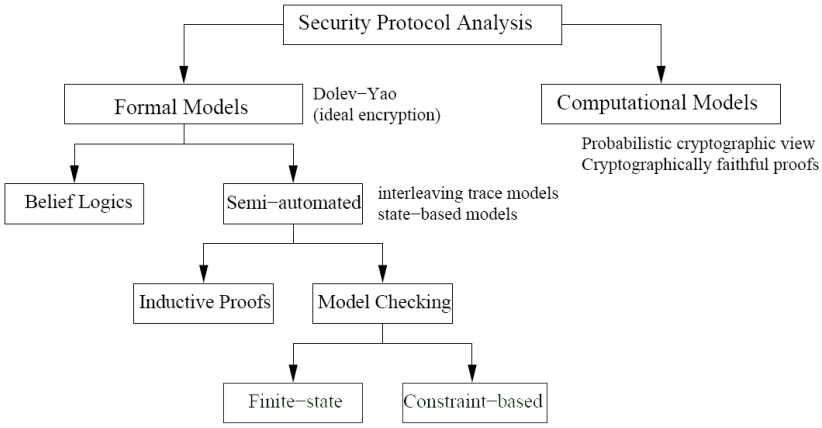
Ebbe a körbe sorolható még M. Abadi és A. Gordon munkája is [1], akik a *pi* kalkulust bővítették kommunikációs csatornák leírásának lehetőségével. Az új rendszert *spi* kalkulusnak nevezték el.

•

Egy másik szempontrendszer lehet az **alkalmazott modellező eszközök besorolása szerinti** osztályozás. Ennek a szempontrendszernek az ezredfordulón történő áttekintését adja meg C. A. Meadows. [106][108] Ennek a csoportosításnak a tagjai például az absztrakt állapotgépeken alapuló modellek, a modális logikai eszközök rendszerei, az algebrai alapokon nyugvó megközelítés és a típus-ellenőrzés módszere, stb.

Az osztályozás bővíthető az **automatizált ellenőrző eszközök** legújabb eredményeinek felsorolásával. Erről a [114] előadás egy összefoglaló képet nyújt, amit a 3.2.1. ábra foglal össze. Ki kell emelnünk az 1990-es évek végére megerősödő tendenciákat, amelyek ma is meghatározók. Ezt a kutatási irányt a fél-automata modellek (*Semi-automated models*) jelentik.

Az 1990-es évek közepén L. Paulson vezette be az induktív megközelítés



3.2.1. ábra Az alkalmazott modellező eszköz szerinti besorolás. [114]

modelljét (*inductive approach*). A modell fogalmi alapjai a már említett CSP (*Commnuicating Sequential Processes*) rendszerrel vannak kapcsolatban. A kulcselemek ebben a megközelítésben nyomvonalak (*traces*), olyan eseménysorozatok, amelyek megtörténhetnek a rendszerben a protokoll futása közben. A biztonsági tulajdonságok (titkosság, hitelesség) a nyomvonalakon értelmezett állításokként foghatók fel. Az, hogy egy bizonyos tulajdonság teljesül minden lehetséges nyomon, induktív módon bizonyítható (*inductive proofs*).[132]

A másik ág a modell-ellenőrzők köre. A vizsgált modellek (véges) állapotátmenetekkel rendelkező rendszerként foghatók fel ebben a megközelítésben. A rendszer tulajdonságai az állapotokon értelmezett relációkkal írhatók le. Az állapottéren végzett ellenőrzés megmutatja, hogy a vizsgált tulajdonságok kielégíthetők, vagy sem. A megközelítés korai verziói a *C*.2. pontban felsorolt rendszerek. Napjaink módszerei és eszközei: PVS, Coq, TAPS, Murphi, SMV, Maude, Athena, AVISPA. [132]

•

Mindkét osztályozásra jellemző, hogy a felosztáson belül, az egyes részcsoportok között nem lehet éles határvonalat húzni. Ezt a tendenciát már tudatosan is erősítették az 1990-es évek közepén megjelent kezdeményezések is, amelyek azt a célt tűzték ki, hogy a különálló modális logikára épülő irányokat vagy egy közös rendszer köré sorakoztassák fel (például a BAN köré [107]), vagy egy közös platformot létrehozva tegyék átjárhatóvá a rendszereket. Hasonló tendencia figyelhető meg az automatizált ellenőrző eszközök esetén is (például az NRL, vagy az AVISPA esetén).

Egy másik erős irányvonal az egymással adatokat cserélő eszközök általánossá válása (a bevezetőben már említett rejtett számítástechnika, *pervasive computing*), ami újabb kihívásokat jelent a protokollok területén is. Néhány újabb problémakör, amely a közelmúltban került előtérbe.

- Az eszközök már nem csak a hagyományos kliens/szerver struktúrába rendeződve működnek. Létrejöttek például a személyhez kötődő hálózatok (PAN, Personal Area Network), amelyek esetén szervertámogatás nélkül kell megvalósítani a biztonsági szolgáltatásokat.
- Új, nem hagyományos kriptográfiai primitíveket kell adaptálni a rendszerekbe. Példa: digitális aláírás, időpecsét, titokmegosztás, anonimitás, stb..
- Új típusú támadások kerültek előtérbe. Példa: számítógépes vírusok, kártevők, időalapú támadások, elárasztásos támadás (*denial service attack*), forgalomanalízis, stb.
- Újabb protokolligények lépnek fel. Példa: nyílt végű protokollok csoportos kommunikáció során, kulcs szétosztása előre nem meghatározható számú partner között.
- Igény a protokollok kompatibilitására és összekapcsolására. Egy összetett probléma megoldására már több protokollt kell alkalmazni. Például egy hagyományos titkosítási rendszerben a kommunikációt meg kell előznie a kulcsok cseréjének, ugyanakkor a kulcsok cseréjéhez sok esetben a partnereknek közös paraméterekkel kell rendelkezniük az induláskor (kriptográfiai inicializálás.)

Rengeteg ehhez hasonló problémakört lehetne még felsorakoztatni, amelyek

újabb és újabb kutatási irányokat jelentenek.[5] Hivatkozva erre a töretlennek látszó fejlődésre és arra, hogy a protokoll-modellek által lefedett biztonsági problémák a vizsgált esetekben az eldönthetetlen problémák körébe tartoznak ([59][74][47])[6], a mai és jövőbeli ellenőrző eszközeink valószínűleg nem jelenthetnek teljes megoldást. Mint a kutatás és fejlesztés számos más ágában, a kriptográfiai protokollok vizsgálata terén is mindig szükség lesz az emberi találékonyságra, leleményességre az előrelépés érdekében.

3.2.2. Modális logikai eszközök - A BAN-logika

Az előző fejezetben szó esett a formális ellenőrző eszközökön belül a modális logika alkalmazási területéről. Ennek a kutatási iránynak az első jelentős állomása a már említett *BAN*-logika volt, amely a szerzők nevének kezdőbetűiről kapta nevét. M. Burrows, M. Abadi és R. Needham az 1980-as évek végén dolgozta ki azt a formális rendszert, amely alkalmas kulcscsere és partner-hitelesítési protokollok vizsgálatára. [39][40]

A BAN-logika lehetővé teszi a protokollok formalizálását, az elérni kívánt célok pontos megfogalmazását, a szisztematikus protokoll-vizsgálatot. Legnagyobb érdeme azonban a modális logikai eszközök alkalmazásának újbóli elindítása.

A BAN-logika alapgondolata a kommunikáló felek közötti bizalom (hit, *knowledge*, *belief*) fejlődésének vizsgálata. A modellben a bizalom elért szintjét vetjük össze a megkívánt mértékkel. Ahhoz, hogy mindezt megfelelő matematikai precizitással tehessük meg, szükségünk van egy logikai nyelvre és következtetési szabályokra.

[5] A kriptográfiai protokollok vizsgálatában az 1990-es évek végére megerősödött elemek egy új kutatási szakaszt indítottak el. Tekinthetjük ezeket az időszakokat I. és II. generációs vizsgálati szakaszoknak. A hálózati megoldások elterjedése, a *rejtett számítástechnika* (*ubiquitous-, pervasive computing*) szintén új kutatási szakaszt jelent - III. generáció. Ezeknek a fejlődési szakaszoknak a pontosabb, megalapozottabb elkülönítése további vizsgálatokat igényel.

[6] Ezekben a vizsgálatokban az elemzők általában a protokollok egy formális modelljéből indulnak ki. A kialakított formalizmusokkal a protokollok egy-egy jól meghatározott osztálya vizsgálható. Az eldönthetetlenségi állítások (például *secret-security*, *time-security* fogalmakra [74]) és bizonyításaik ilyen körben érvényesek.

Modális logika alkalmazásának sémája a kriptográfiai protokollok vizsgálata során a következő:

1. **A vizsgált protokoll formalizálása.** A logikai nyelv segítségével le kell írnunk magát a protokollt, annak lépéseit; formális protokoll létrehozása.
2. **A kezdeti feltételek meghatározása.** Rögzíteni kell a kiindulási feltételeket, az induló paramétereket; környezeti változók definiálása.
3. **A protokoll céljainak megfogalmazása.** Meg kell határozni a következtetési szabályok alkalmazásával elérendő célállításokat.
4. **A logikai posztulátumok alkalmazása.** A logikai konstrukció célja a felek közötti bizalomra vonatkozó célállítások levezetése.
5. **A 4. lépés eseményeinek és a protokoll céljainak (3. lépés) összevetése.** Amennyiben sikerül a célállítás levezetése a modális logika eszközeinek felhasználásával, úgy bizonyítottnak tekintjük a protokollt az adott szempontra vonatkoztatva. Vannak olyan esetek, amikor a célállítás nem vezethető le, például ellentmondásra jutunk a kiinduló feltételekkel kapcsolatban. Ilyenkor a célállítást el kell vetnünk.

A kitűzött célok mindig valamilyen protokolltulajdonságot fednek (üzenetek, kulcsok titkossága, hitelessége, stb.). A célállítás elvetése azt is jelenti, hogy a protokoll egy kívánt célt nem tud megvalósítani, általában támadható is a vizsgált szempontból. Az, hogy a támadás milyen módon hajtható végre, a vizsgálatból általában közvetlenül nem származtatható, azonban az elemzés lényegesen megkönnyíti a további vizsgálatot.

A dolgozat **7.1. fejezetében**, a mellékletek között szerepel a BAN-logika részletes leírása. A felsorolt következtetési szabályok az egyes forrásokban más és más hangsúllyal (és esetleg más formában) szerepelnek. Az eltérések okaként felhozható a szabályok felsorolásának hosszadalmas volta - nem minden szabályt kell mindig alkalmazni a különböző vizsgálatok során. Másik okként említhető a logikai rendszer fejlődése, ami az arányok eltolódását, a súlyponti részek kiemelését jelenti, valamint azt, hogy az

esetleges hibák már nem kerülnek bele az újabb leírásokba. Ennek bizonyítására elegendő megvizsgálni két-három, a BAN-logika körébe tartozó leírást. Példaként hozhatjuk fel az eredeti BAN-cikk korrigált verzióját [40], egy magyar nyelven is elérhető leírás ([45] 8.7. fejezet). A módosítások ellenére mind BAN-logika néven futnak a különböző verziók.

A [40] közlemény nyolc protokoll elemzését tartalmazza, amelyek közül négyben találtak hibát a szerzők. A további alkalmazásra jellemző, hogy a legkülönfélébb területek protokolljai szerepelnek az elemzés tárgyaként. [92][155][7][115][138] Az is igaz, hogy megjelenése óta sokan bírálták a BAN-logikát. [100][36][98][162] Mint minden formális modell esetén, a BAN-logika használatakor is tisztázni kell a modell korlátait. A BAN-logika feltételezi, hogy

- a kriptográfiai primitívek (alapvető kriptográfiai építőelemek) tökéletesen működnek;
- minden résztvevő felismeri saját üzeneteit - ezzel a visszajátszásos támadások lényegében ki vannak zárva az elemzéskor;
- minden üzenet elegendő információt tartalmaz arra nézve, hogy a vevő meg tudja állapítani, hogy jó kulcsot használt-e a visszafejtés során;
- az idő egyszerűen leírható a folyamatokban;
- a protokoll résztvevői becsületesek - rosszindulatú protokollrésztvevő támadása nem detektálható;
- a bizalomra vonatkozó állítások nem változnak.

Azért, hogy válaszoljanak a felvetett hibákra és észrevételekre, M. Abadi és M. Tuttle átdolgozta és újraformálta a BAN-logikát; új szemantikát dolgozott ki rá. [4] Ehhez P. Syverson és P. van Oorschot újabb javító észrevételeket fűzött. [143]

A BAN-logikát számosan és sok irányban kibővítették. Néhány eredmény a következő. Az egyik irány az L. Gong, R. Needham és R. Yahalom által 1990-ben kifejlesztett GNY-logika. [71]. 1994-ben P. Syverson és P. van Oorschot a már említett közleményükben [143] a korábbi modelleket

(BAN, GNY, AT) próbálta egyesíteni. Ez a rendszer később az SvO-logika elnevezést kapta. A BAN-család (pontosabban a GNY logika) első automatizált támogatását S. H. Brackin készítette el HOL alapokon. [37]

Összefoglalva elmondhatjuk, hogy a BAN-logika olyan alapvető megközelítését adja a protokollvizsgálatnak, ami még ma is hatással van a tudományterület fejlődésére. A BAN-logikát követő rendszerek mind visszanyúlnak a BAN-logikáig, sok ebből származtatja magát, ezzel hasonlítja össze az elért eredményeket. Az általunk a továbbiakban részletesebben vizsgált CSN-logika is hasonló tulajdonságokkal bír, az alapok szintén a BAN-logikában gyökereznek.

3.2.3. A CSN-logika

A munkánk során kiindulási pontnak tekintett logikai rendszer az előző fejezetben felsorakoztatott alapokra támaszkodva, két részben jelent meg. Először 1997-ben T. Coffey és P. Saidha tette közzé az alaprendszert. [48] Ebben a cikkben a nyilvános kulcsú titkosítást használó protokollok számára kidolgozott elmélet látott napvilágot. Ezt követően 2003-ban T. Newe és T. Coffey [122] kibővíttette az eredeti modellt a szimmetrikus kulcsú titkosítást alkalmazó rendszerek körére. A forrásokban közölt CSN logikai rendszer matematikai logikai szempontból pontosításra szorul. A hiányosságok pótlására átdolgoztuk a CSN logikai rendszer egyes részeit. A szerzőkre utalva a továbbiakban is a **CSN-logika** nevet használjuk.

A CSN-logika egy többtípusú (fajtájú), multi-modális elsőrendű levezetési rendszer. Különböző rendszerek vizsgálata során többtípusú logika akkor használatos, ha a vizsgált objektumok nem alkotnak homogén halmazt. A többtípusú logikák lefordíthatók egytípusú, hagyományos elsőrendű logikává. A modális operátorok alkalmazásával az állítások eredeti jelentése módosítható. Egy modális logika eredetileg egy klasszikus logikai rendszer bővítése a *„szükségszerű”* és a *„lehetséges”* kifejezésekkel.[7] A

[7] Jelekben: a *szükségszerűség* operátora: $\Box$ a *lehetségesség* operátora: $\Diamond$. $\Box$ és $\Diamond$ egymásból kifejezhetők: $\Box\alpha \leftrightarrow \neg\Diamond\neg\alpha$ és $\Diamond\alpha \leftrightarrow \neg\Box\neg\alpha$ (α formula). A $\Box$ operátornak többféle értelmezése lehetséges. Ezek közül néhány: $\Box\alpha$ igaz, ha ... α *szükségszerű*en

CSN-rendszer többek között a K-val (*knowledge* - tud, ismer) és a B-vel (*belief* - hisz, bízik) operátorokat vezeti be, amivel egy multi-modális logikát hoz létre.

A levezetési rendszer egy klasszikus elsőrendű levezetési rendszerből indul ki, bővítve azt többek között az új operátorokra vonatkozó levezetési szabályokkal (például: $R2(a)$, $R2(b)$) és axiómákkal (például: $A1(a)$, $A1(b)$, $A2(a)$).

Egy másik osztályozás szerint a CSN-rendszer episztemikus-doxatikus[8] rendszer. [88] E szemléletmód szerint a rendszer kidolgozói abból indultak ki, hogy két tendencia figyelhető meg a kriptográfiai protokollok logikai vizsgálatában. Az egyik a bizalom/megbízhatóság felődésének vizsgálata a protokoll lépései során (*logics of belief*), a másik a protokollok működésére alapozott tudás (protokoll szereplőinek ismerete) elemzése (*logics of knowledge*). A kidolgozott új logika célja a kétféle megközelítés ötvözése, lehetővé téve ezáltal a kriptográfiai protokollok szélesebb körű és mélyebb vizsgálatát.

A következőkben részletesen ismertetjük a CSN alaprendszert, és annak egy kisebb bővítését (M. Kudo és A. Mathuria munkája [90]). Alapul minden esetben az említett forrásokat használtuk.[9] A leírásban szabad szövegként szerepel a logikai rendszer szándékolt értelmezésének megadása.

A logikai modellünk egyik kiinduló célja a partnerek közötti védett kommunikáció leírása (formalizálás). Ennek során legelőször a modell típusait kell megadnunk.[112]

A legegyszerűbb kommunikációs kapcsolat során egy küldő fél üzenetet küld egy fogadó fél felé. Ez alapján külön típusnak kell tekintenünk a szereplő partnereket (EGYED típus) és az átküldött üzenetet (ÜZENET típus). A védett kommunikáció a küldött üzenet titkos voltát jelenti, ame-

---

igaz; *tud*om, hogy α igaz; *ismer*etes, hogy α igaz; *hisz*em, hogy α igaz; α igaz *most, és a jövőben mindig* igaz lesz; stb. [60]

[8] angolul: *epistemic-doxastic*

[9] A 7.1. fejezetben szereplő BAN-logika részletes leírásában lábjegyzetként szerepeltetünk néhány olyan kapcsolódási pontot, amelyek a CSN-logika és a BAN-logika közös vonásait emeli ki. Az összevetés nem teljes, célja a CSN-logika és a BAN-logika szoros kapcsolatának bemutatása.

lyet kriptográfiai algoritmusok alkalmazásával érünk el. Az algoritmusok titkosító és visszafejtő kulcsokat használnak a működés során (KULCS típus). Le kell írnunk a vizsgálandó protokollok időbeli viselkedését, ami külön típus bevezetését jelenti (IDŐ típus).

A CSN-logikai rendszerhez tartozó nyelv a következő

$$L^{(CSN)} = \langle Sort, LC, Var, Con, Term, Form \rangle$$

rendezett hatos, ahol

Sort
A típusok (fajták) halmaza:

$$Sort = \{U, E, K, T\}$$

U üzenet-típus; E egyed-típus; K kulcs-típus; T idő-típus.

LC
A nyelv logikai konstansainak halmaza, amelyeket az elsőrendű logikában megszokott módon használunk.

$$LC = \{\neg, \rightarrow, \leftrightarrow, \wedge, \vee, \equiv, =, \forall, \exists, (,)\}$$

Var
A nyelv változóinak megszámlálhatóan végtelen halmaza. Minden változónak meghatározott típusa van. Var_δ a δ típusú változók halmazát jelöli:

$$Var = Var_U \cup Var_E \cup Var_K \cup Var_T$$

Con
A nyelv nemlogikai konstansainak legfeljebb megszámlálhatóan végtelen halmaza. Minden nemlogikai konstansnak meghatározott típusa van. Con_δ a δ típusú nemlogikai konstansok halmazát jelöli, egyes típusok esetén üres is lehet a halmaz:

$$Con = Con_U \cup Con_E \cup Con_K \cup Con_T$$

$F(0)_\delta$ a névkonstansok, $F(n)_\delta$ az n argumentumú függvényjelek halmaza. Az argumentumban szereplő szám a paraméterek számát jelöli. Függvényjelek esetén szokás megadni egy véges $\langle \delta_1, \delta_2, \ldots, \delta_n, \delta \rangle$ indexsorozatot is, amely rendre megadja a konkrét függvényjel n darab argumentumának típusát ($\delta_i \in Sort$) és a függvényjel típusát ($\delta \in Sort$).
$P(0)$ az állításkonstansok, $P(n)$ az n argumentumú predikátumkonstansok halmaza. Itt szintén szokás megadni az egyes predikátumkonstansok argumentumában szereplő $\langle \delta_1, \delta_2, \ldots, \delta_n \rangle$ ($\delta_i \in Sort$) indexsorozatot.

Term
A nyelv terminusainak, termjeinek halmaza, típusonként induktív definícióval megadva. $Term_\delta$ a δ típusú Termek halmazát jelöli, egyes típusok esetén üres is lehet a halmaz:

$$Term = Term_U \cup Term_E \cup Term_K \cup Term_T$$

Az induktív definíció általános formája minden δ típus esetén:

(a) $Var_\delta \cup F(0)_\delta \subseteq Term_\delta$.

(b) Ha $f \in F(n)_\delta$, $(n = 1, 2, \ldots)$ és $s_1, s_2, \ldots, s_n \in Term$, akkor $f(s_1, s_2, \ldots, s_n) \in Term_\delta$.

Form
A nyelv formuláinak halmaza, induktív definícióval megadva:

(a) $P(0) \subseteq Form$.

(b) Ha $s_1, s_2 \in Term_\delta$, akkor $(s_1 = s_2) \in Form$.

(c) Ha $P \in P(n)$, $(n = 1, 2, \ldots)$, és $s_1, s_2, \ldots, s_n \in Term$, akkor $P(s_1, s_2, \ldots, s_n) \in Form$.

(d) Ha $A \in Form$, akkor $\neg A \in Form$.

(e) Ha $A, B \in Form$, akkor $(A \to B), (A \wedge B), (A \vee B), (A \equiv B) \in Form$.

(f) Ha $x \in Var$, $A \in Form$, akkor $\forall x A$, $\exists x A \in Form$.

Kiegészítő részletek és az egyes típusokhoz tartozó sajátosságok a következők:

• i, j általános indexváltozókat jelölnek, a természetes számokon futnak.
• x, y, z általános változók, több típusra vonatkozó esetekben használjuk, megadva, hogy milyen típusú változókat képviselnek.
• A leírás során az egyértelműség érdekében sok esetben alkalmaznunk zárójeleket. Ezeket más matematikai tárgyak keretében megszokott módon kell olvasni, értelmezni.
• A következtetési szabályokban és az axiómákban a szabad változók univerzálisan kötöttek.

U - üzenet típus
Jellemzés: a kommunikációban szereplő üzenetek leírása; MSG az összes üzenetek halmaza, amely megszámlálhatóan végtelen halmaz.
• Var_U:
m, n, r, m_1, m_2, ..., m_i, m_j, ... általános üzenet-változók.
n_A, n_B, ..., n_Σ, ... speciális üzenet-változók - egyedi üzenetrészek jelölésére (általában az üzenet friss voltát hivatottak biztosítani, a visszajátszásos támadások kivédése érdekében).
r_A, r_B, ..., r_Σ, ... speciális üzenet-változók - általában véletlen számok jelölésére.
Az indexekben szereplő latin nagybetűk az üzenetet generáló egyedet jelöli.
• Con_U:
$F(0)_U$:

(a) A kommunikáció során átküldött bitsorozatok (1, 2, ... bájt - ASCI vagy Unicode kódolásként értelmezett bájtok) által reprezentált jelek, karakterek üzenet-konstansok.
(b) A kommunikáció során használt rögzített jelentésű karaktersorozatok (parancsok, utasítások): "enc", "dec", "0", "1", ... üzenet-konstansok. Ezek mindig dupla idézőjelek között szerepelnek, jelentésüket minden esetben megadjuk.

$F(n)_U$:

$\{m_1, m_2\}$	Konkatenáció: kapcsos zárójelbe írt, vesszővel elválasztott, egymás utáni üzenetekből újabb üzenetek származtathatók. $\{\} \in F(2)$; $\langle U, U, U \rangle$.

$E(m, ks_{(\Sigma,\Psi)})$ Titkosító függvény (*encryption function*) - szimmetrikus kulcsú tikosítás esete. $E(m, ks_{(\Sigma,\Psi)})$ jelentése: az m nyílt szöveg (üzenet) titkosítása a Σ és Ψ által használt osztott titkos $ks_{(\Sigma,\Psi)}$ kulcs segítségével.
A függvény kimenete üzenet fajta. $E \in F(2)$; $\langle U, K, U\rangle$.

$D(m, ks_{(\Sigma,\Psi)})$ Visszafejtő függvény (*decryption function*) - szimmetrikus kulcsú tikosítás esete. $D(x, ks_{(\Sigma,\Psi)})$ jelentése: az m titkosított üzenet visszafejtése a Σ és Ψ által használt osztott titkos $ks_{(\Sigma,\Psi)}$ kulcs segítségével. A függvény kimenete üzenet fajta. $D \in F(2)$; $\langle U, K, U\rangle$.

$e(m, k)$ Titkosító függvény (*encryption function*) - nyilvános kulcsú titkosítás során $e(m, k_\Sigma)$ jelentése: m üzenet titkosítása a k_Σ kulcs segítségével. $e(m, k_\Sigma^{-1})$ jelentése: az m üzenet digitális aláírása. A függvény kimenete üzenet fajta. $e \in F(2)$; $\langle U, K, U\rangle$.

$d(m, k)$ Visszafejtő függvény (*decryption function*) - nyilvános kulcsú titkosítás során $d(m, k_\Sigma^{-1})$ jelentése: az m üzenet visszafejtése a k_Σ^{-1} kulcs segítségével. $d(m, k_\Sigma)$ jelentése: digitális aláírás ellenőrzése, ha m digitálisan aláírt üzenet. A függvény kimenete üzenet fajta. $d \in F(2)$; $\langle U, K, U\rangle$.

$P(0)$: -

$P(n)$: -

• $Term_U$:

(a) Ha $m_1, m_2 \in Term_U$, akkor $\{m_1, m_2\} \in Term_U$.

(b) Ha $m \in Term_U$, $ks_{(\Sigma,\Psi)} \in Term_K$, akkor $E(m, ks_{(\Sigma,\Psi)}) \in Term_U$.

(c) Ha $m \in Term_U$, $ks_{(\Sigma,\Psi)} \in Term_K$, akkor $D(m, ks_{(\Sigma,\Psi)}) \in Term_U$.

(d) Ha $m \in Term_U$, $k_\Sigma \in Term_K$, akkor $e(m, k_\Sigma) \in Term_U$.

(e) Ha $m \in Term_U$, $k_\Sigma^{-1} \in Term_K$, akkor $d(m, k_\Sigma^{-1}) \in Term_U$.

• $Form$: -

Megjegyzések:

1. Az irodalmi források $ss_{(\Sigma,\Psi)}$ -vel jelölik a Σ és Ψ egyedek között megosztott (friss) titkot (*shared secret*), ami üzenet, vagy kulcs fajta lehet. $SS_{(\Sigma,\Psi)}$ jelöli Σ és Ψ egyedek között megosztott titkok halmazát.
2. Az EGYED, KULCS és IDŐ fajtájú változóknál értelmezünk kétkét olyan függvényjelet ($E2U(\Sigma)$, $U2E(m)$; $K2U(k)$, $U2K(m)$ és $T2U(t)$, $U2T(m)$), amely az egyedek, kulcsok és időpontok üzenetbe ágyazását (mint karaktersorozatok) és az onnan való kiemelését teszik lehetővé. Ezek a függvényjelek lényegében típus-konverziót hajtanak végre - kötött formátummal.

E - egyed típus

Jellemzés: a kommunikáció szereplőinek leírása. ENT az összes lehetséges egyedek halmaza. ENT számossága véges.

• Var_E:

Σ, Ψ, Γ, Λ, ...

• Con_E:

$F(0)_E$:

A, B, C, D, U, ... Az egyedek jelölése során alkalmazzuk a dolgozat 2.1.1. fejezetének *Jelölések* részében leírtakat. Ezek szerint A *Aliz*, B *Botond*, stb. szereplőket jelöli. Az egyedek elnevezése lehetőség szerint követi a hagyományos szerepköröket: kommunikáló felek A, B; passzív támadó E; abszolút megbízható fél T, stb. - az említett fejezetben leírtak szerint.

$F(n)_E$:

$E2U(\Sigma)$ Az egyed típusú változó átalakítása üzenet típusú változóvá. Ez a függvényjel lehetővé teszi a protokollok üzenetrészeiben egyedek szerepeltetését és továbbítását. A függvényjel input paramétere: egyed fajta, a függvényjel-output: üzenet fajta. A függvényjel helyettesítésekor adódó eredményt egyszeres idézőjelek közé írjuk: $E2U(\Sigma)='\Sigma'$. $E2U \in F(1)$; $\langle E, U \rangle$.

$U2E(m)$ A megfelelő üzenet fajta változó átalakítása egyed fajta változóvá. Ez a függvényjel lehetővé teszi egy protokollban az üzenetként átküldött egyednevek értelmezését.

A függvényjel input paramétere: üzenet fajta, függvényjel-output: egyed fajta. $U2E('\Sigma')=\Sigma$. $U2E \in F(1)$; $\langle U, E \rangle$.

$P(0)$: -

$P(n)$: -

• $Term_E$:

(a) Ha $\Sigma \in Term_E$, akkor $E2U(\Sigma) \in Term_U$.

(b) Ha $m \in Term_U$, akkor $U2E(m) \in Term_E$.

• $Form$: -

K - kulcs típus

Jellemzés: titkosító és visszafejtő kulcsok leírása. KEY jelöli az összes lehetséges kulcsok halmazát; KS jelöli az egyedek közötti szimmetrikus kulcsú kommunikációt lehetővé tevő kulcsok halmazát. Mindkét halmaz megszámlálhatóan végtelen számosságú.

• Var_K:

k, $k_{(\Sigma,\Psi)}$, k_Σ, k_Σ^{-1}, k_{t_i}, $k_{t_i}^{-1}$

k Általános kulcsváltozó.

$ks_{(\Sigma,\Psi)}$ (Ki)osztott titkos kulcs - (*shared secret key*). $ks_{(\Sigma,\Psi)}$ jelenti Σ és Ψ egyedek számára kiosztott, általuk ismert közös titkos kulcsot - szimmetrikus kulcsú tikosítás esete.

k_Σ Nyilvános kulcs (*public key*) - nyilvános kulcsú titkosítás során k_Σ a Σ egyed nyilvános kulcsa.

k_Σ^{-1} Titkos kulcs - (*secret key*) - nyilvános kulcsú titkosítás során k_Σ^{-1} a Σ egyed titkos kulcsa.

$k_{t_i}, k_{t_i}^{-1}$ Idő-kulcs - az indexben megadott t_i időponthoz kötött nyilvános és titkos kulcs.

• Con_K:

$F(0)_K$: -

$F(n)_K$:

$K2U(k)$ A kulcs fajta változó átalakítása üzenet fajta változóvá. Ez a függvényjel lehetővé teszi a protokollok üzenetrészeiben kulcsok szerepeltetését és továbbítását. A függvényjel input paramétere: kulcs fajta, a függvényjel-output: üzenet fajta.

A függvényjel helyettesítésekor adódó eredményt egyszeres idézőjelek közé írjuk: $K2U(k_\Sigma) =' k_\Sigma'$. $K2U \in F(1)$; $\langle K, U \rangle$.

$U2K(m)$ A megfelelő üzenet fajta változó átalakítása kulcs fajta változóvá. Ez a függvényjel lehetővé teszi egy protokollban az üzenetként átküldött kulcsok kulcsként való értelmezését. A függvényjel input paramétere: üzenet fajta, függvényjel-output: kulcs fajta. $U2K('k_\Sigma') = k_\Sigma$. $U2K \in F(1)$; $\langle U, K \rangle$.

$P(0)$: -

$P(n)$: -

• $Term_K$:

(*a*) Ha $k_\Sigma \in Term_K$, akkor $K2U(k_\Sigma) \in Term_U$.

(*b*) Ha $m \in Term_U$, akkor $U2K(m) \in Term_K$.

• $Form$: -

Megjegyzések:

1. A szakirodalmi források definiálják a $KS_{(\Sigma,\Psi)}$ halmazt, amely Σ és Ψ egyedek számára megfelelő, jó kulcsok halmazát (*set of good shared keys* - szimmetrikus kulcsú titkosítás) jelenti.

T - idő típus

Jellemzés: a protokollok időbeli leírása; $TIME$ jelöli az összes lehetséges időpontok halmazát. Ez a halmaz véges.

• Var_T:

t, t_1, t_2, ... t_i, t_j, ..., t', t'', ...

• Con_T:

$F(0)_T$:

(a) t_0 a vizsgált protokoll kezdetének időpontja.

(b) t_g egy protokollban előforduló kulcsgenerálás időpontja.

(c) τ időfeloldó protokollokban rögzített feloldási időpont.

$F(n)_T$:

$T2U(t)$ Az idő fajta változó átalakítása üzenet fajta változóvá. Ez a függvényjel lehetővé teszi egy protokoll üzenetrészében időadatok szerepeltetését. A függvényjel input paramétere: idő fajta, a függvényjel-output: üzenet fajta.

A függvényjel helyettesítésekor adódó eredményt egyszeres idézőjelek közé írjuk: $T2U(t_i) =' t_i'$. $T2U \in F(1)$; $\langle T, U \rangle$.

$U2T(m)$ A megfelelő üzenet fajta változó átalakítása idő fajta változóvá. Ez a függvényjel lehetővé teszi egy protokollban az üzenetként átküldött időadatok értelmezését. A függvényjel input paramétere: üzenet fajta, függvényjel-output: idő fajta. $U2T('t_i') = t_i$. $U2T \in F(1)$; $\langle U, T \rangle$.

$P(0)$: -

$P(n)$: -

• $Term_T$:

(a) Ha $t \in Term_T$, akkor $T2U(t) \in Term_U$.

(b) Ha $m \in Term_U$, akkor $U2T(m) \in Term_T$.

• $Form$:

(a) Ha $t_1, t_2 \in Term_T$, akkor $(t_1 < t_2) \in Form$.

Megjegyzések:

1. A protokoll-leírás során használt összes lehetséges időpontok $TIME$ halmaza lineárisan rendezett halmazt alkot.
2. Értelmezettek a $t_i \leq t_j$, $t_i > t_j$, $t_i \geq t_j$ formulák is.

Operátorok:

Mint már említettük az CSN-logika operátorai által az állítások eredeti jelentése módosul:

$K_{\Sigma,t}\Phi$ Hintikka-féle tudás, ismeret operátor - (*knowledge operator of Hintikka*). $K_{\Sigma,t}\Phi$ jelentése: Σ egyed ismeri (*knows*) a Φ állítást a t időpontban (részletesebben: [75][76]).

$B_{\Sigma,t}\Phi$ Hit operátor - (*belief operator*). $B_{\Sigma,t}\Phi$ jelentése: Σ egyed elhiszi, elfogadja (*believe*) a t időpontban, hogy a Φ állítás igaz.

$L_{\Sigma,t}x$ Tudás operátor - (*knowledge predicate*). $L_{\Sigma,t}x$ jelentése: Σ egyed ismeri és elő tudja állítani (*knows and can reproduce*) az x objektumot (üzenet vagy kulcs) a t időpillanatban.

$S(\Sigma, t, m)$ Kibocsátó operátor - (*emission operator*). $S(\Sigma, t, m)$ jelentése: Σ egyed az m üzenetet bocsátja ki, küldi a t időpontban.

$R(\Sigma, t, m)$ Fogadó operátor - (*reception operator*). $R(\Sigma, t, m)$ jelentése: Σ egyed az m üzenetet fogadja a t időpontban.

$C(x, y)$ Tartalmazás operátor - (*'contains' operator*). $C(x, y)$ jelentése: az x objektum (üzenet vagy kulcs) tartalmazza az y objektumot (üzenet vagy kulcs).

$A(\Sigma, t, \Psi)$ Hitelesítési operátor (*authentication operator*). $A(\Sigma, t, \Psi)$ jelentése: Σ egyed hitelesíti a Ψ egyedet a t időpontban.

$O_{\Sigma,t}(x, y)$ (*'obtain' operator*). $O_{\Sigma,t}(x, y)$ jelentése: az Σ egyed képes kinyerni, megkapni az y objektumot (üzenet vagy kulcs) az x objektumból (üzenet vagy kulcs) a t időpillanatban. [90] az operátort σ betűvel jelöli. Az áttekinthetőbb jelölés érdekében itt a O betűt használjuk.

Következtetési szabályok:
Legyenek α, β a nyelv tetszőleges formulái, p, q tetszőleges állításai. A logikai rendszer következtetési szabályai a következők:

R1 Az α és az $\alpha \rightarrow \beta$ formulák bizonyíthatóságából következik a β formula bizonyíthatósága:
$\alpha \wedge (\alpha \rightarrow \beta) \Rightarrow \beta$ *(modus ponens).*

R2(a) Az α formula bizonyíthatóságából következik a $K_{\Sigma,t}\alpha$ formula bizonyíthatósága:
$\alpha \Rightarrow K_{\Sigma,t}\alpha$ *(generelisation rule I).*

R2(b) Az α formula bizonyíthatóságából következik $B_{\Sigma,t}\alpha$ formula bizonyíthatósága:
$\alpha \Rightarrow B_{\Sigma,t}\alpha$ *(generalisation rule II).*

R3 Az $(\alpha \wedge \beta)$ formula bizonyíthatóságából következik az α formula bizonyíthatósága:
$(\alpha \wedge \beta) \Rightarrow \alpha$ *(simplification).*

R4 Az α és β formulák bizonyíthatóságából következik $\alpha \wedge \beta$ formula bizonyíthatósága:
(α) , $(\beta) \Rightarrow (\alpha \wedge \beta)$ *(conjunction).*

R5 Az α formula bizonyíthatóságából következik az $\alpha \vee \beta$ formula bizonyíthatósága:
$\alpha \Rightarrow (\alpha \vee \beta)$ *(addition).*

R6 A $\neg(\neg\alpha)$ formula bizonyíthatóságából következik az α formula bizonyíthatósága:
$\neg(\neg\alpha) \Rightarrow \alpha$ *(double negation).*

K1(a) A $K_{\Sigma,t}(p \wedge q)$ formula bizonyíthatóságából következik a $K_{\Sigma,t}p$ és $K_{\Sigma,t}q$ formulák bizonyíthatósága:
$K_{\Sigma,t}(p \wedge q) \Rightarrow K_{\Sigma,t}p \wedge K_{\Sigma,t}q$. [90]

K2(a) A $K_{\Sigma,t}p$ és $K_{\Sigma,t}q$ formulák bizonyíthatóságából következik a $K_{\Sigma,t}(p \wedge q)$ formula bizonyíthatósága:
$K_{\Sigma,t}p \wedge K_{\Sigma,t}q \Rightarrow K_{\Sigma,t}(p \wedge q)$. [90]

Axiómák:
A CSN-logikai rendszer axiómái két csoportba sorolhatók: az első a logikai axiómák köre (a CSN alaprendszerben 4 axióma, A1-A4); a másik típusú axiómák a nemlogikai axiómák (a CSN alaprendszerben 10 axióma, A5-A15), amelyek a nyilvános- és a titkos kulcsú kommunikáció körét foglalják magukba. Ezek az axiómák kapcsolódnak az üzenetek kibocsátásához és fogadásához, a üzenet-titkosítás és visszafejtés folyamatához.

A1(a) $K_{\Sigma,t}p \wedge K_{\Sigma,t}(p \rightarrow q) \rightarrow K_{\Sigma,t}q$
A 'modus ponens' szabály alkalmazása a K tudás operátorra.

A1(b) $B_{\Sigma,t}p \wedge B_{\Sigma,t}(p \rightarrow q) \rightarrow B_{\Sigma,t}q$
A 'modus ponens' szabály alkalmazása a B hit, elfogadás operátorra.

A2(a) $K_{\Sigma,t}p \rightarrow p$
Ha valami ismert, akkor az igaz (az axióma a tudás (K operátor) és a hit (B operátor) közötti különbséget fejezi ki, hasonló axióma B-re nincs).

A3(a) $L_{\Sigma,t}x \rightarrow \forall t_i \geq t \;\; L_{\Sigma,t_i}x$
A tudás predikátum monotonitása: amennyiben a tudás egyszer már birtokolt, akkor azt nem lehet elveszíteni. x kulcs vagy üzenet fajta változót jelöl.

A3(b) $K_{\Sigma,t}p \rightarrow \forall t_i \geq t \;\; K_{\Sigma,t_i}p$
A tudás operátor monotonitása: amennyiben a tudás egyszer már birtokolt, akkor azt nem lehet elveszíteni.

A3(c) $B_{\Sigma,t}p \rightarrow \forall t_i \geq t \;\; B_{\Sigma,t_i}p$
A hit operátor monotonitása: amennyiben a hit egyszer már birtokolt, akkor azt nem lehet elveszíteni.

A4(a) $L_{\Sigma,t}y \wedge C(y,x) \rightarrow \exists \Psi \in ENT \;\; L_{\Psi,t}x$
Ha egy üzenetrész egy másik üzenetrészből származik, akkor minden üzenetdarab, ami a konstrukcióban szerepel, ismert kell legyen valamely egyed által. x és y üzenet, vagy kulcs fajta.

A4(b) $C(x,x)$
A C operátor reflexív. [90] x üzenet, vagy kulcs fajta.

A4(c) $C(x,y) \wedge C(y,z) \rightarrow C(x,z)$
A C operátor tranzitív. [90] x, y és z üzenet, vagy kulcs fajta.

A4(d) $C(e(m,k_\Sigma),m) \wedge C(d(m,k_\Sigma^{-1}),m)$
Az m üzenetet tartalmazza minden olyan üzenet, amely az üzenet k_Σ kulccsal történő titkosításával és k_Σ^{-1} kulccsal történő visszafejtésével kapcsolatos. [90]

A5(a) $S(\Sigma,t,m) \rightarrow L_{\Sigma,t}m \wedge \exists \Psi \in ENT \backslash \{\Sigma\} \;\; \exists t_i \geq t \;\; R(\Psi,t_i,m)$
Kibocsátási axióma. Amennyiben a Σ egyed egy m üzenetet küld a t időpontban, akkor Σ ismeri az m üzenetet a t időpontban, valamint valamely Ψ egyed (Σ-n kívül) fogadja majd az m üzenetet egy t utáni t_i időpontban.

A6(a) $R(\Sigma,t,m) \rightarrow L_{\Sigma,t}m \wedge \exists \Psi \in ENT \backslash \{\Sigma\} \; \exists t_i \leq t \; S(\Psi,t_i,m)$
Befogadási axióma. Ha a Σ egyed fogad egy m üzenetet a t időpontban, akkor Σ ismeri az m üzenetet t időpontban, és valamely Ψ egyednek (Σ-n kívül) el kellett küldeni az m üzenetet a t-t megelőző t_i időpontban.

A6(b) $R(\Sigma, t, m_1) \wedge C(m_1, m_2) \wedge O_{\Sigma,t}(m_1, m_2) \rightarrow \exists \Psi \in ENT\ \exists t_i < t\ \exists m_3\ (S(\Psi, t_i, m_3) \wedge C(m_3, m_2) \wedge L_{\Psi,t_i} m_2 \wedge O_{\Sigma,t}(m_1, m_3) \wedge O_{\Sigma,t}(m_3, m_2))$
Ez az axióma A6(a) axióma általánosítása. [90] Amennyiben Σ egy olyan m_1 üzenetet kap, amelynek része a már általa ismert m_2 üzenet, akkor (mivel Σ nem küldhetett üzenetet magának, a rendszer ilyen üzenetek küldését nem teszi lehetővé) kell lennie egy korábbi m_3 üzenetnek (küldővel, küldési időponttal, stb. együtt), amely tartalmazta az m_2 üzenetet.

A7(a) $L_{\Sigma,t} m \wedge L_{\Sigma,t} k_\Psi \rightarrow L_{\Sigma,t} e(m, k_\Psi)$
Egy egyed képessége, hogy titkosítani tud egy üzenetet amennyiben ismeri a partner nyilvános kulcsát.

A7(b) $L_{\Sigma,t} m \wedge L_{\Sigma,t} k_\Sigma^{-1} \rightarrow L_{\Sigma,t} d(m, k_\Sigma^{-1})$
Egy egyed képessége, hogy vissza tud fejteni egy titkosított üzenetet, ha ismeri a (saját) titkos kulcsát.

A8(a) $\neg L_{\Psi,t} k_\Sigma \wedge \forall t_i \leq t\ \neg L_{\Psi,t_i}(e(m, k_\Sigma)) \wedge \neg(\exists n\ (R(\Psi, t_i, n) \wedge C(n, e(m, k_\Sigma)))) \rightarrow \neg L_{\Psi,t}(e(m, k_\Sigma))$
Egy üzenet titkosításának lehetetlensége helyes titkosító kulcs nélkül. Ha egy egyed nem ismeri a k_Σ kulcsot a t időpontban, és ha nem ismeri $e(m, k_\Sigma)$ titkosított üzenetet t időpont előtt, valamint üzenetet sem kap $e(m, k_\Sigma)$ tartalommal t_i időpontban, akkor az egyed nem ismeri $e(m, k_\Sigma)$ titkosított üzenetet a t időpontban.
[90]-féle módosítás az O operátor segítségével:
$\neg L_{\Psi,t} k_\Sigma \wedge \forall t_i \leq t\ \neg L_{\Psi,t_i}(e(m, k_\Sigma)) \wedge \neg(\exists n\ (R(\Psi, t_i, n) \wedge C(n, e(m, k_\Sigma)) \wedge O_{\Psi,t_i}(n, e(m, k_\Sigma)))) \rightarrow \neg L_{\Psi,t}(e(m, k_\Sigma))$

A8(b) $\neg L_{\Psi,t} k_\Sigma^{-1} \wedge \forall t_i \leq t\ \neg L_{\Psi,t_i}(d(m, k_\Sigma^{-1})) \wedge \neg(\exists n\ (R(\Psi, t_i, n) \wedge C(n, d(m, k_\Sigma^{-1})))) \rightarrow \neg L_{\Psi,t}(d(m, k_\Sigma^{-1}))$
Egy titkosított üzenet visszafejtésének lehetetlensége helyes visszafejtő kulcs nélkül. Ha egy egyed nem ismeri a k_Σ^{-1} titkos kulcsot a t időpontban, és ha nem ismeri t időpontot megelőzően a $d(m, k_\Sigma^{-1})$ visszafejtett üzenetet, valamint nem fogad üzenetet $d(m, k_\Sigma^{-1})$ tartalommal a t időpontban, vagy előtte, akkor az egyed nem ismeri a $d(m, k_\Sigma^{-1})$ visszafejtett üzenetet a t időpontban.

[90]-féle módosítás az O operátor segítségével:
$\neg L_{\Psi,t}k_{\Sigma}^{-1} \ \wedge \ \forall t_i \leq t \ \neg L_{\Psi,t_i}(d(m,k_{\Sigma}^{-1})) \wedge \ \neg(\exists n \ (R(\Psi,t_i,n) \ \wedge C(n,d(m,k_{\Sigma}^{-1})) \wedge \ O_{\Psi,t_i}(n,d(m,k_{\Sigma}^{-1})))) \ \rightarrow \neg L_{\Psi,t}(d(m,k_{\Sigma}^{-1}))$

A9(a) $L_{\Sigma,t}k_{\Sigma}^{-1} \ \wedge \ \forall \Psi \in ENT \backslash \{\Sigma\} \ \neg L_{\Psi,t}k_{\Sigma}^{-1}$
Kulcs titkossági axióma. A privát kulcsok használata a rendszerben csak a tulajdonosaik által lehetséges.

A10(a) $L_{\Sigma,t}(d(m,k_{\Sigma}^{-1})) \rightarrow L_{\Sigma,t}m$
Egy titkos kulcs tulajdonosa tudja használni a kulcsát, képes visszafejteni a titkosított üzeneteket.

A11(a) $L_{\Gamma,t}m \wedge L_{\Gamma,t}ks_{(\Sigma,\Psi)} \rightarrow L_{\Gamma,t}(E(m,ks_{(\Sigma,\Psi)}))$
Egy egyed képes titkosított üzenetet létrehozni a szimmetrikus kulcsú rendszerben, használva az általa ismert titkos kulcsot.

A11(b) $L_{\Gamma,t}m \ \wedge \ L_{\Gamma,t}ks_{\{\Sigma,\Psi\}} \rightarrow L_{\Gamma,t}(D(m,ks_{\{\Sigma,\Psi\}}))$
Egy egyed képes visszafejteni titkos üzenetet a rendelkezésére álló szimmetrikus kulcs segítségével.

A11(c) $L_{\Sigma,t}m \wedge O_{\Sigma,t}(m,n) \rightarrow L_{\Sigma,t}n$
Az O és az L operátor kapcsolata. Üzenetek felbontása.

A12(a) $\neg L_{\Gamma,t}ks_{(\Sigma,\Psi)} \wedge \forall t_i \leq t \ \neg L_{\Gamma,t_i}(E(m,ks_{(\Sigma,\Psi)})) \wedge \neg(\exists n \ (R(\Gamma,t_i,n) \wedge C(n,E(m,ks_{(\Sigma,\Psi)})))) \rightarrow \neg L_{\Gamma,t}(E(m,ks_{(\Sigma,\Psi)}))$
Ha valamely Γ egyed nem ismeri a $ks_{(\Sigma,\Psi)}$ kulcsot a t időpontban és nem ismeri t előtti időpontban a $E(m,ks_{(\Sigma,\Psi)})$ titkosított üzenetet és nem fogad $E(m,ks_{(\Sigma,\Psi)})$ üzenetrészt tartalmazó üzenetet t időpontban, akkor Γ nem ismeri a $E(m,ks_{(\Sigma,\Psi)})$ üzenetet t időpontban.

A12(b) $\neg L_{\Gamma,t}ks_{(\Sigma,\Psi)} \wedge \forall t_i \leq t \ \neg L_{\Gamma,t_i}(D(m,ks_{(\Sigma,\Psi)})) \wedge \ \neg(\exists n(R(\Gamma,t_i,n) \wedge C(n,D(m,ks_{(\Sigma,\Psi)})))) \rightarrow \neg L_{\Gamma,t}(D(m,ks_{(\Sigma,\Psi)}))$
Amennyiben valamely Γ egyed nem ismeri a $ks_{(\Sigma,\Psi)}$ kulcsot a t időpontban és a t időpont előtt nem ismeri a $D(m,ks_{(\Sigma,\Psi)})$ visszafejtett üzenetet, valamint nem kap $D(m,ks_{(\Sigma,\Psi)})$ üzenetrészt tartalmazó üzenetet a t időpontban, akkor Γ nem ismeri a $D(m,ks_{(\Sigma,\Psi)})$ üzenetet a t időpontban.

A13(a) $\forall \Gamma \in ENT \backslash \{\Sigma, \Psi\}\ \neg L_{\Gamma,t} ks_{(\Sigma,\Psi)} \wedge \exists \Lambda \in \{\Sigma, \Psi\}\ L_{\Lambda,t} ks_{(\Sigma,\Psi)} \rightarrow ks_{(\Sigma,\Psi)} \in \{KS_{(\Sigma,\Psi)}\}$
Csak az osztott titkos kulcs valódi tulajdonosai ismerik a kulcsot, és csak ők tudják, hogy a kulcsuk helyes kulcs.

A14(a) $\forall \Gamma \in ENT \backslash \{\Sigma, \Psi\}\ \neg L_{\Gamma,t} ss_{(\Sigma,\Psi)} \wedge \exists \Lambda \in \{\Sigma, \Psi\}\ L_{\Lambda,t} ss(\Sigma, \Psi) \rightarrow ss_{(\Sigma,\Psi)} \in \{SS_{\{\Sigma,\Psi\}}\}$
Csak az osztott titkos kulcs tulajdonosai ismerik a megosztott titkot, és csak ők tudják, hogy a megosztott titok ,,helyes" titok (*'good secret'*). Az axióma vonatkozik a titok frissességére is.

A15(a) $[A(\Sigma, t, \Psi) \rightarrow (L_{\Sigma,t} ss_{(\Sigma,\Psi)} \wedge ss_{(\Sigma,\Psi)} \in \{SS_{\{\Sigma,\Psi\}}\} \wedge R(\Sigma, t, m)) \wedge C(m, ss_{(\Sigma,\Psi)}) \wedge \forall t_i \leq t\ \neg S(\Sigma, t_i, m)] \rightarrow K_{\Sigma,t}(S(\Psi, t_i, m))$
Hitelesítési axióma - szimmetrikus forma. Ha Σ egyed hiteles partnernek fogadja el Ψ egyedet, akkor ha Σ ismeri a $ss_{(\Sigma,\Psi)}$ titkot, amit megoszt a Ψ egyeddel (a titok friss), és ez a titok ,,jó titok" (*'good secret'*), valamint Σ üzenetet kap (amit nem ő küldött) a t időpontban, amely tartalmazza a $ss_{(\Sigma,\Psi)}$ üzenetet, akkor Σ tudja azt, hogy Ψ küldte az üzenetet a t időpont előtt.

A15(b) $[A(\Sigma, t, \Psi) \rightarrow (L_{\Sigma,t} k_\Psi \wedge L_{\Sigma,t} m \wedge R(\Sigma, t, n) \wedge C(n, e(m, k_\Psi^{-1}))\ ] \rightarrow \forall t_i \leq t\ K_{\Sigma,t}(S(\Psi, t_i, n))$
Hitelesítési axióma - aszimmetrikus forma. Ha Σ egyed hiteles partnernek fogadja el Ψ egyedet, akkor ha Σ ismeri a Ψ egyed k_Ψ nyilvános kulcsát és az m üzenetet, és Σ az n üzenetet fogadja, amely tartalmazza az $e(m, k_{\Psi^{-1}})$ üzenetdarabot, akkor Σ tudja azt, hogy Ψ küldte az n üzenetet a t időpontot megelőző időpontban.

Az eddig felsorolt axiómák *(A1(a) - A15(b))* a [48] és [122] közleményekben megjelent rendszert foglalják össze. A következő axiómák *(TA1(a) - KM1(a))* a [90] közleményben megjelent bővítések és konkretizálások, amelyek az időfeloldó protokoll (következő, 4. fejezet) vizsgálatához szükséges elemeket tartalmazzák. Munkánk során a 4. fejezetben ezt a bővített, konkretizált rendszert használtuk a további eredmények elérése érdekében. Az 5. fejezetben visszatérünk az eredeti CSN-rendszerhez, és azt bővítjük a többcsatornás protokollok vizsgálatához szükséges összetevőkkel.

TA1(a) $\forall t < \tau\ L_{T,t}k_\tau^{-1} \wedge \forall \Sigma \in ENT\backslash\{T\}\ \neg L_{\Sigma,t}k_\tau^{-1}$
Az A9(a) axióma konkretizált változata. [90] A nyilvános kulcsú rendszerekben a megfelelő időpontokban a titkos kulcsot csak a kulcs tulajdonosa (itt a T generáló megbízható fél, a kulcskezelő szerver) ismerheti. k_τ, k_τ^{-1} nyilvános és titkos kulcsok, τ az időfeloldó titkosítás feloldásának rögzített időpontja (amikor a titkos kulcs elküldhető az üzenet fogadójának).

TA2(a) $L_{\Sigma,t}x \wedge L_{\Sigma,t}k_\tau \rightarrow L_{\Sigma,t}(e(x,k_\tau))$
Az A7(a) axióma konkretizált változata. [90] x üzenet fajta.

TA2(b) $L_{\Sigma,t}x \wedge L_{\Sigma,t}k_\tau^{-1} \rightarrow L_{\Sigma,t}(d(x,k_\tau^{-1}))$
Az A7(b) axióma konkretizált változata. [90] x üzenet fajta.

TA3(a) $\neg L_{\Sigma,t}k_\tau \wedge \forall t_i \leq t\ \neg L_{\Sigma,t_i}(e(m,k_\tau)) \wedge \neg(\exists n(R(\Sigma,t_i,n) \wedge C(n,e(m,k_\tau)) \wedge O_{\Sigma,t}(n,e(m,k_\tau)))) \rightarrow \neg L_{\Sigma,t}(e(m,k_\tau))$
Az A8(a) axióma módosított változata. [90]

TA3(b) $\neg L_{\Sigma,t}k_\tau^{-1} \wedge \forall t_i \leq t\ \neg L_{\Sigma,t_i}(d(m,k_\tau^{-1})) \wedge \neg(\exists n(R(\Sigma,t_i,n) \wedge C(n,d(m,k_\tau^{-1})) \wedge O_{\Sigma,t}(n,d(m,k_\tau^{-1})))) \rightarrow \neg L_{\Sigma,t}(d(m,k_\tau^{-1}))$
Az A8(b) axióma módosított változata. [90]

TA4(a) $L_{\Sigma,t}(e(m,k_\tau)) \rightarrow L_{T,t}k_\tau$
T ismeri az idő-kulcsokat. [90] .

TA5(a) $\forall \Sigma \in ENT\backslash\{T\}\ \forall t < \tau\ L_{\Sigma,t}m \wedge m = e(n,k_\tau) \wedge C(n,z) \rightarrow \neg O_{\Sigma,t}(m,z)$
Az időbizalmas adatokat csak T ismerheti a megadott τ időpont előtt. [90] x, y, z üzenet fajta.

KM1(a) $O_{i,t}(x,y) \wedge O_{i,t}(y,z) \rightarrow O_{i,t}(x,z)$
Az O operátor tranzitivitása. [90] x, y, z üzenet vagy kulcs fajta.

$\cdots\bullet\cdots$

Megjegyzések:

(M1) A típuskonverziót megvalósító függvényjelek ($E2U(\Sigma)$, $U2E(m)$; $K2U(k)$, $U2K(m)$; $T2U(t)$, $U2T(m)$) az egyes típusok üzenetbe ágyazását és az onnan történő kinyerését teszik lehetővé.

(M2) Az $A7(a)$ (konkretizált változat $TA2(a)$) és $A7(b)$ (konkretizált változat $TA2(b)$) axiómák írják le az üzenetek titkosítását és visszafejtését nyilvános kulcsú rendszerekben. Az e és d függvényjelek definiálása során szerepel az $e(m, k_\Sigma^{-1})$ kifejezés a digitális aláírás elkészítésére és a $d(m, k_\Sigma)$ kifejezés az aláírás ellenőrzésére. Az axiómák nem tartalmaznak közvetlenül utalást a digitális aláírás kezelésére. A következőkben feltesszük, hogy az egyedek képesek a digitális aláírás elkészítésére és annak visszafejtésére:
$L_{\Sigma,t} m \wedge L_{\Sigma,t} k_\Sigma^{-1} \rightarrow L_{\Sigma,t}\, e(m, k_\Sigma^{-1})$,
$L_{\Sigma,t}\, e(m, k_\Psi^{-1}) \wedge L_{\Sigma,t} k_\Psi \rightarrow L_{\Sigma,t}\, d(e(m, k_\Psi^{-1}), k_\Psi) = L_{\Sigma,t}\, m$.

(M3) A konkatenációval összekapcsolt üzenetdarabok szétbontásának lehetőségét tartalmazza az $A11(c)$ axióma. Az axiómában $O_{\Sigma,t}(m, n)$ hordozza magában a kapcsolódás jelölését. Mivel a protokollok leírásában sok esetben a konkatenációval egybefűzött üzenetek $\{n, r\}$ formában jelennek meg, ezért az axiómát a következő formában is értelmezzük:
$L_{\Sigma,t}\, m \wedge m = \{n, r\} \rightarrow L_{\Sigma,t}\, n \wedge L_{\Sigma,t}\, r$.

(M4) A következő fejezetekben konkrét protokollok leírása és elemzése történik. A protokollokban szereplő üzenetek leírása során az áttekinthetőség és a könnyebb olvashatóság érdekében egyszerüsítjük a jelölést. Minden olyan esetben, ahol nem okoz félreértést, elhagyjuk a típuskonverziót jelölő függvényjelet, csak az argumentum elemeit tüntetjük fel. Például egy A egyedet, k_Σ kulcsot és t időpontot tartalmazó üzenetet $\{E2U(A), K2U(k_\Sigma), T2U(t)\}$ helyett $\{A, k_\Sigma, t\}$-vel jelölünk. Ez a megoldás igazodik szakirodalomban alkalmazott jelölési szokásokhoz.

$\cdots \bullet \cdots$

A CSN-logikai rendszert több protokoll elemzésére is felhasználták. Ezek közül néhány:

2000 A 'Minimum-Knowledge Authentication Protocol' vizsgálata. [120]
2002 A Boyd-Park protokoll vizsgálata. [121]
2003 A letagadhatatlanságot biztosító protokollok vizsgálata. [49]
2003 A BCY mobil kommunikációs protokoll és variációinak vizsgálata [123]

2007 A CAPSL nyelv és a Coffey-Saidha logika összekapcsolása. [55]
2007 Hitelesített csoportos kulcskiosztás vizsgálata. [94]

A fenti axiómarendszert a továbbiakban két cél elérésére használjuk fel.

1. A negyedik fejezetben tovább vizsgáljuk a Kudo-Mathuria-féle *time release* protokollt. Kibővítjük és átalakítjuk az eredeti protokollt eddig nem alkalmazott felhasználási körre.
2. Az ötödik fejezetben többcsatornás protokollok formális vizsgálatához a CSN-logikát tekintjük kiindulási pontnak.

4. fejezet

A Kudo-Mathuria-féle időfeloldó protokoll vizsgálata a CSN logika eszközeivel

A fejezetben először a *time-release* probléma történetét és alkalmazási körét mutatjuk be. Ezt követően a Kudo-Mathuria-féle megoldást ismertetjük. A továbbiakban bővítjük az eredeti protokollt, majd az AVISPA rendszer alkalmazásával elért eredményeinket tekintjük át. [145][147]

4.1. A *time-release* probléma

Az idő igen fontos szerepet játszik a kriptográfiában. Érzékeny pontjai a kriptográfiai rendszereknek a támadó rendelkezésére álló idő (például nyers erő támadás esetén), a kriptográfiai rendszer időbeli viselkedéséből levonható következtetések (időalapú támadások, *side-channel attacks*), a titok időbeli értékcsökkenése, stb. Az idővel kapcsolatos problémakörök közül egy érdekes feladat a *time-release* titkosítás és az ehhez kapcsolódó protokol-

lok. Magyarul **idő-feloldó problémának** lehetne nevezni a feladatkört (a további rövidítések: *TRE - timed-release encryption*, *TRP - timed-release protocol*). Az alapprobléma célja *titkosított üzenet küldése a jövőbe*, vagyis olyan alkalmazás megalkotása, amely azt teszi lehetővé, hogy az üzenetet fogadó fél (és csak ő) csak egy előre meghatározott időpont után tudja visszafejteni a küldő fél titkosított üzenetét. Számos alkalmazási területtel kapcsolatba lehet hozni az alapproblémát, ezek közül címszavakban néhány példa. [130][78]

- Elektronikus szavazás - szavazatok késleltetett, rögzített időpont után történő felbontása.
- Lepecsételt, zárt árajánlatok - követelmény lehet, hogy az ajánlatok az ajánlattételi periódus lezártáig titkosak legyenek.
- Internet alapú, elektronikus lebonyolítású versenyek - a résztvevők számára ugyanabban az időpontban legyenek megismerhetők a kiírt feladatok.
- Érettségi rendszer - Külön kiemelhető a magyarországi érettségi rendszer, amelyben az ország minden középiskolájában adott időpontban, egyszerre kell elindítani az írásbeli érettségi lebonyolítását. A 2005. évben történt visszásságok megoldására lehetne alkalmazni a középiskolák ma már általánosnak tekintett Internet elérését, amely biztosíthatná a probléma TRE alapokon nyugvó megoldását.
- Távoktatás, elektronikus tanulmányi rendszerek.
- Szerződések aláírása egymásban nem megbízó felek között.
- Jövőben kihirdethető dokumentumok - memoárok, végrendeletek.
- Késleltetett kulcsfelvétel - letétbe helyezett kulcsok elérhetősége egy idő után.
- Online fogadás.

Az alapproblémára ma már több megoldási módot tudunk felmutatni. Ugyanakkor az alapkérdés is továbbfejlődött az anonimitás feltételének vizsgálatával, a passzív szerverek kialakításával, a többszörös időszerverek bevonásával, a költségek vizsgálatával, stb. Részleges képet kaphatunk a

tématerület jelenlegi állásáról a következő szerkezetben:

A probléma felvetését T. C. May nevéhez kapcsolják, akinek 1993-as összefoglaló elektronikus levelét [104] szokták kiindulópontnak tekinteni - ugyanakkor ő is több éves levelezésre hivatkozik.

Az első jelentősebb tudományos közlemény 1996-ban jelent meg. [130] Ebben az írásban R. L. Rivest, A. Shamir és D. A. Wagner két alapvető megközelítést vizsgált:

- Időzáras rejtvény (*time-lock puzzle*) használata - ami olyan probléma megalkotását jelenti, amely biztosan nem fejthető meg legalább egy meghatározott időtartamig.
- Megbízható ügynök, harmadik fél alkalmazása - ami a megbízható félre építve oldaná meg a problémát.

Ezek az irányok ma is érvényesek, többen is ugyanezeket az irányokat jelölik meg mint fő kutatási ágakat. A kétféle megközelítés értékelése és a tízegynéhány éves kutatási munka főbb eredményei címszavakban a következők:

1. megközelítés - kiszámíthatóság

A probléma megoldásához szükséges idő nagyban függ a számításokhoz felhasznált gépi eszközök képességétől, a probléma párhuzamosíthatóságától. A megközelítés elméleti alapjai a 3.1. fejezetben szerepelnek. A probléma ilyen irányú megközelítése lényegében egyes számításelméleti eredmények közvetlen gyakorlati alkalmazását jelentheti. Precíz időpontok és időtartamok rögzítésére még nem igazán alkalmasak a konstrukciók. A kiszámíthatósági megközelítés jelentősebb eredményei közül néhány időrendben, csak a kulcsszavakat kiemelve:

1978 R. C. Merkle - biztonságos kommunikáció vizsgálata nem védett csatornán - passzív támadó elleni védekezés - Merkle-féle multipuzzle rendszer kialakítása - a megoldás időigényének elemzése. [111]

1996 R. L. Rivest, A. Shamir, D. A. Wagner - time-release kriptográfia fogalma - két alapvető megközelítés - a Merkle-féle rendszer elemzése és bírálata - párhuzamos számítások lehetőségének elemzése - kulcsmérettől függő számítási idő - Blum-Blum-Shub véletlenszám generátor alkalmazásának vizsgálata - megbízható harmadik félre vonatkozó lehetőségek és követelmények összefoglalása. [130]

1996 M. Bellare, S. Goldwasser - az időbizalmas titkosítás mindennapi életben történő alkalmazása (politikai megközelítés: az egyének titokhoz való joga és az állami érdekek ellentéte) - a kulcs-letét problémakör elemzése - részleges kulcs-letét kidolgozása. [19][20]

2000 D. Boneh, M. Naor - időzített kötelességvállalás vizsgálata - a megoldás sajátosságai: ellenőrizhetőség, hitelesség, párhuzamos számítások elleni védelem. [33]

2001 W. Mao - R. L. Rivest, A. Shamir, D. A. Wagner rendszerének továbbfejlesztése és részletes vizsgálata. [99]

2002 J. Garay, M. Jakobsson - D. Boneh és M. Naor munkájának továbbvitele - időzített szerződéskötés vizsgálata - *time-line* fogalma: egymásból származtatott időelemek alkalmazása. [63]

2003 J. Garay, C. Pomerance - tükrözött *time-line* kidolgozása.[64]

2004 I. Y. Osipkov, J. H. Cheon - hitelesített idő-feloldó, nyilvános kulcs alapú titkosítás - valószínűségelméleti megközelítés. [124]

2. megközelítés - harmadik *megbízható* fél alkalmazása

Ez a megközelítés legtöbb esetben úgynevezett idő-szervert alkalmaz (*time-server*), amely a résztvevők számára közös és mindenki által elfogadott idővonatkoztatási pontot jelent. Ezekben a szerver-alapú protokollokban a felhasználók a megfelelő időpontban egy információdarabot (*trapdoor* - 2.1.4. fejezet) kapnak az idő-szervertől, aminek a segítségével vissza tudják fejteni az TRE kódot. A séma alkalmas pontos visszaállítási időpont beállítására, ugyanakkor szükséges az idő-szerverek üzemeltetése.

Ennek a megoldási módnak a fejlődése vetett fel sok olyan problémát, amelyek részekre osztották az eredeti kérdést, precízebb probléma-feltevésre sarkallták a kutatókat. Ez jelenti a kitűzött célok mellé újabbak felvételét,

például az anonimitás, passzív szerver; valamint hasonló kutatási irányok kialakítását.

A korai megoldásokban a szerver és a felhasználók között kétoldali üzenetcsere zajlik (aktív szerver). Ezekben az esetekben a szerver aktívan részt vesz a kódolási és visszafejtési folyamatban, ugyanakkor a felhasználói anonimitás nem érhető el. A későbbi protokollok megoldották a küldő fél anonimitását, az egyoldalú felhasználó-szerver kapcsolatot (passzív-szerver, a szerver feladata univerzális idő-specifikus *trapdoor*-ok közzététele). Célként tűzték ki a fejlesztők a szerver-felhasználó kommunikáció minimalizálását, elérve a skálázhatóságot és a felhasználói anonimitást. Megjelentek a problémakör megoldásánál az IBE (*Identity-Based Encryption*), az ECC (*Elliptic Curve Cryptography*) technikák. A *megbízható fél* megközelítés jelentősebb eredményei címszavakban:

1993 T. C. May - [104]

1996 R. L. Rivest, A. Shamir, D. A. Wagner - [130]

1999 G. Di Crescenzo, R. Ostrovsky, S. Rajagopalan - szerveralapú megoldás fejlesztése - a szerver csak a fogadó féllel van kapcsolatban, a küldő fél anonim maradhat - *conditional oblivious transfer* kidolgozása (újabb kriptográfiai primitív megalkotása). [50]

2001 D. Boneh, M. Franklin - az *Identity Based Encricption* (IBE - azonosítón alapuló titkosítás) séma egy megoldásának kidolgozása - ez sok új idő-bizalmas megoldás alapja. [32]

2003 M. C. Mont, K. Harrison, M. Sadler - IBE-n alapuló idő-feloldó rendszer - passzív idő-szerver kialakítása. [116]

2005 D. Boneh, X. Boyen, E.-J. Goh - HIBE (*Hierarchical IBE* - az IBE általánosítása) alkalmazása - fa struktúrájú megoldás - az aktuális idő-kulcsból (*time-specific trapdoor*) a régebbiek visszaállíthatók. [35]

2005 I. F. Blake, A. C.-F. Chan - skálázható passzív szerver - felhasználói anonimitás biztosítása. [30]

2007 D. Hristu-Varsakelis, K. Chalkias, G. Stephanides - felhasználói anonimitás - többszörös időszerver struktúra kialakítása. [78]

A *time-release* feladat egy megoldását mutatta be 1999-ben M. Kudo és A. Mathuria. [90] A megoldás nem jelent lényeges újítást, hiszen az szorosan a 2. megközelítéshez kapcsolódik. Az igazi eredmény a kidolgozott protokoll formális eszközökkel történő vizsgálata. A bizonyítás a már részletesen bemutatott CSN-logikát látja el az időtényezők megfogalmazását lehetővé tévő elemekkel. A továbbiakban bemutatjuk a Kudo-Mathuria-féle protokollt (jelöljük **K-M-P1** -gyel) és annak vizsgálatát, majd bővítjük az eredeti protokollt és újabb formális vizsgálatokat végzünk.

4.2. A Kudo-Mathuria-féle protokoll

A K-M-P1 protokoll úgy titkosít egy üzenetet, hogy azt a küldő fél kivételével egy megadott időpontig senki nem tudja elolvasni.

Az eredeti protokollnak három résztvevője van:[1] A az üzenetküldő fél, B az üzenet fogadója és T a megbízható ügynök. Jelölje az eddigieknek megfelelően $\{m\}_{k_A}$ az m üzenet titkosítását a címzett k_A nyilvános kulcsával. Ennek visszafejtését az $\{m\}_{k_A^{-1}}$ jelöli, ahol k_A^{-1} a címzett titkos kulcsa. Ez utóbbi az üzenet aláírását is jelentheti. A K-M-P1 protokoll lépései a következők:

1. lépés A üzenetet küld T-nek felkérve őt, hogy generáljon időkulcspárt egy jövőbeli t_8 időpontra (*"enc" - encryption*): $A \rightarrow T : "enc", t_8$.
2. lépés A üzenetének megkapása után T generál egy időkulcs-párt (nyilvános- és titkos kulcs): $k_{t_8}, k_{t_8}^{-1}$. T ezután aláírt üzenetet küld A-nak, amely tartalmazza a k_{t_8} időkulcsot. Ez később az üzenet titkosításához szükséges:

[1] Itt az Alice-Bob-féle jelölésrendszert (2.1.1. fejezet) alkalmazzuk, a formális leírásnál térünk át a CSN-rendszerre.

$T \rightarrow A : \{"enc", t_8, k_{t_8}\}k_T^{-1}$.
T a visszafejtő $k_{t_8}^{-1}$ kulcsot titokban tartja a megadott t_8 időpontig.

3. lépés A ellenőrzi T aláírását annak k_T nyilvános kulcsával. Amennyiben az aláírás megfelelő, A üzenetet küld B-nek. Az üzenet tartalma A neve, ami egyben felszólítás a tőle később érkező idő-feloldó üzenet fogadására és kezelésére.
$A \rightarrow B : A$.

4. lépés B válaszol A-nak, küld egy n_B véletlen számot:
$B \rightarrow A : n_B$.
Ez a szám a visszajátszásos támadás (*replay-attack*) ellen véd, az üzenet frissességét biztosítja.

5. lépés Ezután A generál egy r_A véletlen számot és elküldi B-nek a következő üzenetet: $A \rightarrow B$:
$\{\{m, r_A, A\}k_{t_8}, A, B, t_8, n_B, k_{t_8}\}k_A^{-1}, \{"enc", t_8, k_{t_8}\}k_T^{-1}$.
m az időbizalmas üzenet. r_A a titkosított szöveg egyediségét garantálja.

6. lépés B ellenőrzi A aláírását. Amennyiben az megfelelő, megerősítésképpen B aláírva visszaküldi az üzenet első részét:
$B \rightarrow A : \{\{\{m, r_A, A\}k_{t_8}, A, B, t_8, n_B, k_{t_8}\}k_A^{-1}\}k_B^{-1}$.

7. lépés Amikor B el akarja olvasni az A-tól kapott levelet, üzenetet küld T-nek (megadva a t_8 időpontot), kérve tőle a fejtő kulcsot (*"dec" - decryption*): $B \rightarrow T : "dec", t_8$.

8. lépés T várakozik az A által rögzített időpontig, majd elküldi B-nek a visszafejtő kulcsot: $T \rightarrow B : \{"dec", t_8, k_{t_8}^{-1}\}k_T^{-1}$.

Ezt a protokollt használva a B fél nem tudja visszafejteni az m üzenetet a meghatározott idő előtt. Ezen kívül a protokoll bizonyítja az A fél kilétét (partner-hitelesítés) B fél felé.

Ahhoz, hogy ezeket az állításokat pontosabban is megvizsgálhassuk, szükségünk van arra, hogy formalizáljuk a protokollt és annak kiinduló feltételeit. Erre az előbbiekben bemutatott CSN-rendszert alkalmazzuk. Ezután logikai vizsgálatokra alapozva fogadhatjuk el, vagy vethetjük el a protokollra vonatkozó állításokat.

A formalizált protokoll-lépések a következők:[2]

1. $S\Big(A, t_1, \{"enc", t_8\}\Big);\ R\Big(T, t_2, \{"enc", t_8\}\Big)$
2. $S\Big(T, t_2, e\big(\{"enc", t_8, k_{t_8}\}, k_T^{-1}\big)\Big);\ R\Big(A, t_3, e\big(\{"enc", t_8, k_{t_8}\}, k_T^{-1}\big)\Big)$
3. $S\Big(A, t_3, A\Big);\ R\Big(B, t_4, A\Big)$
4. $S\Big(B, t_4, n_B\Big);\ R\Big(A, t_5, n_B\Big)$
5. $S\Big(A, t_5, \Big\{e\big(\{e(\{m, r_A, A\}, k_{t_8}), A, B, t_8, n_B, k_{t_8}\}, k_A^{-1}\big),$
   $e\big(\{"enc", t_8, k_{t_8}\}, k_T^{-1}\big)\Big\}\Big);$
   $R\Big(B, t_6, \Big\{e\big(\{e(\{m, r_A, A\}, k_{t_8}), A, B, t_8, n_B, k_{t_8}\}, k_A^{-1}\big),$
   $e\big(\{"enc", t_8, k_{t_8}\}, k_T^{-1}\big)\Big\}\Big)$
6. $S\Big(B, t_6, e\big(e\big(\{e(\{m, r_A, A\}, k_{t_8}), A, B, t_8, n_B, k_{t_8}\}, k_A^{-1}\big), k_B^{-1}\big)\Big);$
   $R\Big(A, t_7, e\big(e\big(\{e(\{m, r_A, A\}, k_{t_8}), A, B, t_8, n_B, k_{t_8}\}, k_A^{-1}\big), k_B^{-1}\big)\Big)$
7. $S\Big(B, t_7, \{"dec", t_8\}\Big);\ R\Big(T, t_8, \{"dec", t_8\}\Big)$
8. $S\Big(T, t_8, e\big(\{"dec", t_8, k_{t_8}^{-1}\}, k_T^{-1}\big)\Big);\ R\Big(B, t_9, e\big(\{"dec", t_8, k_{t_8}^{-1}\}, k_T^{-1}\big)\Big)$

A protokollra vonatkozó állítások a következők:

G1. Csak a küldő A fél és a megbízhatónak tekintett T szerver képes visszafejteni az idő-bizamas üzenetet a megjelölt időpont előtt.

$$\forall t < t_8\ \forall \Sigma \in ENT \backslash \{T, A\}\ \ \neg L_{\Sigma,t} d(n, k_{t_8}^{-1})\ ,$$

$$ahol\ n = e(\{m, r_A, A\}, k_{t_8})\ .$$

[2] A leírás során alkalmazzuk az előző fejezet végén említett egyszerűsítést. Például az első lépésben $T2U(t_8) =' t_8'$ helyett t_8 szerepel. A zárójelek eltérő nagysága a protokoll-lépések olvasásának megkönnyítését célozza.

G2. A fogadó B fél vissza tudja fejteni az idő-bizalmas üzenetet a megjelölt időpont után. B a T szervertől kapott kulcsot használja a visszafejtéshez.

$$\forall t > t_8 \; L_{B,t} d(n, k_{t_8}^{-1}) \; ,$$
$$ahol \; n = e(\{m, r_A, A\}, k_{t_8}) \; .$$

G3. A fogadó B fél ismeri az idő-bizalmas üzenet eredetét és az üzenet protokollbeli útját.

$$\exists t \; t_0 < t < t_6 \;\; K_{B,t_6} S\big(A, t, d(r, k_A^{-1})\big) \; ,$$
$$ahol \; r = e(\{m, r_A, A\}, k_{t_8}), A, B, t_8, k_{t_8}, n_B \; .$$

A következő kiindulási feltételeket rögzítjük a protokollal kapcsolatban:
t_8 a visszafejtés rögzített, kívánt időpontját jelöli a jövőben (1. lépés);
t_g jelöli azt az időpontot, amikor T a $k_{t_8}, k_{t_8}^{-1}$ kulcspárt generálja;
t_0 a protokoll kezdetének időpontja.
m, m_1, m_2, m_3 üzenet-változók.

F1. Az A egyed megbízhatóan kezeli az időbizalmas adatok titkosítását.
$\forall t < t_8 \; \forall m_1 \; S(A, t, m_1) \wedge C(m_1, d(m_2, k_{t_8}^{-1}))$
$\rightarrow m_1 = e(m_3, k_{t_8}) \wedge C(m_3, d(m_2, k_{t_8}^{-1}))$

F2. A T egyed nem küld titkosított idő-bizalmas üzenetet üzenetrészként egy üzenetben sem a meghatározott idő előtt.
$\forall t < t_8 \; \neg \exists m_1 \; \big(S(T, t, m_1) \wedge C(m_1, d(m_2, k_{t8}^{-1}))\big)$

F3. T az időbizalmas kulcsokat a megadott időpont előtt generálja.
$t_g < t_8$

F4. Az idő-kulcspár titkos része nem használható annak létrehozása előtt.
$\forall t < t_g \; \forall \Sigma \in ENT \backslash \{A\} \; \neg L_{\Sigma,t} d(m, k_{t_8}^{-1})$

F5. Senki sem ismerheti a publikus idő-kulcsot annak létrehozása előtt.
$\forall t < t_g \; \forall \Sigma \;\; \neg L_{\Sigma,t} k_{t_8}$

F6. A protokoll kezdetén B tudja, hogy ismeri A és T nyilvános kulcsát.
$K_{B,t_0} L_{B,t_0} k_A$; $K_{B,t_0} L_{B,t_0} k_T$

F7. A protokoll kezdetén B tudja, hogy senki sem ismerheti az n_B elemet a protokoll indulása előtt.
$\forall t < t_0 \; \forall \Sigma \;\; K_{B,t_0} \neg L_{\Sigma,t} n_B$

A $G1.$, $G2.$, $G3.$ protokoll-célok formalizálását és bizonyítását M. Kudo és A. Mathuria a [90] cikkben jelentette meg. A $G2.$ állítás bizonyítását megismételjük és egy újabb bizonyítást is bemutatunk, mivel a saját eredmények igazolásához a $G2.$ tétel bizonyítási sémáját tudjuk felhasználni.

Tétel 4.2.1 (G1.) $\forall t < t_8 \; \forall \Sigma \in ENT \backslash \{T, A\} \;\; \neg L_{\Sigma,t} d(n, k_{t_8}^{-1})$,
ahol $n = e(\{m, r_A, A\}, k_{t_8})$.

G1. Bizonyítás Az állítás teljes indukcióval bizonyítható. Felhasználásra kerül a $TA3(b)$ axióma és annak kontrapozíciója. Részletes levezetés [90]-ben található.

Tétel 4.2.2 (G2.) $\forall t > t_8 \; L_{B,t} d(n, k_{t_8}^{-1})$, *ahol* $n = e(\{m, r_A, A\}, k_{t_8})$.

G2.1. Bizonyítás A $G2.$ tétel bizonyítása ([90] alapján).
Legyen $t > t_8$. Amennyiben B az 5. és 8. lépésben megkapja a protokoll szerinti üzeneteket, akkor a következő állításoknak igaznak kell lenniük: Az $n = e(\{m, r_A, A\}, k_{t_8})$; $m_1 = \{A, B, t_8, n_B, k_{t_8}\}$ és $m_2 = e(\{\text{"}enc\text{"}, t_8, k_{t_8}\}, k_T^{-1})$ jelölésekkel

$$K_{B,t_6} R\Big(B, t_6, \big\{e(\{n, m_1\}, k_A^{-1}), m_2\big\}\Big) \tag{1}$$

$$K_{B,t} R\Big(B, t, e(\{\text{"}dec\text{"}, t_8, k_{t_8}^{-1}\}, k_T^{-1})\Big) \tag{2}$$

(1) és (2) digitális aláírást tartalmaz, amiről feltehető, hogy egy aláírt üzenet tartalmazza az üzenetet és hozzá kapcsolva az aláírást magát.

(1), (2)-ből kiindulva - és n kifejtése szerint - a következő állításoknak igaznak tehetjük fel:

$$K_{B,t_6}R(B, t_6, e(\{m, r_A, A\}, k_{t_8})) \tag{3}$$
$$K_{B,t}R(B, t, k_{t_8}^{-1}) \tag{4}$$

(3), (4)-ből az $A2(a)$ axióma alkalmazásával következik:

$$R(B, t_6, e(\{m, r_A, A\}, k_{t_8})) \tag{5}$$
$$R(B, t, k_{t_8}^{-1}) \tag{6}$$

Az $A6(a)$ axióma felhasználásával:

$$L_{B,t_6}e(\{m, r_A, A\}, k_{t_8}) \tag{7}$$
$$L_{B,t}k_{t_8}^{-1} \tag{8}$$

Az (7) és $A3(a)$ alapján:

$$L_{B,t}e(\{m, r_A, A\}, k_{t_8}) \tag{9}$$

(8), (9) és $TA2(b)$ ($A7(b)$) pedig a kitűzött állításhoz vezet:

$$L_{B,t}d(e(\{m, r_A, A\}, k_{t_8}), k_{t_8}^{-1}). \square$$

G2.2. Bizonyítás A $G2.$ tételre az $(M2)$ és $(M3)$ megjegyzések alapján új bizonyítást adunk.
Legyen $t > t_8$; $n = e(\{m, r_A, A\}, k_{t_8})$, $m_1 = \{A, B, t_8, n_B, k_{t_8}\}$ és $m_2 = e(\{"enc", t_8, k_{t_8}\}, k_T^{-1})$. Hasonlóan az előző bizonyításhoz, az 5. és 8. protokoll-lépés alapján:

$$K_{B,t_6}R\Big(B, t_6, \big\{e(\{n, m_1\}, k_A^{-1}), m_2\big\}\Big)\,, \tag{1}$$
$$K_{B,t}R\Big(B, t, e(\{"dec", t_8, k_{t_8}^{-1}\}, k_T^{-1})\Big)\,. \tag{2}$$

Az $A2(a)$ axiómát felhasználva adódik:

$$R\Big(B, t_6, \big\{e(\{n, m_1\}, k_A^{-1}), m_2\big\}\Big)\ , \tag{3}$$

$$R\Big(B, t, e(\{"dec", t_8, k_{t_8}^{-1}\}, k_T^{-1})\Big)\ . \tag{4}$$

$A6(a)$ axióma felhasználásával:

$$L_{B,t_6}\big\{e(\{n, m_1\}, k_A^{-1}), m_2\big\}\Big)\ , \tag{5}$$

$$L_{B,t}\ e(\{"dec", t_8, k_{t_8}^{-1}\}, k_T^{-1})\Big)\ . \tag{6}$$

$(M3)$ és 5 szerint

$$L_{B,t_6} e(\{n, m_1\}, k_A^{-1}) \tag{7}$$

$F6.$ és $A2(a)$ szerint:

$$L_{B,t_0} k_A \tag{8}$$

$$L_{B,t_0} k_T \tag{9}$$

$A3(a)$-t alkalmazva:

$$L_{B,t_6} k_A \tag{10}$$

$$L_{B,t} k_T \tag{11}$$

A 3. fejezet $(M2)$ megjegyzésére támaszkodva (7) és (10)-ből

$$L_{B,t_6}\{n, m_1\} \tag{12}$$

Ugyanígy (6) és (11) adja, hogy

$$L_{B,t}\{"dec", t_8, k_{t_8}^{-1}\} \tag{13}$$

$A3(a)$ és (12) adja, hogy

$$L_{B,t}\{n, m_1\} \tag{14}$$

(13) és (14)-re alkalmazva $(M3)$-t, az $A11(c)$ axióma konkretizált alakját:

$$L_{B,t}k_{t_8}^{-1} \tag{15}$$
$$L_{B,t}n \tag{16}$$

(15), (16) a $TA2(b)$ axiómával adja a keresett állítást:

$$L_{B,t}d(n, k_{t_8}^{-1}),\ ahol\ n = e(\{m, r_A, A\}, k_{t_8}).\ \square$$

Tétel 4.2.3 (G3.) $\exists t\ t_0 < t < t_6\ \ K_{B,t_6}S\big(A, t, d(r, k_A^{-1})\big)$,
ahol $r = e(\{m, r_A, A\}, k_{t_8}), A, B, t_8, k_{t_8}, n_B$.

G3. Bizonyítás Részletes levezetés [90]-ben található.

A következőkben a K-M-P1 protokollt további vizsgálatoknak vetjük alá.

4.3. A Kudo-Mathuria-féle protokoll módosításai és a módosítások vizsgálata a CSN logika eszközeivel

4.3.1. Passzív támadás

A lehallgatás a kriptográfiai protokollok egyik gyakori támadási módja (2.1.2. fejezet). Amennyiben ezt a kérdéskört akarjuk tanulmányozni, akkor figyelembe kell vennünk, hogy az axiómák és más feltétezések közvetlenül nem foglalják magukba az E támadó (lehallgató) szerepét. Bővítenünk kell tehát a protokollban résztvevő egyedek körét E-vel, és rögzítenünk kell azt, hogy E minden más szereplő üzeneteit lehallgathatja. E a lehallgatott üzeneteket fel tudja használni úgy, mintha ő kapta volna azokat közvetlenül. Ezt a tényt a K-M-P1 protokoll esetén a következő feltétellel fogalmazhatjuk meg:

F8. $\forall t\ \ \forall i \in ENT\ \ K_{E,t}R(i, t, x)\ \wedge\ L_{E,t}x$.

E szintén ismeri és használni tudja a szereplők nyilvános kulcsait:

F9. $K_{E,t_0} L_{E,t_0} k_A \; ; K_{E,t_0} L_{E,t_0} k_T$.

Az E passzív lehallgató a protokoll végén tehát ugyanazon információkkal rendelkezik, mint a B fél. Így E szintén megszerezheti az időbizalmas információt A-tól. Ehhez az 5. és 8. lépés lehallgatása elégséges. Ez az eredeti protokoll szempontjából nem tekinthető hibának, hiszen az alapfeladat nem köti ki az ilyen irányú biztonságot. Vizsgáljuk tovább a K-M-P1 protokollt ebből az irányból, a CSN-logika alkalmazásával.

Tétel 4.3.1 (G4.) *A K-M-P1 protokoll futása során az E támadó - aki lehallgatja az üzeneteket - ugyanolyan információkkal rendelkezik, mint B a t időpontban, a t_8 időpont után. Vagyis E szintén képes visszafejteni az idő-bizalmas üzenetet.*

$$\forall t > t_8 \;\; L_{E,t} d(n, k_{t_8}^{-1}) \; , ahol \; n = e(\{m, r_A, A\}, k_{t_8}) \; .$$

G4. Bizonyítás A bizonyítás hasonló az eredeti Kudo-Mathuria cikk [90] $G2.$ tételének bizonyításához. Összhangban a kiinduló feltételekkel, E képes lehallgatni a K-M-P1 protokoll 5. és 8. lépéseit. A protokoll lépéseiből és a kiinduló feltevésekből ($F8.$) a $G2.$ tétel [90]-beli bizonyításához hasonló kezdőállításból indulunk ki:
Legyen $t > t_8$ és $n = e(\{m, r_A, A\}, k_{t_8})$, $m_1 = \{A, B, t_8, n_B, k_{t_8}\}$ és $m_2 = e(\{"enc", t_8, k_{t_8}\}, k_T^{-1})$.

$$K_{E,t_6} R\Big(B, t_6, \big\{e(\{n, m_1\}, k_A^{-1}), m_2\big\}\Big) \; , \tag{1}$$

$$K_{E,t} R\Big(B, t, e(\{"dec", t_8, k_{t_8}^{-1}\}, k_T^{-1})\Big) \; . \tag{2}$$

F8. alkalmazásával

$$L_{E,t_6}\{e(\{n, m_1\}, k_A^{-1}), m_2\} \; , \tag{3}$$

$$L_{E,t} e(\{"dec", t_8, k_{t_8}^{-1}\}, k_T^{-1}) \; . \tag{4}$$

($M3$) alapján (3)-ból

$$L_{E,t_6} e(\{n, m_1\}, k_A^{-1}) \ . \tag{5}$$

F9. alapján E ismeri a nyilvános kulcsokat, így $A2(a)$ felhasználásával

$$L_{E,t_0} k_A \ , \tag{6}$$
$$L_{E,t_0} k_T \ . \tag{7}$$

$A3(a)$ alapján

$$L_{E,t_6} k_A \ , \tag{8}$$
$$L_{E,t} k_T \ . \tag{9}$$

(5) és (8)-ból, valamint (4) és (9)-ből ($M2$) alapján

$$L_{E,t_6} \{n, m_1\} \ , \tag{10}$$
$$L_{E,t} \{"dec", t_8, k_{t_8}^{-1}\} \ . \tag{11}$$

($M3$) szerint (10)

$$L_{E,t_6} n \ , \tag{12}$$
$$L_{E,t} k_{t_8}^{-1} \ . \tag{13}$$

$A3(a)$ alapján (12)

$$L_{E,t} n \ . \tag{14}$$

(13), (14) a $TA2(b)$ axiómára alkalmazva adja az állítást:

$$L_{E,t} d(n, k_{t_8}^{-1}) \ , ahol \ n = e(\{m, r_A, A\}, k_{t_8}) \ . \ \square$$

T a protokoll szabályai szerint abszolút megbízható partner. Egy időre tegyük félre ezt a feltételt a vizsgálataink bővítése céljából. Tegyük fel, hogy T ismeri és fel tudja használni az üzeneteket és a nyilvános kulcsokat.

F10. $\forall t \ \forall i \in ENT \ \ K_{T,t}\, R(i,t,x) \ \wedge \ L_{T,t}x$

F11. $K_{T,t_0} L_{T,t_0} k_A$

Állítjuk, hogy T képes visszafejteni az idő-bizalmas üzeneteket t_6 időpontban, B előtt.

Tétel 4.3.2 (G5.) $L_{T,t_6}(d(n, k_{t_8}^{-1}))$ *, ahol* $n = e(\{m, r_A, A\}, k_{t_8})$.

G5.1. Bizonyítás A tétel bizonyításához kihasználhatjuk a $G4$. tételhez való hasonlóságot. Legyen $n = e(\{m, r_A, A\}, k_{t_8})$, $m_1 = \{A, B, t_8, n_B, k_{t_8}\}$ és $m_2 = e(\{\text{"}enc\text{"}, t_8, k_{t_8}\}, k_T^{-1})$. A kulcsok generálása a t_g időpontban (F3. alapján $t_g < t_8$) történik T által, így

$$L_{T,t_g} k_{t_8} \ , \ L_{T,t_g} k_{t_8}^{-1} \ . \tag{1}$$

Azt is tudjuk, hogy T elküldi t_2 időpontban k_{t8} kulcsot A-nak, így teljesülnie kell annak, hogy $t_g \leq t_2$

$$L_{T,t_2} k_{t_8} \ , \ L_{T,t_2} k_{t_8}^{-1} \ . \tag{2}$$

$A3(a)$ alapján

$$L_{T,t_6} k_{t_8}^{-1} \ . \tag{3}$$

Az 5. protokoll-lépés elfogásának eredménye F10. szerint

$$K_{T,t_6} R\Big(B, t_6, \big\{e(\{n, m_1\}, k_A^{-1}), m_2\big\}\Big) \ , \tag{4}$$

és

$$L_{T,t_6}\big\{e(\{n, m_1\}, k_A^{-1}), m_2\big\} \ . \tag{5}$$

$(M3)$ alapján

$$L_{T,t_6} e(\{n, m_1\}, k_A^{-1}) \ . \tag{6}$$

Mivel T ismeri a nyilvános kulcsokat (F11.) és használni is tudja azokat (M2):

$$L_{T,t_6}\{n, m_1\} \ . \tag{7}$$

(M3) alapján

$$L_{T,t_6} n \ . \tag{8}$$

(8) és (3) a $TA2(a)$ axióma alapján

$$L_{T,t_6} d(n, k_{t_8}^{-1}) \ . \ \square$$

Ezek után feltehetjük a kérdést: tudjuk úgy bővíteni a protokollt, hogy a lehallgató ne szerezhessen információt A-tól?

Amennyiben megvizsgáljuk a protokollt, arra juthatunk, hogy két védekezési pontot határozhatunk meg:
I. vagy a kulcsot,
II. vagy az üzenetet védjük.

I. Módosítsuk az eredeti K-M-P1 protokollt az első esetben a következő módon (**K-M-P2 protokoll**): védjük a visszafejtő kulcsot (8. lépés), a következő kódrészlet helyett:

$$T \to B : \{"dec", t_8, k_{t_8}^{-1}\} k_T^{-1}$$

használjuk a

$$T \to B : \{"dec", t_8, \{k_{t_8}^{-1}\} k_B\} k_T^{-1} \ .$$

protokoll lépést. Ezzel a visszafejtő $k_{t_8}^{-1}$ kulcsot védjük a B fél nyilvános kulcsával. Formálisan, a CNS-logika jelöléseivel:

8'. $S(T, t_8, e(\{"dec", t_8, e(k_{t_8}^{-1}, k_B)\}, k_T^{-1}));$
$R(B, t_9, e(\{"dec", t_8, e(k_{t_8}^{-1}, k_B)\}, k_T^{-1}))$

II. A második esetben az eredeti K-M-P1 protokollt módosítsuk a következő módon (**K-M-P3 protokoll**): védjük magát az üzenetet, az 5. lépésben a következő kódrészlet helyett

$$A \to B : \{\{m, r_A, A\}k_{t_8}, A, B, t_8, n_B, k_{t_8}\}k_A^{-1}, \{"enc", t_8, k_{t_8}\}k_T^{-1}$$

használjuk az 5'.

$$A \to B : \Big\{\{\{m, r_A, A\}k_{t_8}\}k_B, A, B, t_8, n_B, k_{t_8}\Big\}k_A^{-1}, \{"enc", t_8, k_{t_8}\}k_T^{-1}$$

vagy az 5".

$$A \to B : \Big\{\{\{m\}k_B, r_A, A\}k_{t_8}, A, B, t_8, n_B, k_{t_8}\Big\}k_A^{-1}, \{"enc", t_8, k_{t_8}\}k_T^{-1}$$

protokoll lépést. Ez hasonló ahhoz, mint amikor egy dobozt két lakattal zárunk le: használjuk az idő-lakatot (k_{t_8}) és a B fél (k_B) nyilvános kulcsát.[3] Formálisan, a CNS-logika jelöléseivel:
Legyen $m_1 = \{A, B, t_8, n_B, k_{t_8}\}$ és $m_2 = e(\{"enc", t_8, k_{t_8}\}, k_T^{-1})$

5'. $S\Big(A, t_5, \Big\{e\Big(\big\{e\big(e(\{m, r_A, A\}, k_{t_8}), k_B\big), m_1\big\}, k_A^{-1}\Big), m_2\Big\}\Big)$
$R\Big(B, t_6, \Big\{e\Big(\big\{e\big(e(\{m, r_A, A\}, k_{t_8}), k_B\big), m_1\big\}, k_A^{-1}\Big), m_2\Big\}\Big)$
vagy

5". $S\Big(A, t_5, \Big\{e\Big(\big\{\{e\big(\{e(m, k_B), r_A, A\}, k_{t_8}\big)\}, m_1\big\}, k_A^{-1}\Big), m_2\Big\}\Big);$
$R\Big(B, t_6, \Big\{e\Big(\big\{\{e\big(\{e(m, k_B), r_A, A\}, k_{t_8}\big)\}, m_1\big\}, k_A^{-1}\Big), m_2\Big\}\Big)$

Ennek a módosításnak hatása van a 6. protokoll lépésre is, mivel B megismétli A üzenetének egy részét. Ezek szerint a

$$B \to A : \Big\{\{\{m, r_A, A\}k_{t_8}, A, B, t_8, n_B, k_{t_8}\}k_A^{-1}\Big\}k_B^{-1}$$

üzenetrész helyett használjuk a 6'.

$$B \to A : \Big\{\Big\{\{\{m, r_A, A\}k_{t_8}\}k_B, A, B, t_8, n_B, k_{t_8}\Big\}k_A^{-1}\Big\}k_B^{-1}$$

[3]Hasonló eredményre jutunk ha különböző kulcs-sorrendeket használunk.

vagy 6".

$$B \to A : \Big\{\Big\{\{\{m\}k_B, r_A, A\}k_{t_8}, A, B, t_8, n_B, k_{t_8}\Big\}k_A^{-1}\Big\}k_B^{-1}$$

protokollrészt. Formálisan, a CNS-logika jelöléseivel: $m_1 = A, B, t_8, n_B, k_{t_8}$

$$\text{6'.} \quad S\Big(B, t_6, e\Big(e\Big(\Big\{e\Big(e\big(\{m, r_A, A\}, k_{t_8}\big), k_B\Big), m_1\Big\}, k_A^{-1}\Big), k_B^{-1}\Big)\Big)$$
$$R\Big(A, t_7, e\Big(e\Big(\Big\{e\Big(e\big(\{m, r_A, A\}, k_{t_8}\big), k_B\Big), m_1\Big\}, k_A^{-1}\Big), k_B^{-1}\Big)\Big)$$

vagy

$$\text{6".} \quad S\Big(B, t_6, e\Big(e\Big(\Big\{e\big(\{e(m, k_B), r_A, A\}, k_{t_8}\big), m_1\Big\}, k_A^{-1}\Big), k_B^{-1}\Big)\Big)$$
$$R\Big(A, t_6, e\Big(e\Big(\Big\{e\big(\{e(m, k_B), r_A, A\}, k_{t_8}\big), m_1\Big\}, k_A^{-1}\Big), k_B^{-1}\Big)\Big)$$

Ezek után vizsgáljuk meg a K-M-P2 és K-M-P3 protokollokat a korábban kitűzött célok elérésével kapcsolatban. Az eddigi $F1.$ - $F11.$ kiinduló feltevések érvényben vannak.

Tétel 4.3.3 (G6. - A K-M-P2 esete) *Az E támadó (lehallgató) nem képes visszafejteni a titkosított üzenetet a K-M-P2 protokoll használatakor - még akkor sem, ha E ismeri a protokoll teljes üzenetforgalmát.*

$$\forall t > t_8 \;\; \neg L_{E,t} d(n, k_{t_8}^{-1}) \text{ , ahol } n = e(\{m, r_A, A\}, k_{t_8}) \text{ .}$$

Bizonyítás G6. Legyen $t > t_8$, $n = e(\{m, r_A, A\}, k_{t_8})$,
$m_1 = A, B, t_8, n_B, k_{t_8}$ és $m_2 = e(\{\text{"}enc\text{"}, t_8, k_{t_8}\}, k_T^{-1})$.
Az 5. és $8.'$ lehallgatása következtében $F8.$ alapján:

$$K_{E,t_6} R\big(B, t_6, \big\{e(\{n, m_1\}, k_A^{-1}), m_2\big\}\big) \text{ ,} \tag{1}$$
$$K_{E,t}\big(B, t_9, e(\{\text{"}dec\text{"}, t_8, e(k_{t_8}^{-1}, k_B)\}, k_T^{-1})\big). \tag{2}$$

Tovább szintén $F8.$ alapján:

$$L_{E,t_6}\left\{e(\{n, m_1\}, k_A^{-1}), m_2\right\} , \tag{3}$$

$$L_{E,t}e(\{\text{"}dec\text{"}, t_8, e(k_{t_8}^{-1}, k_B)\}, k_T^{-1}) . \tag{4}$$

(3) és $(M3)$ alapján

$$L_{E,t_6}e(\{n, m_1\}, k_A^{-1}) . \tag{5}$$

Mivel E ismeri a nyilvános kulcsokat $F9.$, $A2(a)$ és $A3(a)$ alapján $(L_{E,t_6}k_A\ ,\ L_{E,t}k_T)$; $(M2)$ alapján (5) és (4)-ből

$$L_{E,t_6}\{n, m_1\} , \tag{6}$$

$$L_{E,t}\{\text{"}dec\text{"}, t_8, e(k_{t_8}^{-1}, k_B)\} . \tag{7}$$

(6) és $(M3)$ alapján

$$L_{E,t_6}n . \tag{8}$$

$A3(a)$ szerint

$$L_{E,t}n . \tag{9}$$

(7) és $(M3)$ alapján

$$L_{E,t}e(k_{t_8}^{-1}, k_B) . \tag{10}$$

A továbblépéshez az $(e(k_{t_8}^{-1}, k_B))$ titkosított üzenet megfejtésére lenne szükség, ami B k_B^{-1} titkos kulcsával lehetséges, amit E nem ismer. Ebből a $T2A(a)$ axióma felhasználásával adódik a tétel állítása. □

A K-M-P2 protokoll T egyed esetén nem bizonyítható a $G6.$ tételhez hasonló állítás, mivel T generálja a k_{t_8}, $k_{t_8}^{-1}$ kulcspárt, így ismeri és fel is tudja használni azokat.

Térjünk rá a K-M-P3 protokoll vizsgálatára. Ebben az esetben, már T sem képes megfejteni az idő-bizalmas üzeneteket.

Tétel 4.3.4 (G7. - A K-M-P3 protokoll esete - T egyed) *Az abszolút megbízható T fél nem képes visszafejteni a titkosított üzenetet a K-M-P3 protokoll használatakor - még akkor sem, ha T a partnerek minden üzenetváltását ismeri.*

$$\forall t > t_8 \ \ \neg L_{T,t} m \ .$$

m az 5'. (vagy 5".) lépésben átküldött időbizalmas üzenet.

Bizonyítás G7. Az 5". esetet bizonyítjuk - az 5.' eset hasonlóan látható be. Legyen $t > t_8$, $n = e(\{m, r_A, A\}, k_{t_8})$,
$m_1 = \{A, B, t_8, n_B, k_{t_8}\}$ és $m_2 = e(\{"enc", t_8, k_{t_8}\}, k_T^{-1})$.
F10. szerint T lehallgathatja az üzeneteket, így ismeri az 5". lépésben átküldött üzenetet.

$$K_{T,t_6} R\Big(B, t_6, \Big\{e\Big(\big\{\big\{e\big(\{e(m, k_B), r_A, A\}, k_{t_8}\big)\big\}, m_1\big\}, k_A^{-1}\Big), m_2\Big\}\Big).$$

Az előző bizonyításokhoz hasonlóan F10., F11., $(M2)$, $(M3)$, $A3(a)$ alapján

$$L_{T,t} e(m, k_B).$$

Ennek az üzenetnek a visszafejtéséhez a B egyed k_B^{-1} titkos kulcsára lenne szükség, amivel T nem rendelkezik. $TA2(a)$ axióma alkalmazásával a tételben megfogalmazottakhoz jutunk. □

Az E egyedre hasonló tételt bizonyíthatunk.

Tétel 4.3.5 (G8. - A K-M-P3 protokoll esete - E egyed) *Az üzeneteket lehallgató E fél nem képes visszafejteni a titkosított üzenetet a K-M-P3 protokoll használatakor - még akkor sem, ha E a partnerek minden üzenetváltását ismeri.*

$$\forall t > t_8 \ \ \neg L_{E,t} m \ .$$

m az 5'. (vagy 5".) lépésben átküldött időbizalmas üzenet.

Bizonyítás G8. Az 5". esetet bizonyítjuk - az 5.' eset hasonlóan látható be. Legyen $t > t_8$, $n = e(\{m, r_A, A\}, k_{t_8})$,

$m_1 = \{A, B, t_8, n_B, k_{t_8}\}$ és $m_2 = e(\{"enc", t_8, k_{t_8}\}, k_T^{-1})$.
F8. szerint E lehallgathatja az üzeneteket, így ismeri az 5". lépésben átküldött üzenetet.

$$K_{E,t_6} R\Big(B, t_6, \Big\{e\Big(\{\{e\big(\{e(m, k_B), r_A, A\}, k_{t_8}\big)\}, m_1\}, k_A^{-1}\Big), m_2\Big\}\Big)$$

Az előző bizonyításokhoz hasonlóan F8., F9., $(M2)$, $(M3)$, $A3(a)$ alapján

$$L_{E,t} e(m, k_B)$$

Ennek a visszafejtéséhez a B egyed k_B^{-1} titkos kulcsára lenne szükség, amivel E nem rendelkezik. $TA2(a)$ axióma alkalmazásával a tételben megfogalmazottakhoz utunk. □

Összefoglalva elmondhatjuk, hogy a passzív támadók (E és T lehallgatói szerepben) a K-M-P1 protokollban közvetített idő-bizalmas üzeneteket képesek megismerni. E a protokoll lefutása után, T pedig már előtte is képes erre. A javított protokollok esetén K-M-P2 az E támadásának, K-M-P3 pedig E és T támadásának is ellenáll. Ekkor T szerepe a kulcsok generálására és a megfelelő időben történő közlésére korlátozódik. A és B biztos lehet abban, hogy sem E, a lehallgató, sem T, az abszolút megbízhatónak tekintett szerver sem ismeri a titkosított üzenet tartalmát. Ez a tény T számára is előnyös, hiszen az eljárás védi T-t a lehallgatás vádja alól.

4.3.2. Aktív támadás

Az előző részben a K-M-P1 protokoll lehallgatásakor, a passzív támadáskor adódó lehetőségeket vizsgáltuk. Ekkor a támadó nem módosítja a protokollt, csak a kommunikációs vonalakon történő adatátvitelt figyeli és rögzíti. Az aktív támadás viszont beleavatkozik a protokoll működésébe. A leggyakrabban a beékelődő támadásról (*MiM - Man-in-the-Middle attack*) lehet hallani, mint végrehajtott aktív támadásról (**2.1.2.** és **2.2.1. fejezet**).

Az K-M-P1 protokoll elemzése során hangsúlyoztuk, egy időre tekintsünk el attól, hogy T abszolút megbízható fél. Ennek során jutottunk el a

K-M-P2 és K-M-P3 protokollokhoz. Ez a feltételezés azt a tényt használta ki, hogy a BAN-logika nem képes legális, de a protokoll szempontjából támadólag viselkedő résztvevők támadásait detektálni. [45] Hasonló mondható el a beékelődő támadásról, és általában az aktív támadásokról. Ennek oka az, hogy a BAN-logika és a hozzá hasonló rendszerek általában nem képesek egy aktív támadó tevékenységét modellezni. Az aktív támadások vizsgálatára alkalmasabbnak tűnnek az absztrakt állapotgépeken alapuló rendszerek, amelyek az automatizált ellenőrző eszközök fő modelljei. A következő fejezet ezzel a témakörrel foglalkozik.

4.4. A Kudo-Mathuria protokollok vizsgálata az AVISPA rendszer segítségével

A protokollok vizsgálati eszközeinek bemutatásakor már esett szó az AVISPA rendszerről (**3.2.1. fejezet**). Most az eddig vizsgált Kudo-Mathuria K-M-P1 protokoll és módosított változatai (K-M-P2 és K-M-P3) kerülnek további elemzésre az AVISPA rendszer alkalmazásával. Először egy rövid összefoglalót olvashatunk az elemző eszközről, majd a protokoll-ellenőrzés eredményeit mutatjuk be. A protokollokat leíró kódok és az ellenőrzéskor adódó eredmények a dolgozat mellékletében kaptak helyet (**7.2. fejezet**). További részletek az AVISPA rendszerrel, a HLPSL leíró nyelvvel, IF közbenső kódformáról és a SPAN Animator-ról a [12][13][67][14][73] forrásokban találhatók.

4.4.1. Az AVISPA rendszer

Az AVISPA-rendszer (*Automated Validation of Internet Security Protocols and Applications*) egy fél-automata alkalmazás (a vizsgálat során emberi beavatkozást igényel), amelyet biztonsági protokollok fejlesztésére és elemzésére hoztak létre. Az AVISPA olyan szerep-alapú (*role-base*), formális nyelv bázisú eszköz - protokollok pontos leírására és vizsgálatára -, amely négy különböző ellenőrzési lehetőséget foglal magába (OFMC, CL-AtSe,

SATMC, TA4SP). Ezt egészíti ki a SPAN (*Security Protocol ANimator*) nevű alkalmazás - ami nem közvetlenül része az eredeti fejlesztésnek, de ma már megfelelően integrálódott a rendszerbe. A SPAN a protokollok vizsgálatakor grafikus felületet biztosít a fejlesztők számára a protokoll kód összeállítására, és a futási modellek vizuális megjelenítésére.[4]

Az eszköz használatának első lépése az analizálandó protokoll leírása, megjelenítése HLPSL (*High Level Protocol Specification Language*) nyelven. Az elemzés következő lépésében a HLPSL protokoll-leírást a rendszer átalakítja egy alacsonyabb szintű nyelvre (*IF Intermediate Format*). Ennek a transzformációnak az oka az, hogy az IF protokoll-forma már a négy vizsgáló alrendszer inputjaként használható (közös platformot jelent az eltérő rendszerek esetében). Az alrendszerekben külön-külön lehet elvégezni az elemzéseket, ezek kölcsönösen kiegészítik egymás képességeit. Az alrendszerek speciális formában rögzített outputja adja a protokollvizsgálat eredményét, ami alrendszerenként eltérő lehet. Minden rész működése feltételezi azt, hogy a támadó nem képes feloldani a titkosítást a teljes kulcs birtoklása nélkül. A csatornák modellezése a Dolev-Yao támadási modellt követi, a támadó alapvetően teljes hozzáféréssel rendelkezik a csatornák felett.

A HLPSL *szerep-alapúsága* azt jelenti, hogy minden résztvevő tevékenységét szétbontva, különálló modulban írjuk le. Ezeket a modulokat alap-szerepeknek (*basic role*) nevezzük. A *basic role*-ok leírják, hogy a résztvevők milyen kezdeti paraméterekkel és kezdőállapottal indulnak a protokollban, milyen állapotváltozások történnek a protokoll-lépések során (*transition*). A következő kódrészlet ezeket szemlélteti: [5]

```
role alice(A, B: agent, RCV, SND: channel(dy))
  played_by A
  def=
```

[4] Az AVISPA rendszer fejlesztői 2008-ban hivatalosan is csatlakoztak az AVANTSSAR projekthez, amely még nagyobb erőket mozgósít a formális vizsgálatok automatizálása céljából. A projekt résztvevői: IBM, Siemens, SAP, európai egyetemek. [9]

[5] A mintapélda egy üzenet eküldése: A → B: Uzenet

```
    local Allapot: nat, Uzenet: text
  init Allapot:=0
  transition
  1 . Allapot = 0 /\ RCV(start) =|> Allapot':=2
    /\ Uzenet':=new() /\ SND(Uzenet')
end role


role bob(A, B: agent, RCV, SND: channel(dy))
  played_by B
  def=
    local Allapot: nat, Uzenet: text
  init Allapot:=1
  transition
  1. Allapot = 1 /\ RCV(Uzenet') =|> Allapot':=2
    /\ secret(Uzenet',u,A,B)
end role
```

A HLPSL a *basic role*-okon kívül, azok után, *composition role*-okat definiál. Ezekben határozzuk meg a rendszer számára, hogy a különböző *basic role*-ok hogyan kapcsolódnak egymáshoz. A *basic role*-ok kapcsolódása a *role*-ok párhuzamos végrehajtását jelenti. A *composition role*-ok a protokoll leírásának *session* részébe kerülnek. A *composition role*-ok nem definiálnak *transition*-okat, mint a *basic role*-ok, helyettük létrehozzák a *basic role*-ok példányait és definiálják a *basic role*-ok által használt csatornákat.

```
role session(A, B: agent)
  def=
    local SNDA, RCVA, SNDB, RCVB : channel(dy)
  composition
  alice(A, B, SNDA, RCVA) /\ bob(A, B, SNDB, RCVB)
end role
```

Az utolsó lépésben a HLPSL egy felső szintű *role*-t definiál, értelmez. Ezt a *role*-t *envinronment*-nek nevezi. Ez a globális változók bevezetését

tartalmazza, valamint a különböző *session*-ok összekapcsolását. Ebben a felső szintű *role*-ban kerül rögzítésre, hogy a protokoll elemzése során a támadó fél milyen információkkal rendelkezik (*intruder_knowledge*), és hogy milyen módon éri el a protokollt, annak egyes lépéseit. Például ha egy támadó egy legitim felhasználó szerepét játszhatja a protokoll futása során, azt a felső szinten kell definiálni.

Az elemzés során pontosan rögzítenünk kell az elérni kívánt célokat, hogy vizsgálhassuk azok teljesülését. A HLPSL formában történő leírás a *goals* szekcióban teszi lehetővé a biztonsági követelmények összefogását, célok leírását. A biztonsági követelmények tényleges rögzítése viszont a *basic role*-ok *transition* szintjén megtörténik. Az ezen a szinten definiált célokat *goal facts*-nak nevezzük. A *goals* részben egyszerűen a *goal facts*-ok kerülnek összekapcsolásra.

```
role envinronment()
  def=
    const a, b: agent, u: protocol_id
  composition
    session(a,b)
end role
goal
  secrecy_of u
end goal
envinronment()
```

4.4.2. A Kudo-Mathuria protokoll vizsgálati eredményei

Munkánk során először a K-M-P1 protokollt vizsgáltuk. A protokoll teljes HLPSL kódja és az elemzés részletei a **7.2.1.** fejezetben (Melléklet) található meg.

Az általunk vizsgált paraméter a visszafejtő kulcs és annak titkossága volt. A szükséges kódrészek:

- 'goal' szekció: *'secrecy_of itktreq'* - 076, 078 sorok
- 'role server' szekció: *'secret(inv(Tktreq), itktreq, {S,B}'* - 054 sor

PC-s környezetben a rendszer az OMFC és ATSE modelleket tette elérhetővé. Mindkét modell hibát jelzett (7.2.2.fejezet - 004, 006 sorok, és 7.2.3.fejezet - 003, 005 sorok): A két modell a protokoll támadását is leírja ('ATTACK TRACE' rész: 7.2.2.fejezet - 019 - 036 sorok; 7.2.3.fejezet - 018 - 028 sorok), amelyek jellemzően aktív támadást jelentenek. A külső fél aktívan kezdeményezi a protokoll futását hamis üzenetek generálásával (a támadás leírásában i a támadó felet, és az általa kezdeményezett üzenetküldést jelzi). Az aktív támadás sokféle módon lebonyolítható, amit ki is lehet próbálni a SPAN rendszeren belül (*Attack simulation*). A fejezetek ábrái egy-egy támadás kezdőlépéseit mutatják (7.2.2.fejezet és 7.2.3. ábra).

A K-M-P2 és K-M-P3 protokollok esetén már nem mutatott hasonló hibát a rendszer (7.2.4.fejezet: 004 sor, 7.2.5.fejezet: 002 sor, 7.2.6.fejezet: 004 sor, 7.2.7.fejezet: 002 sor).

5. fejezet

A többcsatornás kriptográfiai protokollok vizsgálata a bővített CSN logika eszközeivel

Ebben a fejezetben folytatjuk a CSN logika fejlesztését és alkalmazási körének további bővítését. A fő cél annak elérése, hogy a CSN-logika alkalmas legyen többcsatornás kriptográfiai protokollok vizsgálatára. Először a kriptográfiai inicializálás témakörét vizsgáljuk, majd a többcsatornás protokollok hasonló feladatait tekintjük át személyi hálózatok esetén (*PAN - Personal Area Network*). A különbség lényeges, hiszen a személyi hálózatokban általában szerverek támogatása nélkül kell megvalósítani a kriptográfiai feladatokat. Ezután rátérünk a CSN-logika bővítésére, amivel az eszközrendszer alkalmassá válik az egyre szélesebb körben alkalmazott többcsatornás rendszerek logikai elemzésére.

5.1. Kriptográfiai inicializálás személyi hálózatokban

2005-ben több mint 9 milliárd mikroprocesszort állítottak elő világszerte. Ezen alkatrészeknek csak kevesebb mint 2%-át építették be számítógépekbe (PC, Mac, Unix, stb.). A fennmaradó közel 8,8 milliárd processzor-egység beágyazott rendszerekbe került. [15] Ma minden modernnek nevezett elektronikus eszköz rendelkezik egy vagy több beépített processzorral és a megfelelő működtető háttérrel. A zenés képeslapoktól kezdve a közlekedési lámpákon át a nukleáris erőművek vezérlő rendszeréig mind tartalmaznak processzorokat. Ezek az egységek az egyszerűnek tekintett 4 bites mikrokontrollerektől kezdve a felsőbb szintű 128 bites processzorokig terjednek. Az alkalmazott megoldások egy része a felhasználók számára közvetlenül elérhető, kezelhető és kezelendő, más része viszont *„rejtett”*, a háttérben működik. Környezetünkben biztosan találunk több olyan berendezést, elektronikus eszközt, amely szintén tartalmaz mikroprocesszort: mobiltelefon, televízió, rádió, mosógép, stb. Ezen eszközök egyre nagyobb hányada kommunikációra, adatcserére is képes társaival, más eszközökkel. A kapcsolat vezeték segítségével, vagy vezeték nélkül - ekkor általában elektromágneses sugárzás felhasználásával -, valósul meg. Ennek okán a hagyományos, területi kiterjedés szerinti osztályozás szokásos kategóriái (*LAN, MAN, WAN*) mellett megjelent a PAN (*Personal Area Network - Személyi hálózat*) és a BAN vagy WBAN (*Wireless Body Area Network*) fogalma. A kapcsolódás lehet emberi beavatkozást igénylő (*kézi, manual*), vagy automatikus. Az automatikus beállítású és önmenedzselő eszközökből épített PAN rendszerek ismét külön kategóriát alkotnak - ASPAN (*Auto-configuration and Self-management of Personal Area Networks*) néven.

A PAN eszközök és hálózataik legalább egy alapvető különbséget mutatnak nagyobb méretű társaikhoz képest. A kisebb, személyes környezetben alkalmazott eszközök kommunikációját legtöbb esetben nem segíti szerver háttér. Ez az alapvető különbség számos következménnyel jár. Az egyik ezek közül a biztonság kérdése. PAN és ASPAN környezetben egyrészt hasonló biztonsági és titkosítási kérdések merülnek fel, mint a nagyobb

hálózatoknál: felhasználó azonosítás, kulcskezelés, rejtjelezés, digitális aláírás, üzenethitelesítés, stb. Másrészt újabb kihívásokkal kell szembenézni a vezeték nélküli adatátvitel elterjedésével: fokozott lehallgathatóság, eszközök kommunikációjának felépítése előzetes beállítások nélkül, több eszköz párhuzamos kommunikációja, stb. Fontos kérdés tehát az, hogy hogyan oldhatók meg ezek a feladatok szerver háttér, külső támogatás nélkül.

A továbbiakban egy speciális területtel, az *imprinting* kérdésével foglalkozunk. Ennek a feladatkörnek a lényege az, hogy a biztonságos és megfelelően védett kommunikációs folyamat kialakításához megfelelő paraméterek beállítására és átadására van szükség az eszközökön, az eszközök között. Ez lényegében a kriptográfiai inicializálást jelenti. Ezeken a beállításokon múlnak a későbbi működés alapvető vonásai, így elmondható, hogy az inicializálási folyamat itt is igen érzékeny részét képezi a kriptográfiai rendszereknek. [68][127] Amennyiben lehetséges a kriptográfiai inicializálás lehallgatása, vagy más irányú támadása, az azt követő - az inicializálásra épülő, védeni kívánt - kommunikációs folyamat sebezhetővé válik.

5.2. Többcsatornás protokollok vizsgálata

Sok esetben védett csatornákat alkalmaznak az alapparaméterek beállítása során. PAN környezetben a biztonsági csatorna alapulhat

- rögzített kapcsolódáson (kábel, USB interfész, vonalkód, stb.),
- emberi beavatkozáson (jelszavak leolvasása, beírása, átírása, stb.),
- egyéb *kis hatósugarú* technikán (másodlagos vezeték nélküli csatorna, stb.). [68]

Az emberi beavatkozáson alapuló biztonságos csatorna feltételezi azt, hogy az eszközök inicializálási folyamatában részt vesz egy aktívan közreműködő felhasználó is. Ez egyrészt megteremti a biztonsági csatorna könnyű kialakíthatóságának lehetőségét, másrészt módot ad a többfaktoros kriptográfia eredményeinek alkalmazására is. Ugyanakkor az emberi közreműködés bizonyos korlátokat is felvet. A protokollok tervezése során például figyelembe kell venni az emberek által átlagosan megjegyezhető

karaktersorozatok hosszát, a gépelési sebességet, stb.

Az emberi beavatkozást alkalmazó kapcsolatépítő protokollok (*Human Assisted Pairing Protocols*) két alosztályra bonthatók.

Numerikus összehasonlításon alapuló protokollok - Numeric comparsion based protocols

A numerikus összehasonlításon alapuló protokollok esetén az emberi közreműködőnek az egyes berendezéseken, eszközökön lévő (ott megjelenített, arról leolvasott) adatokat kell összehasonlítania. Néhány példa erre a típusra:

- MANual Authentication Protocol - MANA I. [66][68]
- MANA II Protocol. [66][68]
- MANA IV Protocol. [93]
- Three-round Mutual Authentication Protocol. [93]
- Bidirectional Authentication Protocol [46]

Azonosító kulcson alapuló protokollok - Passkey-based protocols

Az azonosító kulcson alapuló protokollok az eszközök közötti megosztott titkon alapulnak. Két példa erre a típusra:

- EKE Protocol [22][23]
- MANA III Protocol [66]

Mint látható, a MANA protokollok egy fontos családját alkotják az emberi beavatkozást igénylő protokolloknak. Munkánk során ezekkel a protokollokkal foglalkozunk, ezeknek a protokolloknak a vizsgálatát tűzzük ki célul. A kiindulópontot F.-L. Wong és F. Stajano 2005-ben megjelent közleménye [157] jelenti, amelyben a szerzők a MANA III protokollt tárgyalják. A közlemény végén, a kitűzött további kutatási feladatok elemzésénél hívják fel a figyelmet a szerzők arra, hogy szükséges a többcsatornás protokollok formális logikai eszközrendszerének kialakítása.

' ... Finally, the last and perhaps the most important tool we need is a logic for multi-channel protocols in the spirit of BAN.' [157]

Kutatási célunk ennek a hiányzó elemnek a megalkotása. Nyilvánvalóan több megoldása is lehet a kitűzött feladatnak.[1] Munkánk során a már bemutatott CSN-logikai rendszert bővítjük úgy, hogy alkalmas legyen a többcsatornás protokollok vizsgálatára.

5.3. A CSN-logika bővítése többcsatornás kriptográfiai protokollok tanulmányozására

Az előzőekben bemutatott CSN-logikai rendszert széles körben alkalmazhatjuk protokollok vizsgálatára (lásd 4. fejezet), de az alaprendszer nem alkalmas a többcsatornás protokollok tanulmányozására. A logika ilyen irányú szintaktikai bővítését és az axiómarendszer kiegészítését mutatjuk itt be részletesen.

5.3.1. A szintaktikai bővítés

A módosított CSN'-rendszer nyelve

$$L^{(CSN')} = \langle Sort', LC, Var', Con', Term', Form \rangle$$

rendezett hatos, ahol vessző (') jelzi az L^{CSN} nyelvtől való eltérést. A bővítés fő iránya a C csatorna-típus bevezetése, a kapcsolódó módosítások kialakítása. Az eredeti CSN-rendszer leírásához igazodva, a következőkben a változtatásokat és az új elemeket soroljuk fel.

Sort'

A típusok (fajták) halmaza:

$$Sort' = \{U, E, K, T, C\}$$

[1] A kommunikációs csatornák kriptográfiai tulajdonságainak kérdése már régen felmerült. Például a titkos kulcsú kommunikáció hallgatólagosan mindig feltételezi a kulcsok előzetes cseréjét, ami általában egy kiegészítő biztonsági csatorna. Többen is foglalkoztak hasonló csatornatulajdonságokkal: [43][70][157][158]. Az AVISPA rendszerben tervezik többféle csatorna alkalmazhatóságának kidolgozását. [11]

C csatorna-típus.

LC
A nyelv logikai konstansainak halmaza - változatlan.

Var'
A nyelv változóinak halmaza. Var_δ a δ típusú változók halmazát jelöli:

$$Var' = Var_U \cup Var_E \cup Var_K \cup Var_T \cup Var_C$$

Con'
A nyelv nemlogikai konstansainak halmaza. Con_δ a δ típusú nemlogikai konstansok halmazát jelöli, egyes típusok esetén üres is lehet a halmaz:

$$Con' = Con_U \cup Con_E \cup Con_K \cup Con_T \cup Con_C$$

Az $F(0)_\delta$ a névkonstansok, az $F(n)_\delta$ n argumentumú függvényjelek, a $P(0)$ állításkonstansok, a $P(n)$ n argumentumú predikátumkonstansok úgyanúgy értelmezettek.

Term'
A nyelv termjeinek halmaza, típusonként induktív definícióval megadva. $Term_\delta$ a δ típusú Termek halmazát jelöli, egyes típusok esetén üres is lehet a halmaz:

$$Term' = Term_U \cup Term_E \cup Term_K \cup Term_T \cup Term_C$$

Az induktív definíció ugyanaz, mint a CSN-rendszerben.

Form
A nyelv formuláinak halmaza - változatlan.

Az egyes konkrét típusok esetén a következő változásokat eszközöljük:

U - üzenet típus
$F(n)_U$ kiegészítése:

$h(m,k)$ Kulcsolt üzenetkivonat (hash) fügvény. $h(m,k)$ jelöli az m üzenet k kulcs segítségével előállított üzenetkivonat értékét. A függvény kimenete üzenet fajta. $h(m,k) \in F(2)$; $\langle U,K,U\rangle$.

$H(m)$ Üzenetkivonat (hash) függvény - MD sorozat, SHA sorozat, HAVAL, RIPEM, stb. A függvény kimenete üzenet fajta. $h(m) \in F(1)$; $\langle U,U\rangle$.

• $Term_U$ kiegészítése:

(f) Ha $m \in Term_U$, $k \in Term_K$, akkor $h(m,k) \in Term_U$.

(g) Ha $m \in Term_U$, akkor $h(m) \in Term_U$.

Megjegyzések kiegészítés:

2. Az EGYED, KULCS és IDŐ fajtájú változóknál értelmezett függvényjelek ($E2U(\Sigma)$, $U2E(m)$; $K2U(k)$, $U2K(m)$ és $T2U(t)$, $U2T(m)$) mellett a CSATORNA típusú változóknál is értelmezzük az üzenetbe ágyazást lehetővé tevő függvényjeleket: $C2U(ch)$, $U2C(m)$ - típus-konverzió.

E - egyed típus

Megjegyzések:

1. A különböző csatornák esetén értelmezzük a csatornát használni képes egyedek halmazát. Nyilvános csatorna esetén ez az ENT halmaz, jelölésben: $ENT_{ch_i} = ENT$. Védett csatorna esetén pedig vesszővel elválasztva felsoroljuk azokat egyedeket, akik a csatornához hozzáférnek: $ENT_{ch_i} = \{\ldots\}$. Természetesen $ENT_{ch_i} \subseteq ENT$.

Az új típus részletes leírása a következő:

C - csatorna típus

Jellemzés: a protokollokban alkalmazott kommunikációs csatornák leírása; CH jelöli az összes lehetséges csatornák halmazát. Ez a halmaz véges.

• Var_C:

ch, ch_1, ch_2, ... ch_i, ch_j csatorna-változók.

• Con_C:

$F(0)_C$: -

$F(n)_T$:

$C2U(t)$ Az csatorna fajta változó átalakítása üzenet fajta változóvá. Ez a függvényjel lehetővé teszi egy protokoll üzenetrészében csatorna adatok szerepeltetését. A függvényjel input paramétere: csatorna fajta, a függvényjel-output: üzenet fajta.
A függvényjel helyettesítésekor adódó eredményt egyszeres idézőjelek közé írjuk: $C2U(ch_i) =' ch_i'$. $C2U \in F(1)$; $\langle C, U \rangle$.

$U2C(m)$ A megfelelő üzenet változó átalakítása csatorna fajta változóvá. Ez a függvényjel lehetővé teszi egy protokollban az üzenetként átküldött csatorna adatok értelmezését. A függvényjel input paramétere: üzenet fajta, függvényjel-output: csatorna fajta. $U2C('ch_i') = ch_i$. $U2C \in F(1)$; $\langle U, C \rangle$.

$P(0)$: -

$P(n)$: -

• $Term_T$:

(a) Ha $ch \in Term_C$, akkor $C2U(t) \in Term_U$.

(b) Ha $m \in Term_U$, akkor $U2C(m) \in Term_C$.

• $Form$:

Megjegyzések:

1. Szükséges a csatornák tulajdonságainak leírása a rendszerben. Az egyszerűség kedvéért csak kétféle csatornát (védett és nyilvános) különböztettünk meg. Jelölje $CH(ch_i, sec)$ azt, hogy a ch_i csatorna védett. Hasonlóan jelölje $CH(ch_i, pub)$ azt, hogy a ch_i csatorna nyilvános. A nyilvános csatorna alapfelfogása követi a Dolev-Yao-féle támadási modellt. [57] Amennyiben egy csatorna védett, úgy meg kell határozni azoknak az egyedeknek a körét, akik használhatják azt. Erre az egyed-típusnál említett $ENT_{ch_i} = \{\ldots\}$ jelölést alkalmazzuk.

Operátorok:

A csatornák megkülönböztetésére szükséges egy csatornaindex bevezetése az R fogadó és S kibocsátási operátor argumentumában. Az eredeti R operátor $R(\Sigma, t, x)$ formátumú. Ennek jelentése az, hogy a Σ egyed az x üzenetet fogadja a t időpillanatban. Az eredeti S operátor $S(\Sigma, t, x)$ formátumú. Ennek jelentése az, hogy a Σ egyed az x üzenetet küldi el a t időpontban. Ezt a formát a következőképpen bővítettük:

- Legyen a *fogadó* operátor új alakja $R(ch_i, \Sigma, t, x)$. Ennek jelentése: a Σ egyed az x üzenetet fogadja a t időpillanatban a ch_i csatornán.
- Legyen a *küldő* operátor új alakja $S(ch_i, \Sigma, t, x)$. Ennek jelentése: a Σ egyed az x üzenetet küldi a t időpillanatban a ch_i csatornán.

5.3.2. A bővített axiómatikus rendszer

A bővítés során nincs szükség a következtetési szabályok változtatására. Az axiómák közül az $A5$, $A6$, $A8$, $A12$,$A15$ és $TA3$ axiómák változnak. Az új verziókat vesszővel (') jelöljük.

A5'(a) $S(ch_i, \Sigma, t, m)$
$\rightarrow L_{\Sigma,t}m \ \wedge \ \exists \Psi \in ENT_{ch_i} \backslash \{\Sigma\} \ \exists t_i > t \ R(ch_i, \Psi, t_i, m).$

A6'(a) $R(ch_i, \Sigma, t, m)$
$\rightarrow L_{\Sigma,t}m \ \wedge \ \exists \Psi \in ENT_{ch_i} \backslash \{\Sigma\} \ \exists t_i < t \ S(ch_i, \Psi, t_i, m).$

A6'(b) $R(ch_i, \Sigma, t, m_1) \wedge C(m_1, m_2) \wedge O_{\Sigma,t}(m_1, m_2) \rightarrow$
$\exists \Psi \in ENT \ \exists t_i < t \ \exists m_3 \ (S(ch_i, \Psi, t_i, m_3) \wedge C(m_3, m_2) \wedge L_{\Psi,t_i}m_2$
$\wedge O_{\Sigma,t}(m_1, m_3) \wedge O_{\Sigma,t}(m_3, m_2))$

A8'(a) $\neg L_{\Psi,t}k_\Sigma \ \wedge \ \forall t_i < t \ \neg L_{\Psi,t_i}(e(m, k_\Sigma)) \ \wedge \ \neg(\exists n(R(ch_i, \Psi, t_i, n) \wedge$
$C(n, e(m, k_\Sigma)))) \rightarrow \neg L_{\Psi,t}(e(m, k_\Sigma)).$

A8'(b) $\neg L_{\Psi,t}k_\Sigma^{-1} \ \wedge \ \forall t_i < t \ \neg L_{\Psi,t_i}(d(m, k_\Sigma^{-1})) \wedge \ \neg(\exists n(R(ch_i, \Psi, t_i, n) \ \wedge$
$C(n, d(m, k_\Sigma^{-1})))) \rightarrow \neg L_{\Psi,t}(d(m, k_\Sigma^{-1})).$

A12'(a) $(\neg L_{\Gamma,t}ks_{(\Sigma,\Psi)} \wedge \forall t_i \leq t \ \neg L_{\Gamma,t_i}(E(m, ks_{(\Sigma,\Psi)}))\wedge$
$\neg(\exists n(R(ch_i, \Gamma, t_i, n) \ \wedge \ C(n, E(m, ks_{(\Sigma,\Psi)}))))$
$\rightarrow \neg L_{\Gamma,t}(E(m, ks_{(\Sigma,\Psi)}))).$

A12'(b) $(\neg L_{\Gamma,t}ks_{(\Sigma,\Psi)} \wedge \forall t_i \leq t \ \neg L_{\Gamma,t_i}(D(m, ks_{(\Sigma,\Psi)}))\wedge$
$\neg(\exists n(R(ch_i, \Gamma, t_i, n) \ \wedge \ C(n, D(m, ks_{(\Sigma,\Psi)}))))$
$\rightarrow \neg L_{\Gamma,t}(D(m, ks_{(\Sigma,\Psi)}))).$

A15'(a) $[A(\Sigma, t, \Psi) \rightarrow (L_{\Sigma,t}ss_{(\Sigma,\Psi)} \wedge ss_{(\Sigma,\Psi)} \in \{SS_{\{\Sigma,\Psi\}}\} \wedge R(ch_i, \Sigma, t, m)) \wedge$
$C(m, ss_{(\Sigma,\Psi)}) \wedge \forall t_i \leq t \ \neg S(ch_i, \Sigma, t_i, m)] \rightarrow K_{\Sigma,t}(S(ch_i, \Psi, t_i, m))$

A15'(b) $[A(\Sigma, t, \Psi) \rightarrow (L_{\Sigma,t}k_\Psi \wedge L_{\Sigma,t}m \wedge R(ch_i, \Sigma, t, n)$
$\wedge C(n, e(m, k_\Psi^{-1}))] \rightarrow \forall t_i \leq t \ \ K_{\Sigma,t}(S(ch_i, \Psi, t_i, n))$

TA3'(a) $\neg L_{\Sigma,t} k_\tau \wedge \forall t_i \leq t\ \neg L_{\Sigma,t_i}(e(m,k_\tau)) \wedge$
$\neg(\exists n(R(ch_i,\Sigma,t_i,n) \wedge\ C(n,e(m,k_\tau))\ \wedge\ O_{\Sigma,t}(n,e(m,k_\tau))))$
$\rightarrow\ \neg L_{\Sigma,t}(e(m,k_\tau))$

TA3'(b) $\neg L_{\Sigma,t} k_\tau^{-1} \wedge \forall t_i \leq t\ \neg L_{\Sigma,t_i}(d(m,k_\tau^{-1})) \wedge$
$\neg(\exists n(R(ch_i,\Sigma,t_i,n) \wedge C(n,d(m,k_\tau^{-1}))\ \wedge\ O_{\Sigma,t}(n,d(m,k_\tau^{-1}))))$
$\rightarrow \neg L_{\Sigma,t}(d(m,k_\tau^{-1}))$

Mindezeken kívül új axiómákra is szükségünk van az üzenetkivonat-függvények tulajdonságainak leírására.

A16(a) $L_{\Sigma,t} m \wedge L_{\Sigma,t} k \rightarrow L_{\Sigma,t} h(m,k)$.
A Σ egyed képes elkészíteni a t időpontban a kulcsolt üzenetkivonatot (kulcsolt hash függvény), ha rendelkezésre áll az m üzenet és a k kulcs. Az eredmény $h(m,k)$ üzenet típusú.

A17(a) $h(n_A,k_A) = h(n_B,k_B) \leftrightarrow n_A = n_B\ \wedge\ k_A = k_B$.
A kulcsolt hash függvény alaptulajdonságainak rögzítése.

A18(a) $L_{\Sigma,t} m \rightarrow L_{\Sigma,t} H(m)$.
A Σ egyed képes elkészíteni a t időpontban az üzenetkivonatot (hash függvény), ha rendelkezésre áll az m üzenet. Az eredmény $H(m)$ üzenet típusú.

A19(a) $H(n_A) = H(n_B) \leftrightarrow n_A = n_B$.
A hash függvény alaptulajdonságainak rögzítése.

Megjegyzések - kiegészítés:

(M5) A protokollok formális alakban történő rögzítése az eddigiekben leírt formális rendszer alkalmazását jelenti konkrét protokollok esetében. Többcsatornás protokollok esetén rögzítjük azt a kódolási előírást, hogy nyilvános csatorna esetén a küldött és a fogadott üzenetekről feltesszük, hogy azok nem egyeznek meg. Ez a Dolev-Yao támadási modell alkalmazását jelenti, ami azt mondja ki, hogy a kommunikációs hálózatot úgy kell tekintenünk, hogy azt a támadó teljes mértékben lehallgathatja, megváltoztathatja az üzeneteket, új üzeneteket generálhat.
Így, ha $CH(ch_i,pub)$ és $S(ch_i,A,t_1,n_A)$, akkor $R(ch_i,B,t_2,n_B)$.

Védett csatornák esetén, ha $CH(ch_i, sec)$ és $S(ch_i, A, t_1, n_A)$, akkor $R(ch_i, B, t_2, n_A)$.

Ezekkel a bővítésekkel már alkalmasnak bizonyul a CSN rendszer a többcsatornás kriptográfiai protokollok tanulmányozására. Alkalmazásként a MANA protokollcsalád több tagját is megvizsgáltuk. Eredményeinket a következő fejezetek tartalmazzák.

5.4. A MANA protokollcsalád

Négy protokoll (és néhány variáns) tartozik jelenleg a MANA protokollcsaládba. (MANA I-IV, MA-DH, etc.) A MANA elnevezés az angol **MAN**ual **A**uthentication Protocol névből alakult ki. A protokollok célja a személyi hálózatok esetén felmerülő titkosítási feladatok megoldásának előkészítése, a kriptográfiai inicializálás. A protokollok közötti leglényegesebb különbség a rendelkezésre álló eszközök (billentyűzet, LED, képernyő, beviteli gomb, stb.) között van. Az alkalmazott nyilvános csatorna gyors és szélessávú, a nem nyilvános, védett csatorna tipikusan keskenysávú és lassú (általában manuális, a felhasználó írja és olvassa a csatornajeleket). [68][156][66][93][157]

Ezek a protokollok fontos szerepet játszottak abban a SHAMAN nevű projektben is, amelyben vezető európai mobiltelefon gyártó cégek és az Európai Közösség Information Society Technologies programja vett részt. [68][156]

A következő részekben a protokollok vizsgálata során először leírjuk az adott protokollal kapcsolatos legfontosabb tudnivalókat. Ezután felsoroljuk a protokollok lépéseit, majd a kezdeti feltevéseket rögzítjük. A protokoll tényleges formális leírása és azoknak a célállításoknak a tételszerű megfogalmazása a következő lépés, amelyeket vizsgálni kívánunk. A célállítás bizonyítása és az eredmény értékelése, értelmezése zárja a vizsgálati folyamatot.

5.4.1. MANA I

A MANA I protokollban az A-val és B-vel jelölt egység akar egy közös n_A üzenetet egyeztetni. Például ez az üzenet lehet a két eszköz publikus kulcsainak konkatenációja, vagy más kriptográfiai inicializáló paraméter. Az A egységnek egy kijelzője (*display*) és egy egyszerű nyomógombja (bináris kapcsoló) van. A másik, B egység billentyűzettel és egy LED-del (*Light Emitter Dioda* - kijelző eszköz) rendelkezik. Az eszközök egy nyilvános szélessávú ch_1 csatornát (például vezeték nélküli összeköttetést) használnak. A protokollban közreműködik egy U-val jelölt felhasználó, aki az eszközöket kezeli, felügyeli. U két biztonságosnak tekintett ch_2 és ch_3 csatornát használ az eszközökkel való kapcsolattartásra. A ch_1 csatorna minden felhasználó számára elérhető, publikus.

Itt és a többi protokollban is alkalmazásra kerülnek a már tárgyalt kulcsolt hash függvények. Az $h(m, k)$ jelöli a teljes konstrukciót, az ellenőrző összeget. Az h jelöli a hash függvényt, k kulcs típusú változó, m a függvény üzenet típusú paramétere.

A MANA I protokoll lépései

1-2. lépés	Az A egység a n_A üzenetet küldi B egységnek a ch_1 csatornán. A B egység a n_B üzenetet fogadja ch_1-en. Ez a szélessávú csatorna nyilvános, nem védett. Ez a jelölés feltételezi azt a lehetőséget, hogy az átküldött üzenet módosulhat a támadó fél által: $n_A \neq n_B$.
3-4. lépés	Az A egység generál egy véletlen k kulcsot, és kiszámítja az $h(n_A, k)$ ellenőrző összeget. Ezután az A egység elküldi $h(n_A, k)$ és k üzenetdarabokat az U felhasználónak a ch_2 csatornán. Ez a gyakorlatban azt jelenti, hogy az A eszköz képernyőjén megjeleníti a k és $h(n_A, k)$ számokat U számára. U fogadja az üzenetet, és továbbítja azt B felé a ch_3 csatornán. A gyakorlati megvalósítás: U begépeli az üzenetet a B eszköz billentyűzetén. (Ebben a protokollban az adatbevitel *„vakon”* történik, B-nek nincs kijelzője.)

5. lépés A B egység kiszámítja a $h(n_B, k)$ értéket a megkapott paraméterek felhasználásával, és összehasonlítja azt a kapott $h(n_A, k)$ ellenőrző összeggel.
Jelölje az x üzenet típusú változó az összehasonlítás eredményét oly módon, hogy x="1", ha $h(n_A, k) = h(n_B, k)$ és "0" egyébként. Az összehasonlítás a logikai rendszerben formulák egyenlőségének vizsgálatát jelenti.
Ezután B átküldi x-et az U felhasználónak, a ch_3 csatornát használva. A gyakorlatban ez azt jelenti, hogy B a LED-et használja, azon jelzi az eredményt. U a LED felvillanását figyeli, így kapja meg az x-et.
U továbbítja az x üzenetet az A eszköz felé a ch_2 csatornán. Ez a gyakorlatban azt jelenti, hogy U az A eszköz bináris kapcsolóját használja.

6. lépés A megkapja a küldött x értéket, így A ismeri annak az összehasonlításnak az eredményét, amelyet B végzett. □

Kezdeti feltételek rögzítése

Ebben a részben a csatornatulajdonságokat és más fontos protokollparamétereket rögzítünk. A paraméterek rögzítése már szorosan hozzátartozik a protokollvizsgálat menetéhez. Ezek a feltételek a bizonyítás során is felhasználásra kerülnek.

I11. $CH(ch_1, pub)$; $CH(ch_2, sec)$; $CH(ch_3, sec)$.

I12. $ENT_{ch_2} = \{A, U\}$; $ENT_{ch_3} = \{B, U\}$.

I13. $K_{B,t_7}(h(n_A, k) = h(n_A, k)) \rightarrow S(ch_3, B, t_7, \text{"1"})$

I14. $K_{B,t_7}(h(n_A, k) \neq h(n_A, k)) \rightarrow S(ch_3, B, t_7, \text{"0"})$

I15. $\forall t' > t_3\ R(ch_2, A, t', \text{"1"}) \rightarrow K_{A,t'}(n_A = n_B)$.

I16. $\forall t' > t_3\ R(ch_2, A, t', \text{"0"}) \rightarrow K_{A,t'}(n_A \neq n_B)$.

A MANA I protokoll formális alakja

$t_1, \ldots t_{10}$ jelöli az egymást követő időpontokat a protokollban. B az x üzenet típusú változó értékét az 5. lépésben leírtak szerint számítja.

1. $S(ch_1, A, t_1, n_A)$; $R(ch_1, B, t_2, n_B)$
2. $S(ch_2, A, t_3, \{k, h(n_A, k)\})$; $R(ch_2, U, t_4, \{k, h(n_A, k)\})$
3. $S(ch_3, U, t_5, \{k, h(n_A, k)\})$; $R(ch_3, B, t_6, \{k, h(n_A, k)\})$.
4. $S(ch_3, B, t_7, x)$; $R(ch_3, U, t_8, x)$.
5. $S(ch_2, U, t_9, x)$; $R(ch_2, A, t_{10}, x)$.

A protokoll célja - tétel és bizonyítás

A MANA I protokollal kapcsolatban a következő állítást fogalmazhatjuk meg.

Tétel 5.4.1 *A MANA I protokoll végén mind A, mind B tudja, hogy $n_A = n_B$ teljesül vagy sem.*

$$n_A = n_B \rightarrow K_{A,t_{10}}(n_A = n_B) \wedge K_{B,t_{10}}(n_A = n_B)$$
$$n_A \neq n_B \rightarrow K_{A,t_{10}}(n_A \neq n_B) \wedge K_{B,t_{10}}(n_A \neq n_B).$$

Bizonyítás Az első lépés és az $A5'(a)$ és $A6'(a)$ axióma alapján

$$L_{A,t_1} n_A \ , \tag{1}$$
$$L_{B,t_2} n_B \ . \tag{2}$$

A második lépés és $A5'(a)$ alapján

$$L_{A,t_3}\{k, h(n_A, k)\} \ . \tag{3}$$

(3) és $(M3)$ adja, hogy

$$L_{A,t_3}k\ , \tag{4}$$
$$L_{A,t_3}h(n_A,k)\ . \tag{5}$$

$A6'(a)$ alapján

$$L_{U,t_4}\{k,h(n_A,k)\}\ . \tag{6}$$

(6) és $(M3)$ adja, hogy

$$L_{U,t_4}k\ , \tag{7}$$
$$L_{U,t_4}h(n_A,k)\ . \tag{8}$$

A harmadik lépés és $A6'(a)$ alapján

$$L_{B,t_6}\{k,h(n_A,k)\}\ . \tag{9}$$

(9) és $(M3)$ adja, hogy

$$L_{B,t_6}k\ , \tag{10}$$
$$L_{B,t_6}h(n_A,k)\ . \tag{11}$$

A 4. lépésben B képes kiszámítani $h(n_B,k)$ értékét, mert (2) és (10) $A3(a)$ axióma alapján

$$L_{B,t_7}n_B\ , \tag{12}$$
$$L_{B,t_7}k\ . \tag{13}$$

Így az $A16(a)$ axióma alapján

$$L_{B,t_7}h(n_B,k)\ . \tag{14}$$

t_7-ben B ismeri $h(n_A,k)$-t, hiszen (11) és $A3(a)$ alapján $L_{B,t_7}h(n_A,k)$. A logikai formulákra vonatkozó induktív definíció (b) része alapján B el tudja végezni a $h(n_A,k)$ és $h(n_B,k)$ üzenetdarabok összehasonlítását. Az $n_A = n_B$ feltétel és a közös k kulcs alapján

$h(n_A, k) = h(n_B, k)$, így $K_{B,t_7}(n_A = n_B))$ kell teljesüljön - ami az $A3(b)$ axiómával az állítás második részét adja. $I13.$ alapján a 4. lépésben az $S(ch_3, B, t_7, "1")$ teljesül, ami a védett ch_3 és ch_2 csatornákon történő adatátvitel miatt (M5) az 5. lépésben azt eredményezi, hogy A az "1" üzenetet fogadja:

$$R(ch_2, A, t_{10}, "1") \ . \tag{15}$$

Ez viszont az $I15$ kezdeti feltétellel a keresett $K_{A,t_{10}}(n_A = n_B)$ első állításrészt adja.

$n_A \neq n_B$ esetén az előző levezetés (14) pontja után B szintén képes az összehasonlítás elvégzésre, de ennek eredményére $h(n_A, k) \neq h(n_B, k)$ kell adódjon. Ebből következik, hogy $K_{B,t_7}(n_A \neq n_B))$ kell teljesüljön - ami az $A3(b)$ axiómával az állítás második részét adja. $I14.$ alapján a 4. lépésben az $S(ch_3, B, t_7, "0")$ teljesül, ami a védett ch_3 és ch_2 csatornákon történő adatátvitel miatt (M5) az 5. lépésben azt eredményezi, hogy A az "0" üzenetet fogadja:

$$R(ch_2, A, t_{10}, "0") \ . \tag{16}$$

Ez viszont az $I16.$ kezdeti feltétellel a másik, $K_{A,t_{10}}(n_A \neq n_B)$ állításhoz vezet. □

Elmondhatjuk tehát, hogy a MANA I protokoll teljesíti a tételben megfogalmazott kitűzött célt.

5.4.2. MANA II

Ez a protokoll a MANA I protokoll egyszerű variánsa. Mindkét eszköz (A és B) kijelzővel és egyszerű input lehetőséggel (kétállású kapcsoló) rendelkezik. A protokoll legfőbb biztonsági lépése a 4. lépés - felhasználva a ch_3 csatorna tulajdonságait -, ahogy az analízisből is kitűnik.

A MANA II protokoll lépései

1. lépés Az A egység a n_A üzenetet küldi a B egységnek a ch_1 csatornán. B a n_B üzenetet fogadja a ch_1 csatornán, (ch_1 nem védett csatorna, hasonlóan a MANA I protokollhoz).
2. lépés A generál egy k_A kulcsot, és kiszámítja az $h(n_A, k_A)$ értéket. Ezek után az A egység elküldi a $\{k_A, h(n_A, k_A)\}$ üzenetet az U felhasználónak a védett ch_2 csatornán. A gyakorlatban ez azt jelnti, hogy a k_A és $h(n_A, k_A)$ értékek megjelennek az A egység kijelzőjén. U innen olvashatja le azokat.
3. lépés Az A egység elküldi a k_A kulcs értékét a B egységnek a ch_1 csatronán. B k_B értéket kap a nem védett csatornán.
4. lépés A B egység kiszámítja az $h(n_B, k_B)$ értéket, és elküldi a $\{k_B, h(n_B, k_B)\}$ üzenetet az U felhasználónak a védett ch_3 csatornán. Ez a gyakorlatban azt jelenti, hogy a B egység megjeleníti a k_B és $h(n_B, k_B)$ értéket a kijelzőjén. U innen olvashatja le azokat.
5. lépés U összehasonlítja k_A és k_B valamint $h(n_A, k_A)$ és $h(n_B, k_A)$ értékeket. Jelölje x üzenet típusú változó az összehasonlítás eredményét oly módon, hogy x="1", ha $k_A = k_B$ és $h(n_A, k_A) = h(n_B, k_B)$ és "0" egyébként.
   Az összehasonlítás a logikai rendszerben formulák egyenlőségének vizsgálatát jelenti.
   U átküldi x értékét az A egységnek a ch_2 védett csatornán.
6. lépés U átküldi x értékét a B egységnek a ch_3 védett csatornán. □

Kezdeti feltételek rögzítése

A MANA II protokollra vonatkozó kezdeti feltételek a következők.

I21. $CH(ch_1, pub)$; $CH(ch_2, sec)$; $CH(ch_3, sec)$.

I22. $ENT_{ch_2} = \{A, U\}$; $ENT_{ch_3} = \{B, U\}$.

I23. $K_{U,t_9}(h(n_A, k_A) = h(n_B, k_B)) \rightarrow S(ch_2, U, t_9, \text{"1"}) \wedge S(ch_3, U, t_{11}, \text{"1"})$.

I24. $K_{U,t_9}(h(n_A, k_A) \neq h(n_B, k_B)) \rightarrow S(ch_2, U, t_9, \text{"0"}) \wedge S(ch_3, U, t_{11}, \text{"0"})$.

I25. $R(ch_2, A, t_{10}, \text{"1"}) \rightarrow K_{A,t_{10}}(n_A = n_B)$.

I24. $R(ch_3, B, t_{12}, \text{"1"}) \rightarrow K_{B,t_{12}}(n_A = n_B)$.

I25. $R(ch_2, A, t_{10}, \text{"0"}) \rightarrow K_{A,t_{10}}(n_A \neq n_B)$.

I26. $R(ch_3, B, t_{10}, \text{"0"}) \rightarrow K_{A,t_{12}}(n_A \neq n_B)$.

A MANA II protokoll formális alakja

Mint ahogy az előzőekben is, $t_1, \ldots t_{10}$ jelöli az egymást követő időpontokat a protokollban. B az x értékét az 5. lépés szerint számítja.

1. $S(ch_1, A, t_1, n_A)$; $R(ch_1, B, t_2, n_B)$.
2. $S(ch_2, A, t_3, \{k_A, h(n_A, k_A)\})$; $R(ch_2, U, t_4, \{k_A, h(n_A, k_A)\})$.
3. $S(ch_1, A, t_5, k_A)$; $R(ch_1, B, t_6, k_B)$.
4. $S(ch_3, B, t_7, \{k_B, h(n_B, k_B)\})$; $R(ch_3, U, t_8, \{k_B, h(n_B, k_B)\})$.
5. $S(ch_2, U, t_9, x)$; $R(ch_2, A, t_{10}, x)$.
6. $S(ch_3, U, t_{11}, x)$; $R(ch_3, B, t_{12}, x)$.

A protokoll céljai - tételek és bizonyítások

Tétel 5.4.2 *Tegyük fel, hogy a n_A és n_B paraméterek nem egyenlők. Ekkor a MANA II protokoll végén mind A, mind B tudja azt, hogy $n_A \neq n_B$. Formalizálva:*

$$n_A \neq n_B \rightarrow K_{A,t_{12}}(n_A \neq n_B) \wedge K_{B,t_{12}}(n_A \neq n_B).$$

Bizonyítás Az első lépés és az $A5'(a)$ és $A6'(a)$ axióma alapján

$$L_{A,t_1} n_A \ , \tag{1}$$
$$L_{B,t_2} n_B \ . \tag{2}$$

A második lépés és $A5'(a)$ alapján

$$L_{A,t_3}\{k_A, h(n_A, k_A)\} \ . \tag{3}$$

(3) és $(M3)$ adja, hogy

$$L_{A,t_3} k_A \ , \tag{4}$$
$$L_{A,t_3} h(n_A, k_A) \ . \tag{5}$$

$A6'(a)$ alapján

$$L_{U,t_4}\{k_A, h(n_A, k_A)\} \ . \tag{6}$$

(6) és $(M3)$ adja, hogy

$$L_{U,t_4} k_A \ , \tag{7}$$
$$L_{U,t_4} h(n_A, k_A) \ . \tag{8}$$

A negyedik lépés és $A6'(a)$ alapján

$$L_{U,t_8}\{k_B, h(n_B, k_B)\} \ . \tag{9}$$

(9) és $(M3)$ adja, hogy

$$L_{U,t_8} k_B \ , \tag{10}$$
$$L_{U,t_8} h(n_B, k_B) \ . \tag{11}$$

Az $A3(a)$ axióma alapján t_9-re U ismeri mind a $h(n_A, k_A)$, mind a $h(n_B, k_B)$ egyirányú függvények értékeit. Az $A17(a)$ axióma kontrapozíciója alapján az értékelés eredménye a kezdőfeltételt figyelembe véve: $h(n_A, k_A) \neq h(n_B, k_B)$. Így $I24$. alapján

$$S(ch_2, U, t_9, \text{"0"}) \ , \tag{12}$$
$$S(ch_3, U, t_{11}, \text{"0"}) \tag{13}$$

teljesül az 5. és 6. lépésben. Ez azt eredményezi a védett csatornák miatt (M5), hogy

$$R(ch_2, A, t_{10}, \text{"0"}) \ , \tag{14}$$
$$R(ch_3, B, t_{12}, \text{"0"}) \ . \tag{15}$$

Az $I27.$ és $I28.$ alapján adódik

$$K_{A,t_{10}}(n_A \neq n_B) \ , \tag{16}$$
$$K_{B,t_{12}}(n_A \neq n_B) \ . \ \square \tag{17}$$

Tétel 5.4.3 *$n_A = n_B$ nem garantálja azt, hogy a MANA II protokoll végén A és B tudja azt, hogy $n_A = n_B$. Formalizálva:*

$$n_A = n_B \rightarrow \neg K_{A,t_{12}}(n_A = n_B) \wedge \neg K_{B,t_{12}}(n_A = n_B)$$

Bizonyítás Az előző bizonyítás részleteire támaszkodva tudjuk, hogy t_9-re U ismeri mind a $h(n_A, k_A)$, mind a $h(n_B, k_B)$ egyirányú függvények értékeit. Amennyiben $k_A \neq k_B$, az $A17(a)$ axióma kontrapozíciójára hivatkozva: $h(n_A, k_A) \neq h(n_B, k_B)$. Ez azt vonja maga után, hogy hasonlóan az előző tételhez

$$K_{A,t_{10}}(n_A \neq n_B) \ ,$$
$$K_{B,t_{12}}(n_A \neq n_B) \ .$$

adódik. A kiinduló feltétel szerint $n_A = n_B$. Amiből viszont következik az állítás. $\square$

Így kijelenthetjük, hogy a MANA II protokoll a kitűzött célokat csak részben teljesíti. A felek hiába rendelkeznek a helyes n_A értékkel, annak közös elfogadhatóságát a protokoll nem tudja garantálni. Mivel a felhasznált kulcsok nyilvános csatornán kerülnek továbbításra, egy támadó fél módosítani tudja értéküket, megzavarva a protokollt. Ismételt kulcsküldés már a protokoll hatáskörén kívül esik. Az újra és újra lejátszott eredménytelen protokoll-ismétlések pedig a felhasználók bizalmának elvesztését jelenthetik.

Hasonló eset alakulhat ki, mint amikor biztonsági rendszereket kapcsolnak ki a generált téves riasztások elkerülésére. Amenynyiben a $k_A = k_B$ egyenlőséget garantálni lehet, akkor a hiányzó bizonyítási rész az előzőekhez hasonlóan igazolható.

A protokollt módosíthatjuk úgy, hogy a 2. és 4. lépésben $h(n_A, k_A)$ és $h(n_B, k_B)$ kulcsolt hash értékek helyett *„hagyományos"* hash függvényeket ($H(n_A)$, $H(n_B)$ - 2.1.4. fejezet - MD sorozat, SHA sorozat, HAVAL, RIPEM sorozat, stb. [45][113]) használunk. Ekkor feleslegessé válik a kulcsok alkalmazása és a kulcsok átküldése.

A kiinduló feltételek:

I21'. $CH(ch_1, pub)$; $CH(ch_2, sec)$; $CH(ch_3, sec)$.

I22'. $ENT_{ch_2} = \{A, U\}$; $ENT_{ch_3} = \{B, U\}$.

I23'. $K_{U,t_7}(H(n_A) = H(n_B)) \rightarrow S(ch_2, U, t_7, \text{"1"}) \wedge S(ch_3, U, t_9, \text{"1"})$.

I24'. $K_{U,t_7}(H(n_A) \neq H(n_B)) \rightarrow S(ch_2, U, t_7, \text{"0"}) \wedge S(ch_3, U, t_9, \text{"0"})$.

I25'. $R(ch_2, A, t_8, \text{"1"}) \rightarrow K_{A,t_8}(n_A = n_B)$.

I24'. $R(ch_3, B, t_{10}, \text{"1"}) \rightarrow K_{B,t_{10}}(n_A = n_B)$.

I25'. $R(ch_2, A, t_8, \text{"0"}) \rightarrow K_{A,t_8}(n_A \neq n_B)$.

I26'. $R(ch_3, B, t_{10}, \text{"0"}) \rightarrow K_{A,t_{10}}(n_A \neq n_B)$.

A **MANA II'** protokoll formalizált lépései:

1'. $S(ch_1, A, t_1, n_A)$; $R(ch_1, B, t_2, n_B)$.

2'. $S(ch_2, A, t_3, H(n_A)\})$; $R(ch_2, U, t_4, H(n_A)\})$.

3'. $S(ch_3, B, t_5, H(n_B)\})$; $R(ch_3, U, t_6, H(n_B)\})$.

4'. $S(ch_2, U, t_7, x)$; $R(ch_2, A, t_8, x)$.

5'. $S(ch_3, U, t_9, x)$; $R(ch_3, B, t_{10}, x)$.

A protokoll ilyen irányú változtatása után kimondhatjuk a következő tételt:

Tétel 5.4.4 *A fentiek alapján módosított MANA II' protokoll elején helyesen átküldött n_A paraméter garantálja, hogy a protokoll lezárásakor A és B is tudja, hogy $n_A = n_B$. Formalizálva:*

$$n_A = n_B \rightarrow K_{A,t_8}(n_A = n_B) \wedge K_{B,t_{10}}(n_A = n_B)$$

valamint $n_A \neq n_B$ esetén a MANA II' protokoll lefutása után mind A és mind B tudja, hogy $n_A \neq n_B$

$$n_A \neq n_B \rightarrow K_{A,t_8}(n_A \neq n_B) \wedge K_{B,t_{10}}(n_A \neq n_B).$$

Bizonyítás Az előző bizonyítások vázlatát követjük. Az első lépés és az $A5'(a)$ és $A6'(a)$ axióma alapján

$$L_{A,t_1} n_A \ , \tag{1}$$
$$L_{B,t_2} n_B \ . \tag{2}$$

A második lépés és $A5'(a)$ alapján

$$L_{A,t_3} H(n_A) \ . \tag{3}$$

$A6'(a)$ alapján

$$L_{U,t_4} H(n_A) \ . \tag{4}$$

A harmadik lépésben $A5'(a)$ és $A6'(a)$ alapján

$$L_{B,t_5} H(n_B) \ , \tag{5}$$
$$L_{U,t_6} H(n_B) \ . \tag{6}$$

Az $A3(a)$ axióma alapján t_7-re U ismeri mind a $H(n_A)$, mind a $H(n_B)$ egyirányú függvények értékeit. Az $A19(a)$ axióma és az $n_A = n_B$

feltétel alapján az értékelés eredménye a kezdőfeltételt figyelembe véve: $H(n_A) = H(n_B)$. Így $I23.$ alapján

$$S(ch_2, U, t_7, \text{"1"}) \,, \tag{7}$$
$$S(ch_3, U, t_9, \text{"1"}) \tag{8}$$

teljesül az 4'. és 5'. lépésben. Ez azt eredményezi a védett csatornák miatt (M5), hogy

$$R(ch_2, A, t_8, \text{"1"}) \,, \tag{9}$$
$$R(ch_3, B, t_{10}, \text{"1"}) \,. \tag{10}$$

Az $I25'.$ és $I26'.$ alapján adódik

$$K_{A,t_8}(n_A = n_B) \,, \tag{11}$$
$$K_{B,t_{10}}(n_A = n_B) \,. \tag{12}$$

Amennyiben $n_A \neq n_B$, akkor a (6) pont után az $A19(a)$ axióma kontrapozíciója és az $I24.$ feltétel alapján

$$S(ch_2, U, t_7, \text{"0"}) \,, \tag{13}$$
$$S(ch_3, U, t_9, \text{"0"}) \tag{14}$$

Ez azt eredményezi a védett csatornák miatt (M5), hogy

$$R(ch_2, A, t_8, \text{"0"}) \,, \tag{15}$$
$$R(ch_3, B, t_{10}, \text{"0"}) \,. \tag{16}$$

Az $I25.$ és $I26.$ alapján adódik

$$K_{A,t_8}(n_A \neq n_B) \,, \tag{17}$$
$$K_{B,t_{10}}(n_A \neq n_B) \,. \,\square \tag{18}$$

A módosított protokoll kialakítása során természetesen figyelembe kell venni a szakirodalom ajánlásait, amelyekről többek között [113]-ben olvashatunk. A paraméterek beállítása során azon támadásoknak, amely a hash függvény alkalmazásakor előre beépített paraméterek ismeretén alapul gátja, hogy egyelőre nem ismeretes olyan algoritmikus támadás az általánosan használt hash-függvények esetén, amely tetszőleges input szöveghez vele azonos hash-értékkel rendelkező másik szöveget konstruál. Ugyanakkor az is elmondható, hogy a véletlen kulcsok megfelelő minőségű generálásához (eredeti protokoll 2. lépés) szintén megfelelő kezdeti paraméterek kellenek.

5.4.3. MANA III

Ebben a protokollban két eszköz (A és B) és az eszközöket kezelő felhasználó (U - user) vesz részt. Mindkét eszköz rendelkezik egy-egy input egységgel, ami jelen esetben billentyűzet, és egy-egy output egységgel, ami egy világító dióda (LED). A protokoll célja az, hogy mindkét eszköz bizonyítottan rendelkezzen ugyanazzal a kezdeti paraméterrel (n_A), amelyet a későbbi védett kommunikáció során használhat fel.

A MANA III protokoll lépései

1. lépés — Az A eszköz generál egy n_A számot. Ezt és azonosítóját (A) átküldi a B eszköznek a ch_1 csatornán. A B eszköz egy n_B számot és Σ azonosítót kap a ch_1 csatornán (a ch_1 csatorna nem védett, így feltételezzük, hogy egy támadó képes megváltoztatni az üzenet tartalmát (fennállhat $n_A \neq n_B$ és $\Sigma \neq A$), ezt jelöljük ezen a módon).
2. lépés — A B eszköz a ch_1 csatornán elküldi az B azonosítóját. Az A eszköz Ψ számot fogad a ch_1 csatornán (hasonlóan fennállhat, hogy $\Psi \neq B$).
3. lépés — Az U user egy r_U véletlenszámot generál, és ezt a védett ch_2 és ch_3 csatornákon eljuttatja az A és a B félhez.

4. lépés Az A eszköz egy k_A véletlenszámot generál és kiszámítja az $m_1 = h(\{A, n_A, r_U\}, k_A)$ számot.

5. lépés Az A eszköz elküldi az m_1-t B-nek a ch_1 csatornán. B m_{11}-t kap üzenetként (fennállhat, hogy $m_1 \neq m_{11}$).

6. lépés B egy k_B véletlenszámot generál és kiszámítja az $m_2 = h(\{B, n_B, r_U\}, k_B)$ számot.

7. lépés B elküldi m_2-t A-nak a ch_1 csatornán. A m_{22}-t kap üzenetként (fennállhat, hogy $m_2 \neq M_{22}$).

8. lépés Miután A fogadja m_{22}-t B-től (és nem előbb), A átküldi a k_A kulcsot B-nek a ch_1 csatornán (B k_Σ-et kap, fennállhat, hogy $k_\Sigma \neq k_A$).

9. lépés Amikor B megkapja az m_{11} értéket A-tól (és nem előbb), B átküldi a k_B számot A-nak a ch_1 csatornán (A k_Ψ-t kap, fennállhat, hogy $k_\Psi \neq k_B$).

10. lépés A újraszámítja m_2-t, m_{222}-t kap eredményül. Amennyiben ez megegyezik a B-től kapott m_{22} értékkel, úgy A erről egy jelet küld (világító LED) U-nak a ch_2 csatornán. $m_{222} = h(\{\Psi, n_A, r_U\}, k_\Psi)$ a kapott üzenetek alapján.

11. lépés B újraszámítja m_1-t, m_{111}-t kap eredményül. Amennyiben ez megegyezik az A-tól kapott m_{11} értékkel, úgy B erről egy jelet küld (világító LED) U-nak a ch_3 csatornán. $m_{111} = h(\{\Sigma, n_B, r_U\}, k_\Sigma)$ a kapott üzenetek alapján.

12. lépés Amennyiben mindkét eszköz sikeres számítást jelez (és csak ekkor), U visszajelzi ezt mindkét eszköznek. □

Kezdeti feltételek rögzítése

I301. $CH(ch_1, pub)$; $CH(ch_2, sec)$; $CH(ch_3, sec)$.

I302. $ENT_{ch_2} = \{A, U\}$; $ENT_{ch_3} = \{B, U\}$.

I303. $K_{A,t_{17}}(m_{22} = m_{222}) \rightarrow S(ch_2, A, t_{17}, \text{"1"})$.

I304. $K_{A,t_{19}}(m_{11} = m_{111}) \rightarrow S(ch_3, B, t_{19}, \text{"1"})$.

I305. $K_{A,t_{17}}(m_{22} \neq m_{222}) \rightarrow S(ch_2, A, t_{17}, \text{"0"})$.

I306. $K_{A,t_{19}}(m_{11} \neq m_{111}) \rightarrow S(ch_3, B, t_{19}, \text{"0"})$.

I307. $K_{U,t_{21}}(R(ch_2, U, t_{18}, \text{"1"}) \wedge R(ch_3, U, t_{20}, \text{"1"}))$
$\rightarrow S(ch_2, U, t_{21}, \text{"1"}) \wedge S(ch_3, U, t_{23}, \text{"1"})$

I308. $K_{U,t_{21}}(R(ch_2, U, t_{18}, \text{"0"}) \vee R(ch_3, U, t_{20}, \text{"0"}))$
$\rightarrow S(ch_2, U, t_{21}, \text{"0"}) \wedge S(ch_3, U, t_{23}, \text{"0"})$

I309. $R(ch_2, A, t_{22}, \text{"1"}) \rightarrow K_{A,t_{22}}(n_A = n_B)$.

I310. $R(ch_3, B, t_{24}, \text{"1"}) \rightarrow K_{B,t_{24}}(n_A = n_B)$.

I311. $R(ch_2, A, t_{22}, \text{"0"}) \rightarrow K_{A,t_{22}}(n_A \neq n_B)$.

I312. $R(ch_3, B, t_{24}, \text{"0"}) \rightarrow K_{A,t_{24}}(n_A \neq n_B)$.

A MANA III protokoll formális alakja

1. $S(ch_1, A, t_1, \{n_A, A\})$; $R(ch_1, B, t_2, \{n_B, \Sigma\})$
2. $S(ch_1, B, t_3, B)$; $R(ch_1, A, t_4, \Psi)$
3. $S(ch_2, U, t_5, r_U)$; $R(ch_2, A, t_6, r_U)$
4. $S(ch_3, U, t_7, r_U)$; $R(ch_3, B, t_8, r_U)$
5. $S(ch_1, A, t_9, m_1)$; $R(ch_1, B, t_{10}, m_{11})$
6. $S(ch_1, B, t_{11}, m_2)$; $R(ch_1, A, t_{12}, m_{22})$
7. $S(ch_1, A, t_{13}, k_A)$; $R(ch_1, B, t_{14}, k_\Sigma)$
8. $S(ch_1, B, t_{15}, k_B)$; $R(ch_1, A, t_{16}, k_\Psi)$
9. $S(ch_2, A, t_{17}, x)$; $R(ch_2, U, t_{18}, x)$
10. $S(ch_3, B, t_{19}, y)$; $R(ch_3, U, t_{20}, y)$

11. $S(ch_2, U, t_{21}, z)$; $R(ch_2, A, t_{22}, z)$

12. $S(ch_3, U, t_{23}, z)$; $R(ch_3, B, t_{24}, z)$

A protokoll céljai - tételek és bizonyítások

Tétel 5.4.5 *Tegyük fel, hogy a n_A és n_B paraméterek nem egyenlők a protokoll végrehajtása során (egy illetéktelen felhasználó módosítja a kommunikációt). Ekkor a MANA III protokoll lefutásának a végén az A és B partnerek (eszközök) mindketten tudják azt, hogy $n_A \neq n_B$. Formálisan:*

$$n_A \neq n_B \rightarrow K_{A,t_{22}}(n_A \neq n_B) \wedge K_{B,t_{24}}(n_A \neq n_B).$$

Bizonyítás Az első lépés és az $A5'(a)$ és $A6'(a)$ axióma alapján

$$L_{A,t_1}\{n_A, A\}\,, \tag{1}$$
$$L_{B,t_2}\{n_B, \Sigma\}\,. \tag{2}$$

(M3) alapján

$$L_{A,t_1} n_A\,, \tag{3}$$
$$L_{A,t_1} A\,, \tag{4}$$
$$L_{B,t_2} n_B\,, \tag{5}$$
$$L_{B,t_2} \Sigma\,. \tag{6}$$

A második lépésben $A5'(a)$ és $A6'(a)$ alapján

$$L_{B,t_3} B\,, \tag{7}$$
$$L_{A,t_4} \Psi\,. \tag{8}$$

A harmadik és negyedik lépésben $A5'(a)$ és $A6'(a)$ alapján

$$L_{U,t_5} r_U\,, \tag{9}$$
$$L_{A,t_6} r_U\,, \tag{10}$$
$$L_{B,t_8} r_U\,. \tag{11}$$

Az ötödik lépésben $A5'(a)$ és $A6'(a)$ alapján

$$L_{A,t_9} m_1 \ , \tag{12}$$

$$L_{B,t_{10}} m_{11} \ . \tag{13}$$

$(m_1 = h(\{A, n_A, r_U\}, k_A))$
A hatodik lépésben $A5'(a)$ és $A6'(a)$ alapján

$$L_{B,t_{11}} m_2 \ , \tag{14}$$

$$L_{A,t_{12}} m_{22} \ . \tag{15}$$

$(m_2 = h(\{B, n_B, r_U\}, k_B))$
A hetedik lépésben $A5'(a)$ és $A6'(a)$ alapján

$$L_{A,t_{13}} k_A \ , \tag{16}$$

$$L_{B,t_{14}} k_\Sigma \ . \tag{17}$$

A nyolcadik lépésben $A5'(a)$ és $A6'(a)$ alapján

$$L_{B,t_{15}} k_B \ , \tag{18}$$

$$L_{A,t_{16}} k_\Psi \ . \tag{19}$$

A protokoll szerint t_{16} után A újraszámítja m_2-t és m_{222}-t kap eredményül. m_{222} formailag, az A rendelkezésére álló, fogadott üzenetek alapján: $m_{222} = h(\{\Psi, n_A, r_U\}, k_\Psi)$. Eredetileg $m_2 = h(\{B, n_B, r_U\}, k_B)$. Amennyiben $m_2 = m_{22}$, $\Psi = B$ és $k_\Psi = k_B$, még akkor sem lehet $m_{22} = m_{222}$, mivel a tétel feltételei szerint $n_A \neq n_B$. Így $K_{A,t_{17}}(m_{22} \neq m_{222})$, amivel $I305.$ alapján $S(ch_2, A, t_{17}, \text{"0"})$. A védett csatornák miatt(M5) U az $R(ch_2, U, t_{18}, \text{"0"})$ üzenetet kapja, ami alapján az $I308.$ kezdeti feltétel érvényes. Ez azt jelenti, hogy $S(ch_2, U, t_{21}, \text{"0"}) \wedge S(ch_3, U, t_{23}, \text{"0"})$. A védett csatornák miatt (M5) $I311.$ és $I312.$ érvényes: $R(ch_2, A, t_{22}, \text{"0"})$ és $R(ch_3, B, t_{24}, \text{"0"})$. Tehát $K_{A,t_{22}}(n_A \neq n_B)$ és $K_{B,t_{24}}(n_A \neq n_B)$, ami a keresett állítás. Hasonló gondolatmenettel belátható, hogy a B egyed által elvégzett $m_{11} = m_{111}$ összehasonlítás is egyenlőtlenséget ad. Ez szintén a keresett állításhoz vezet. □

Tétel 5.4.6 *Tegyük fel, hogy a n_A és n_B paraméterek egyenlők a protokoll végrehajtása során. Ekkor a MANA III protokoll nem garantálja, hogy lefutásának a végén az A és B partnerek (eszközök) mindketten tudják azt, hogy $n_A = n_B$. Formálisan:*

$$n_A = n_B \rightarrow \neg K_{A,t_{22}}(n_A \neq n_B) \wedge \neg K_{B,t_{24}}(n_A \neq n_B).$$

Bizonyítás Az előző tétel bizonyítása alapján az A és B felhasználók eljutnak a fogadott m_{22} és számított m_{222} (m_{11} és m_{111}) összehasonlításához. Ez abban az esetben, ha a támadó aktívan nem avatkozik a protokollba, a keresett állításhoz vezet. Amennyiben a protokollt támadás éri, az $n_A = n_B$ feltételt megtartva, de az A vagy k_A értékek közül bármelyiket változtatva, a protokoll az $m_{22} = m_{222}$ vagy $m_{11} = m_{111}$ állítások bármelyikének elvetéséhez vezet. Ennek eredményeképpen $K_{A,t_{22}}(n_A \neq n_B)$ és $K_{B,t_{24}}(n_A \neq n_B)$. Ez azt jelenti, hogy hiába lett helyesen átküldve az n_A üzenet, mégsem fogadják el a felhasználók ezt. □

Ebben az esetben is elmondható tehát, hogy a felek hiába rendelkeznek a helyes n_A értékkel, annak közös elfogadhatóságát a protokoll nem tudja garantálni. Mivel az A, B egyednevek és a k_A, k_B kulcsok nyilvános csatornán kerülnek továbbításra, egy támadó fél módosítani tudja értéküket, megzavarva a protokollt. A fenti két tétel elemzése megmutatja, hogy a javítás a MANA II protokolléhoz hasonló módon nem oldható meg. Kulcshasználat nélküli $h(m)$ egyirányú függvények alkalmazása az összetett m üzenet miatt nem jelent megoldást. n_A, illetve n_B részeket tartalmaznia kell az átküldött üzeneteknek, de r_U elhagyása már megszemélyesítő támadást tesz lehetővé. A protokoll ezen hiányosságainak javítása új protokoll kidolgozását igényli.

5.4.4. Összefoglalás, további vizsgálatok

Összefoglalásként elmondhatjuk, hogy CSN-logika bemutatott bővítése alkalmas a többcsatornás protokollok vizsgálatára. Nyilvánvaló, hogy nem ez az egyetlen lehetséges megoldás a hasonló problémák vizsgálatára. A

bemutatott példák evidensnek és egyszerűnek tűnhetnek, viszont megnyithatnak olyan vizsgálati utakat, amelyek komolyabb protokoll-hibákra is rámutathatnak.

Többcsatornás protokollokat a vizsgált három protokollon kívül más körben is alkalmaznak. Szaporodik azoknak az alkalmazásoknak a köre, amelyek összekapcsolják az Inernetet és a személyes kommunikációs eszközöket. A mobiltelefonra küldött SMS, amely a banki szolgáltatások elérését teszi védettebbé; a mobiltelefonokkal történő hang- és képátvitel (2D-s BAR-kódok, biometriai azonosítás, stb.) összekapcsolva más kommunikációs csatornákkal (Internet, fax, stb.), mind azt erősítik, hogy a többcsatornás protokolloknak jogosultsága van a kommunikációs fejlődésben.

6. fejezet

Kitekintés - A protokollvizsgálat további lehetőségei

Ebben a fejezetben célunk - a formális módszerekhez kapcsolódva -, betekintést nyújtani más tudományterületek hasonló irányú kutatásaiba. Először az informatikában megjelenő protokollfogalommal foglalkozunk. Ehhez szorosabban kapcsolódnak a dolgozat fő tárgykörébe eső kriptográfiai protokollok. Ezt követően eltávolodunk az informatikától, az orvostudományban és a gazdasági tudományokban vizsgáljuk a protokollok szerepét.

6.1. Informatikai protokollok

Az informatikai hálózatok világában egy protokoll egy szabványt, egyezményt jelent, amely leírja, hogy a hálózat szereplői miképp tudnak egymással kommunikálni. Ez általában a kapcsolat felépítését, az adattovábbítást és a kapcsolat szabályos megszüntetetését foglalja magába. Gyakorlati szempontból egy protokoll meghatározza, hogy a résztvevők milyen sorrendben, milyen tartalmú üzeneteket váltsanak egymással a kommunikációs cél elérése érdekében. Egyes protokollok igen kötött formában

rögzítik a részleteket, mások csak kereteket határoznak meg a fejlesztők számára.

Egy protokollt nyilvánvalóan tervezni kell. Figyelembe kell venni, hogy a protokoll hogyan gazdálkodjon az erőforrásokkal (hatékonyság). Számítani kell a kommunikáció során bekövetkező hibákra és zavarokra, azokat hiba-detektáló és -javító algoritmusokkal kezelni kell (megbízhatóság). Ugyancsak figyelni kell az eltérő számítógépes konfigurációk összehangolására is (skálázhatóság).

Mindezek megvalósítására a szakemberek olyan szervezetekbe tömörülnek, amelyek szabványokat, előírásokat bocsátanak ki a protokollokkal kapcsolatban. Az Interneten használt protokollokat az IETF (Internet Engineering Task Froce) tartatja karban. A szervezet RFC-kben (Request for Comment) rögzíti a fejlesztési folyamat során már lezárt követelményeket.

A protokollok fejlesztése során sok olyan eszközt, modellező rendszert, alkalmazást alakítottak ki, amelyek a fent említett célok minél teljesebb körű megvalósítását teszik lehetővé. [16] Erről röviden a 3.2. fejezet is említést tesz.

A protokoll-fogalom nem csak a hálózatok körében, hanem a programozáselméletben, a számítási folyamatok modellezésében, operációs rendszerek vizsgálatakor, stb. is megjelenik.

Elmondható tehát, hogy az informatikában igen elterjedt a protokollfogalom és a köré épülő módszerek (interface, modularitás, stb.) felhasználása, alkalmazása.

6.2. Orvosi protokollok

A XX. század végétől az orvostudomány és a hozzá kapcsolódó tudományterületek tudásanyaga robbanásszerű fejlődést mutat. A gyakorló orvosok egyre nehezebb döntési helyzetekbe kerülnek a rengeteg rendelkezésre álló információ miatt. A hagyományos, konszenzus-, vagy véleményalapú medicina (*Consensus Based Medicine, CBM*) mellett megerősödött az úgynevezett tényeken, bizonyítékokon alapuló orvoslás (*Evidence Based Medicine, EBM*).

A tényeken alapuló orvoslás a legújabb eredmények tudatos és kritikus alkalmazása egy konkrét klinikai helyzetben. A fő feladat az orvoslás rendelkezésére álló hatalmas tudásanyagban megtalálni azokat a bizonyítottan helyes módszereket és eljárásokat, amelyek segítségével felállítható a pontos diagnózis, megtervezhető és végigvihető a megfelelő terápia -, a lehető legnagyobb egészségjavulás érdekében. Természetesen a döntések során a legkisebb kockázatú eljárásokat, és a költségeket is figyelembe kell venni. [151][51]

Mindezek gyakorlati megvalósulásának egyik formája az orvosi irányelvek és protokollok kidolgozása. Magyarországon a 23/2006. (V.18.) Egészségügyi Miniszteri rendelet írja elő ezen irányelvek kidolgozását. [87] A Magyar Közlönyben 2006-tól sorban jelennek meg ezek az irányelvek. Más országokban is hasonló tendenciák figyelhetők meg. Például az Egyesült Királyságban 1999 óta foglalkoznak orvosi irányelvek és protokollok kidolgozásával, Finnországban 2008-ra már majdnem 1000 szakmai irányelvet dolgoztak ki. [129]

Az orvosi irányelvek és protokollok támogatják azt az irányt, hogy az orvosi gyakorlat standardizálttá váljon, ugyanakkor az egyénre szabott gyógyítási formák is erősen jelenjenek meg az ellátásban. Növelhető az ellátás biztonsága, minősége, ugyanakkor a költséghatékonyság is szerepet játszik.

A támogatók mellett megjelennek a EBM támadói, bírálói is. Sokan *,,szakácskönyv medicinának"* tekintik az irányzatot, amely elgépiesíti az orvoslást, az egyéni motivációkat elsorvasztja. [135]

Informatikai szempontból az említett robbanásszerű fejlődés hatalmas mennyiségű tudásanyag kezelését jelenti. Ez nyilvánvalóan csak modern és korszerű informatikai módszerek és eszközök bevonásával oldható meg. Ennek során az irányelvek és protokollok formalizálása elkerülhetetlenné és mindenképpen szükségessé válik. A tudásanyagot megfelelő formában kell tárolni. A formális leírás viszont, mint láttuk a kriptográfiai protokollok esetében is, a további logikai elemzés alapjául szolgálhat. Lehetővé válik a validitás és a verifikálás. [102]

Az orvosi protokollok vizsgálata során alkalmazott formális nyelvek, például: Asbru, EON, GLIF, GUIDE, PRODIGY, PROforma, stb. [126]

A témakör fontosságát tükrözi az is, hogy az Európai Unió az Információs Társadalom és Technológia program 5. keretprogramjában indított egy orvosi protokollok formális eszközökkel történő fejlesztését célzó projektet, ami a Protocure (*Improving medical protocols by formal methods*) nevet kapta. A projekt az Asbru nyelvet választotta a protokollok formalizálásának alapjául. [103] Két orvosi irányelvet formalizáltak a projekt keretében. Az egyik az újszülöttkori sárgaság kezelésére vonatkozó irányelv, az Amerikai Gyermekgyógyászati Akadémia által kidolgozott irányelv. A másik a II-es típusú diabetes kezelésére kidolgozott irányelv -; Általános Orvosok Holland Tudományos Testülete irányelve.

A projekt tapasztalatai szerint egy-egy irányelv formalizálása igen időigényes feladat. A formális leírások igen összetetté váltak. [101]

A Protocure projekt a 6. keretprogramban folytatódott - Protocure II projekt (2002-2006). A legfontosabb projektcél az orvosi protokollok formális ellenőrzési módszertanának kidolgozása. A szemantikus ellenőrzés alapjául a KIV (Karlsruhe Interactive Verifier - KIV) nyelvet választották, amelyet összekapcsoltak az Asbru nyelvvel. További feladat az irányelvek naprakészen tartása, a változtatások átvezetése a formális leírásba.

6.3. Gazdasági-, biztosítási protokollok

Az orvosi protokollok esetén az egyik fő hajtóerő a protokollok kidolgozása és fejlesztése során a költséghatékonyság növelése. A magyar törvénykezés finanszírozási eljárásrend kidolgozását írta elő az egyes főbb betegségcsoportokra vonatkoztatva (Magyar Közlöny 47/2006. (XII. 27.) [86]). Ez a rendelet előírja, hogy ki kell dolgozni a kötelező egészségbiztosítás keretében nyújtható egészségügyi szolgáltatás leírását, az igénybevétel rendjét, illetve a szükséges beavatkozás, eljárás, gyógyszer (hatóanyag), gyógyászati segédeszköz megnevezését és mennyiségét.

A bankok, biztosítótársaságok hasonló szabályrendszerekkel dolgoznak az egyéni és a vállalati hitelfelvételek elbírálására. A 2008. évben kezdődött gazdasági válság egyik velejárója ezeknek a szabályrendszereknek a felborulása, szigorúbbá válása.

Amennyiben a protokoll-elméletek előbbi tendenciái itt is érvényesülnek, akkor a formalizálás, a kötött leírás után felmerül a továbblépés igénye és lehetősége: validitási és verifikálási vizsgálatok szükségessége.

6.4. Általános protokoll-elmélet

Az előző részekben példákat láthattunk protokoll-alapú szemléletmódra igen különböző területeken. Erre alapozva elmondhatjuk, hogy a folyamatok és tevékenységek ilyen módon történő felfogása egyre nagyobb súllyal jelenik meg a kutatás említett területein. Az eltérő tudományágakban megjelenő hasonló megoldások előbb-utóbb egymásra találnak, megteremtődnek azok a kapcsok, amelyek tapasztalatokat közvetítenek. Az egyes területeken megszerzett tapasztalatok egymás között cserélődve segíthetik a további fejlődést. Ennek hátterében az áll, hogy a matematikai segédeszközök alkalmazása minden esetben egzaktabbá teszi az adott tudományterültet, ugyanakkor megteremti az említett kapcsolódási pontokat is.

$\cdots\bullet\cdots$

A bennünket körülvevő természet vizsgálata során az emberek három alapelemet különítettek el. Az egyik a kézzel fogható anyag, a másik a változásokat okozó energia. A huszadik század végére mindezek mellé felsorakozott harmadiknak az információ. Ez egy olyan vonása a természetnek, amely mindenütt megtalálható, mindenhol nyomára bukkanunk, de egységes megközelítésével még adósak vagyunk. Információval találkozunk a biológia, az élet vizsgálata során (genetikai információ, sejtek közötti kommunikáció, állatok és növények közötti kommunikáció). Az emberi társadalom működése elképzelhetetlen információátadás, információcsere nélkül (beszéd, írás). Mai technológiai eszközeink egy része szintén olyan megoldásokkal vesz körül bennünket, amelyek vagy az információcserét biztosítják (számítógépes hálózatok, Internet), vagy már a létrejöttük, létezésük is az információcserére épül (virtuális pénz, elektronikus bankszámla).

Az információcsere során a Shannon-féle modellre hivatkozva [134] legalább két szereplőt kell megjelölnünk. Az egyik az információt kibocsátó fél, a másik pedig az azt fogadó partner. Ahhoz, hogy a két fél között megvalósulhasson az információcsere, meg kell egyezniük sok-sok apró részletben (kódolás, csatorna, stb.). Ugyanúgy, mint egy-egy protokoll esetén.

Az anyag és az energia átvitele, transzportja már mindennapi életünk része, tudásunkkal le tudjuk írni ezeket a folyamatokat. Az információ átvitele, transzportja is leírható, de az elkülönülő tudományterületek elméletei még nem ötvöződtek egységes felfogássá. [62] Egy olyan általános protokollelmélet, amely igen tág keretek között képes az igen eltérő sajátosságú és tartalmú protokollok közös vonásait feltárni, a különböző területek tapasztalatait egységes elméletbe foglalni, még nem született meg, bár ez talán megoldhatná az elméletek egységbe fogását.

7. fejezet

Mellékletek

7.1. A BAN-logika

A BAN-logika nyelve

A BAN-logika nyelve a következő jelöléseket használja:[1]

- A, B, S konkrét szereplők (CSN: Σ, Ψ),
- K_{ab}, K_{as}, K_{bs} közös kulcsok - szimmetrikus titkosítás (CSN: $ks_{(\Sigma,\Psi)}$),
- K_a, K_b, K_s nyilvános kulcsok - aszimmetrikus titkosítás (CSN: k_Σ),
- K_a^{-1}, K_b^{-1}, K_s^{-1} titkos kulcsok - aszimmetrikus titkosítás (CSN: k_Σ^{-1}),
- N_a, N_b, N_c konkrét állítások, kijelentések (CSN: $\Phi, \ldots, p, q$),
- P, Q, R szereplők,
- X, Y állítás, kijelentés,
- K titkosító kulcsok

[1] A zárójelben szereplő részek a CSN-logika megfelelő elemei.

A BAN-logika a következő alapelemeket használja:

- $P \models X$ - P elhiszi az X állítást. P az aktuális ismeretei alapján igaznak fogadja el az X állítást, a továbbiakban ennek megfelelő lépéseket tesz a protokoll végrehajtása során. (CSN: $B_{\Sigma,t}\Phi$)
- $P \lhd X$ - P látja X-et. P hozzáfér az X üzenethez, vagy annak egy részéhez. Használt üzenet fogadásának jelölésére. (CSN: $\sigma_{i,t}(x, y)$)
- $P \mid\sim X$ - az X üzenet(rész) P-től származik. P valamikor (nem garantált X frissessége) küldött egy X üzenetet, vagy üzenetrészt. (CSN: $S(\Sigma, t, x)$)
- $P \Rightarrow X$ - P kompetens az X állítás igazságának eldöntésében. (CSN: $K_{\Sigma,t}\Phi$)
- $\sharp(X)$ - az X üzenet friss. A BAN-logika időfelosztása (jelen és múlt) szerint az X üzenet a protokoll-futás kezdete után keletkezett. (CSN: $ss_{(\Sigma,\Psi)}$)
- $P \overset{K}{\leftrightarrow} Q$ - K egy titkos szimmetrikus kulcs P és Q között. P és Q szereplőkön kívül más nem (kivéve megbízható harmadik fél, például a kulcsot generáló szerver) ismeri a K kulcsot. (CSN: $ks_{(\Sigma,\Psi)}$)
- $\overset{K}{\mapsto} P$ - P nyilvános kulcsa K. (CSN: k_Σ)
- $P \overset{X}{\rightleftharpoons} Q$ - az X állítást csak P és Q (vagy az általuk megbízhatónak tekintett szereplők) ismerik. Csak P és Q használhatja X-et, hogy bizonyítsák azonosságukat a másik felé. (CSN: $ss_{(\Sigma,\Psi)}$)
- $\{X\}_K$ - az X állítás a K kulcs felhasználásával van titkosítva. (CSN: $E(x, ks_{(\Sigma,\Psi)})$,$e(x, k_\Sigma)$)
- $\langle X \rangle_Y$ - az X állítás kombinálva van az Y állítással. (CSN: $C(x, y)$)

A BAN-logika következtetési szabályai

Az egyes következtetési szabályok és jelentésük a következő:

$$(1)\quad \frac{P \models Q \overset{K}{\leftrightarrow} P, P \lhd \{X\}_K}{P \models Q \mid\sim X} \qquad (2)\quad \frac{P \models \overset{K}{\mapsto} Q, P \lhd \{X\}_K^{-1}}{P \models Q \mid\sim X}$$

(1) Amennyiben P kap egy X üzenetet, amely a K kulccsal van titkosítva, és P megbízik abban, hogy a K kulcsot rajta kívül csak Q ismeri, akkor P megbízhat abban, hogy az X üzenetet Q küldte. (CSN: A6(a), A11(a))

(2) Amennyiben P kap egy digitálisan aláírt X üzenetet, és P megbízik abban, hogy az aláírás ellenőrzéséhez Q nyilvános kulcsát használta, akkor P megbízhat abban, hogy az X üzenet Q-tól származik. (CSN: A6(a), A8(a), A9(a))

$$(3)\quad \frac{P \models Q \overset{Y}{\rightleftharpoons} P, P \lhd \langle X \rangle_Y}{P \models Q \mid\sim X} \qquad (4)\quad \frac{P \models \sharp(X), P \models Q \mid\sim X}{P \models Q \models X}$$

(3) Amennyiben P megbízik abban, hogy az Y állítást csak P és Q ismeri, valamint P kap egy Y és X kombinációját tartalmazó üzenetet, akkor P megbízik abban, hogy az X üzenetet Q küldte. (CSN: A15(a))

(4) Amennyiben P megbízik abban, hogy az állítás (üzenet) friss, és megbízik abban is, hogy az üzenetet Q küldte, akkor P elfogadja azt, hogy Q az állítást igaznak tartja. Ez a formula azt fejezi ki, hogy a partnerek csak olyan üzeneteket küldenek egymásnak, amelyeknek az igazságtartalmában megbíznak - becsületesek a résztvevők.

$$(5)\quad \frac{P \models Q \Rightarrow X, P \models Q \models X}{P \models X} \qquad (6)\quad \frac{P \models X, P \models Y}{P \models (X, Y)}$$

(5) Amennyiben P megbízik abban, hogy Q kompetens az X állítás igazságtartalmának eldöntésében, valamint P abban is megbízik, hogy Q igaznak tartja az X állítást, akkor P megbízik X állítás igazságtartalmában.

(6) Amennyiben P megbízik X és Y igazságtartalmában, akkor megbízik az (X, Y) állítás igazságtartalmában is.

$$(7)\quad \frac{P \models (X,Y)}{P \models X} \qquad (8)\quad \frac{P \models Q \models (X,Y)}{P \models Q \models X}$$

(7) Amennyiben P megbízik (X,Y) állítás igazságtartalmában, akkor megbízik külön X igazságtartalmában is, - ami igaz Y-ra is.

(8) Amennyiben P megbízik abban, hogy Q igaznak fogadja el (X,Y) igazságtartalmát, akkor P abban is megbízik, hogy Q igaznak fogadja el X igazságtartalmát.

$$(9)\quad \frac{P \models Q \mid\sim (X,Y)}{P \models Q \mid\sim X} \qquad (10)\quad \frac{P \lhd (X,Y)}{P \lhd X}$$

(9) Amennyiben P megbízik abban, hogy (X,Y) üzenet Q-tól származik, akkor P megbízik abban is, hogy X is Q-tól származik, - ami igaz Y-ra is.

(10) Amennyiben P látja, hozzáfér (X,Y)-hoz, akkor hozzáfér X-hez külön is, - ami igaz Y-ra is. (CSN: A4(a))

$$(11)\quad \frac{P \lhd \langle X \rangle_Y}{P \lhd X} \qquad (12)\quad \frac{P \models Q \overset{K}{\leftrightarrow} P, P \lhd \{X\}_K}{P \lhd X}$$

(11) Amennyiben P hozzáfér X és Y üzenetek kombinációjához (például konkatenációjukhoz), akkor P hozzáfér X-hez külön is, - ami igaz Y-ra is. (CSN: A4(a))

(12) Amennyiben P megbízik abban, hogy K egy közös titkos kulcs Q-val, valamint P kap egy K-val titkosított üzenetet, akkor P látja X-et, - P vissza tudja fejteni a megfelelő kulcs segítségével a titkosított üzeneteket. (CSN: A8(a))

$$(13)\quad \frac{P \models \overset{K}{\mapsto} P, P \lhd \{X\}_K}{P \lhd X} \qquad (14)\quad \frac{P \models \overset{K}{\mapsto} Q, P \lhd \{X\}_K^{-1}}{P \lhd X}$$

(13) Amennyiben P megbízik abban, hogy saját, nyilvános kulcsa K, és P kap egy üzenetet a nyilvános kulcsával titkosítva, akkor P hozzáfér X-hez, - P képes visszafejteni a nyilvános kulcsával titkosított üzeneteket (nyilván a saját tikos kulcsa segítségével). (CSN: A9(a))

(14) Amennyiben P megbízik abban, hogy a K kulcs a Q szereplő nyilvános kulcsa, és P kap egy olyan X üzenetet, ami a K nyilvános kulcs titkos párjával van titkosítva, akkor P hozzáfér X-hez, - P képes ellenőrizni Q aláírását. (CSN: A10(a))

$$(15)\quad \frac{P \models \sharp(X)}{P \models \sharp(X,Y)} \qquad (16)\quad \frac{P \models R \overset{K}{\leftrightarrow} R'}{P \models R' \overset{K}{\leftrightarrow} R} \qquad (17)\quad \frac{P \models Q \models R \overset{K}{\leftrightarrow} R'}{P \models Q \models R' \overset{K}{\leftrightarrow} R}$$

(15) Amennyiben P megbízik abban, hogy X állítás friss, akkor elfogadja azt is, hogy az (X, Y) állítás is friss.

(16) Amennyiben P megbízik abban, hogy K titkos kulcs R és R' között, akkor, azt mindketten ismerik.

(17) Amennyiben P megbízik abban, hogy Q elfogadja, hogy K titkos kulcs R és R' között, akkor P megbízik abban is, hogy Q elfogadja, hogy a titkos kulcsot mindketten (R és R') ismerik.

$$(18)\quad \frac{P \models R \overset{X}{\rightleftharpoons} R'}{P \models R' \overset{X}{\rightleftharpoons} R} \qquad (19)\quad \frac{P \models Q \models R \overset{X}{\rightleftharpoons} R'}{P \models Q \models R' \overset{X}{\rightleftharpoons} R}$$

(18) A (16)-os szabály állításokra vonatkozó formája.

(19) A (17)-es szabály állításokra vonatkozó formája.

7.2. A Kudo-Mathuria protokoll HLPSL kódjai és eredménylisták

A leírásánál a sorok bal oldalán szereplő számok nem tartoznak a forráskódhoz, a sorok azonosítását, a sorokra történő hivatkozást segítik.

7.2.1. A K-M-P1 protokoll HLPSL kódja

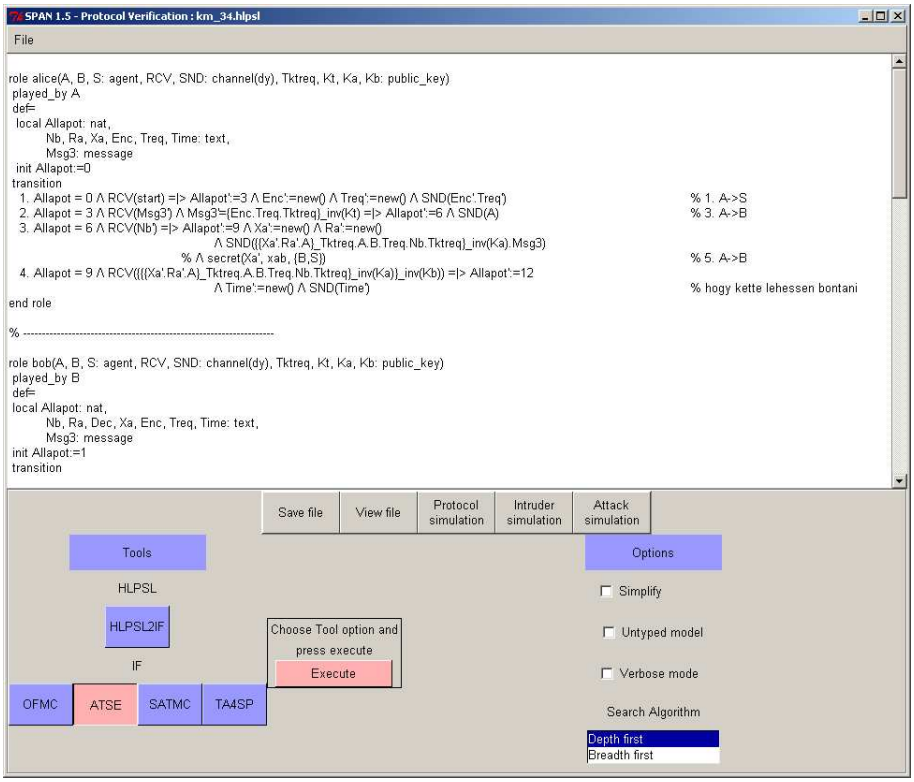
```
001 role alice(A, B, S: agent, RCV, SND: channel(dy), Tktreq, Kt, Ka,
002             Kb: public_key)
003   played_by A
004   def=
005   local Allapot: nat,
006         Nb, Ra, Xa, Enc, Treq, Time: text,
007         Msg3: message
008   init Allapot:=0
009   transition
010   1. Allapot = 0 /\ RCV(start) =| > Allapot':=3 /\ Enc':=new() /\
011         Treq':=new() /\ SND(Enc'.Treq')
012   2. Allapot = 3 /\ RCV(Msg3') /\ Msg3'={Enc.Treq.Tktreq}_inv(Kt)
013         =| > Allapot':=6 /\ SND(A)
014   3. Allapot = 6 /\ RCV(Nb') =| > Allapot':=9 /\ Xa':=new() /\
015         Ra':=new() /\ SND({{Xa'.Ra'.A}_Tktreq.A.B.Treq.
016         Nb.Tktreq}_inv(Ka).Msg3)
017         % /\ secret(Xa', xab, {B,S})
018   4. Allapot = 9 /\ RCV({{{Xa'.Ra'.A}_Tktreq.A.B.Treq.
019         Nb.Tktreq}_inv(Ka)}_inv(Kb)) =| > Allapot':=12 /\
020         Time':=new() /\ SND(Time')
021 end role
022 % ----------------------------------------------------------------
023 role bob(A, B, S: agent, RCV, SND: channel(dy), Tktreq, Kt, Ka,
024             Kb: public_key)
025   played_by B
026   def=
```

```
027   local Allapot: nat,
028         Nb, Ra, Dec, Xa, Enc, Treq, Time: text,
029         Msg3: message
030   init Allapot:=1
031   transition
032   1. Allapot = 1 /\ RCV(A) =| > Allapot':=4 /\ Nb':=new() /\
033         SND(Nb')
034   2. Allapot = 4 /\ RCV(Xa'.Ra'.A_Tktreq.A.B.Treq'.Nb.Tktreq_inv(Ka).
035         Msg3') =| > Allapot':= 7 /\
036         SND(Xa'.Ra'.A_Tktreq.A.B.Treq.Nb.Tktreq_inv(Ka)_inv(Kb))
037   3. Allapot =7 /\ RCV(Time') =| > Allapot':=10 /\ Dec':=new() /\
038         SND(Dec'.Treq)
039   4. Allapot = 10 /\ RCV(Dec.Treq.inv(Tktreq)) =| > Allapot':=14
040 end role
041 % ------------------------------------------------------------------
042 role server(A, B, S: agent, RCV, SND: channel(dy), Tktreq, Kt, Ka,
043             Kb: public_key)
044   played_by S
045   def=
046   local Allapot: nat,
047         Enc, Dec, Treq: text
048   init Allapot:=2
049   transition
050   1. Allapot = 2 /\ RCV(Enc'.Treq') =| > Allapot':=5
051         /\ SND(Enc.Treq.Tktreq_inv(Kt))
052   2. Allapot = 5 /\ RCV(Dec'.Treq) =| > Allapot' := 8
053         /\ SND(Dec.Treq.inv(Tktreq))
054         /\ secret(inv(Tktreq), itktreq, {S,B})
055 end role
056 % ------------------------------------------------------------------
057 role session(A, B, S: agent, Tktreq, Kt, Ka, Kb: public_key)
058   def=
058   local SNDA, RCVA, SNDB, RCVB, SNDS, RCVS : channel(dy)
059   composition
```

```
060         alice(A, B, S, SNDA, RCVA, Tktreq, Kt, Ka, Kb) /\
061         bob(A, B, S, SNDB, RCVB, Tktreq, Kt, Ka, Kb) /\
062         server(A, B, S, SNDS, RCVS, Tktreq, Kt, Ka, Kb)
063 end role
064 % ----------------------------------------------------------------
065 role envinronment()
066   def=
067   const
068         a, b, s: agent,
069         tktreq, kt, ka, kb: public_key,
070         xab, itktreq: protocol_id
071   intruder_knowledge=a, b, s, i, tktreq, kt, ka, kb
072   composition
073         session(a,b,s, tktreq, kt, ka, kb)
074 end role
075 % ----------------------------------------------------------------
076 goal
077   % secrecy_of xab
078   secrecy_of itktreq
079 end goal
080 % ----------------------------------------------------------------
081 envinronment()
```

7.2.2. K-M-P1 protokoll - OMFC elemzés eredménye

```
001 % OFMC
002 % Version of 2006/02/19
003 SUMMARY
004     UNSAFE
005 DETAILS
006     ATTACK_FOUND
007 PROTOCOL
008     C:\SPAN\testsuite\results\km_34.if
```

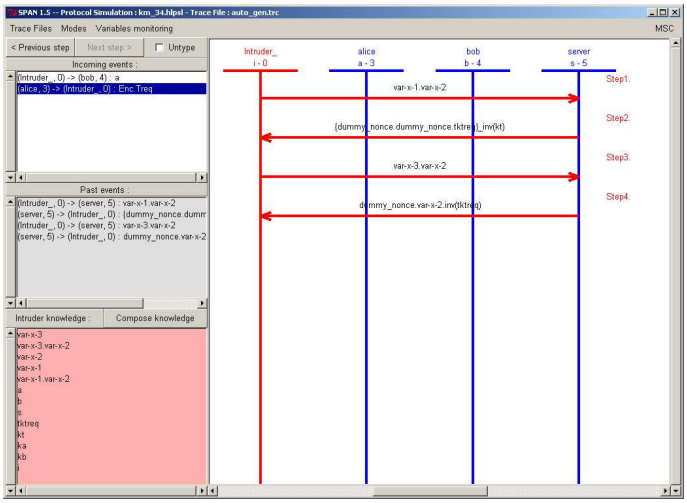


7.2.1. ábra A K-M-P1 protokoll az AVISPA rendszerben.

```
009 GOAL
010     secrecy_of_itktreq
011 BACKEND
012     OFMC
013 COMMENTS
014 STATISTICS
015     parseTime: 0.00s
016     searchTime: 0.06s
017     visitedNodes: 3 nodes
018     depth: 1 plies
019 ATTACK TRACE
020 i -> (s,3): x234.x235
021 (s,3) -> i: dummy_nonce.dummy_nonce.tktreq_inv(kt)
022 i -> (s,3): x248.x235
023 (s,3) -> i: dummy_nonce.x235.inv(tktreq)
024 i -> (i,17): inv(tktreq)
025 i -> (i,17): inv(tktreq)
026 % Reached State:
027 % secret(inv(tktreq),itktreq,set_112)
028 % contains(s,set_112)
029 % contains(b,set_112)
030 %state_server(s,a,b,tktreq,kt,ka,kb,8,x234,x248,x235,set_112,3)
031 %state_bob(b,a,s,tktreq,kt,ka,kb,1,dummy_nonce,dummy_nonce,
032     dummy_nonce,dummy_nonce,dummy_nonce,dummy_nonce,
033     dummy_nonce,dummy_msg,3)
034 %state_alice(a,b,s,tktreq,kt,ka,kb,0,dummy_nonce,dummy_nonce,
035     dummy_nonce,dummy_nonce,dummy_nonce,
036     dummy_nonce,dummy_msg,3)
```

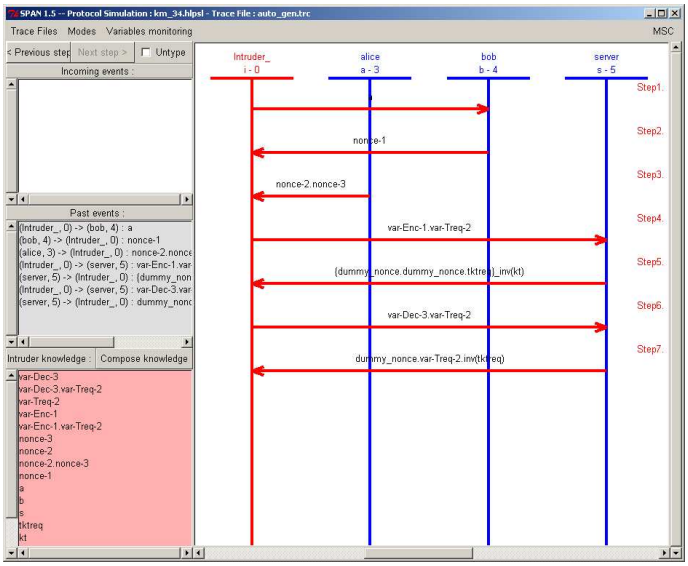
7.2.3. K-M-P1 protokoll - ATSE elemzés eredménye

```
001 % ----------------------------------------------------------------------
002 SUMMARY
```



7.2.2. ábra A K-P-P1 protokoll OMFC elemzés szerinti támadás első lépései.

003 UNSAFE
004 DETAILS
005 ATTACK_FOUND
006 TYPED_MODEL
007 PROTOCOL
008 C:\SPAN\testsuite\results\km_34.if
009 GOAL
010 Secrecy attack on (inv(tktreq))
011 BACKEND
012 CL-AtSe
013 STATISTICS
014 Analysed : 1 states
015 Reachable : 1 states
016 Translation: 0.02 seconds
017 Computation: 0.00 seconds



7.2.3. ábra A K-M-P1 protokoll ATSE elemzés szerinti támadása.

018 ATTACK TRACE
019 i − > (b,4): a
020 (b,4) − > i: n9(Nb)
021 i − > (a,3): start
022 (a,3) − > i: n1(Enc).n1(Treq)
023 i − > (s,5): Enc(17).Treq(17)
024 (s,5) − > i: dummy_nonce.dummy_nonce.tktreq_(inv(kt))
025 i − > (s,5): Dec(18).Treq(17)
026 (s,5) − > i: dummy_nonce.Treq(17).inv(tktreq)
027 & Secret(inv(tktreq),set_112); Add s to set_112;
028 & Add b to set_112;

7.2.4. K-M-P2 protokoll - OMFC elemzés eredménye

001 % OFMC

```
002 % Version of 2006/02/19
003 SUMMARY
004     SAFE
005 DETAILS
006     BOUNDED_NUMBER_OF_SESSIONS
007 PROTOCOL
008     C:\SPAN\testsuite\results\km_41_kmp2.if
009 GOAL
010     as_specified
011 BACKEND
012     OFMC
013 COMMENTS
014 STATISTICS
015     parseTime: 0.00s
016     searchTime: 0.09s
017     visitedNodes: 12 nodes
018     depth: 4 plies
```

7.2.5. K-M-P2 protokoll - ATSE elemzés eredménye

```
001 SUMMARY
002     SAFE
003 DETAILS
004     BOUNDED_NUMBER_OF_SESSIONS
005     TYPED_MODEL
006 PROTOCOL
007     C:\SPAN\testsuite\results\km_41_kmp2.if
008 GOAL
009     As Specified
010 BACKEND
011     CL-AtSe
012 STATISTICS
013     Analysed : 1 states
```

014 Reachable : 1 states
015 Translation: 0.02 seconds
016 Computation: 0.00 seconds

7.2.6. K-M-P3 protokoll - OMFC elemzés eredménye

001 % OFMC
002 % Version of 2006/02/19
003 SUMMARY
004 SAFE
005 DETAILS
006 BOUNDED_NUMBER_OF_SESSIONS
007 PROTOCOL
008 C:\SPAN\testsuite\results\km_40_kmp3.if
009 GOAL
010 as_specified
011 BACKEND
012 OFMC
013 COMMENTS
014 STATISTICS
015 parseTime: 0.01s
016 searchTime: 0.08s
017 visitedNodes: 12 nodes
018 depth: 4 plies

7.2.7. K-M-P3 protokoll - ATSE elemzés eredménye

001 SUMMARY
002 SAFE
003 DETAILS
004 BOUNDED_NUMBER_OF_SESSIONS
005 TYPED_MODEL

```
006 PROTOCOL
007     C:\SPAN\testsuite\results\km_40_kmp3.if
008 GOAL
009     As Specified
010 BACKEND
011     CL-AtSe
012 STATISTICS
013     Analysed : 2 states
014     Reachable : 2 states
015     Translation: 0.03 seconds
016     Computation: 0.00 seconds
```

8. fejezet

Összefoglalás

Értekezésünk tárgya a kriptográfiai protokollok formális alapú vizsgálata. Az első fejezet áttekinti a témát, bemutatja a kutatási munka irányát.

A második fejezet fő célja a kriptográfia alapvető fogalmainak ismertetése és tisztázása. Ez a rész azokat a meghatározó elemeket hangsúlyozza, amelyek a kriptográfiai protokollok elemzéséhez elengedhetetlenül szükségesek.

Ki kell emelnünk a fejezet jelölési módokra vonatkozó részét. A későbbiekben több jelölési rendszert is alkalmazunk. Ennek egyik oka a szakirodalomban kialakult hagyományok követése. A másik ok az alkalmazott logikai rendszer (CSN-logika) viszonylag összetett leírási módja. A hagyományos jelölést a fogalmak tisztázása során használtuk. Az összetettebb jelölési rendszert az általunk elvégzett logikai vizsgálatok során alkalmaztuk.

Hosszabb részt foglal el a fejezetben az alapvető protokollok bemutatása. Ennek oka a protokollok sokrétűségének hangsúlyozása. Az áttekintés bemutatja, hogy az egyes protokollok egymásra épülnek. Az egyik protokoll hiányosságait egy következő javítja. Jellemző a protokollokra az is, hogy igen apró eltérések is zavarokat, támadhatóságot okozhatnak. Ebben a fejezetben bemutatásra kerültek különböző támadási módok (lehallgatás, beékelődő támadás, szótáralapú támadás, stb.).

Mindezek a 4. és 5. fejezet tematikáját és eredményeit készítik elő.

A dolgozat harmadik részének célja a kriptográfiai protokollok vizsgálati eszközeinek bemutatása. Ennek során két nagy terület különíthető el. Az egyik a számításelméleti megközelítés, a másik pedig a formális vizsgálat. A kétféle nézőpont napjainkban összefonódni látszik. Ez nyilván a vizsgálati módszerek közös céljából eredeztethető: megbízható, biztonságos, a kitűzött céloknak megfelelő protokollok megalkotása.

A formális módszerek bemutatása során kétféle megközelítést alkalmaztunk. A másodikként szereplő osztályozás napjaink felfogását tükrözi.

Vizsgálódásaink eredményeként megfogalmazhatjuk azt az állítást, amely szerint az 1990-es évek végéig végzett kutatások elkülöníthetők a későbbiektől. Tekinthetjük ezeket az időszakokat I. és II. generációs vizsgálati szakaszoknak. További alapvető változást jelent a protokollok fejlődésében a vezeték nélküli kommunikáció általánossá válása. Ez új eszközrendszert, új protokollokat és új vizsgálati modelleket jelent. Itt kellett szóvá tennünk és vizsgálnunk a protokollok összekapcsolását, a többszereplős és sok esetben nyílt végű protokollokat. Az, hogy az így elkülönített szakaszok tényleges fejlődési fázisokat jelentenek csak hosszabb időtávlatban igazolható. A megközelítés egy új szemléletét jelentheti a tudományterületnek, de a végső törvényszerűségek kibontása további vizsgálatokat igényel.

A 3.2.2. fejezetben a formális vizsgálatok első jelentősnek tekintett rendszerét, a BAN-logikát mutattuk be részletesen. A 7.1. fejezet részletesen is tartalmazza a BAN-logika leírását. Ennek oka a 3.2.3. fejezetben leírt CSN-logika összevethetősége a BAN-logikával. A CSN-logika első leírása 1997-ben jelent meg, majd 2003-ban tettek közzé egy jelentős bővítést a logika alkotói (T. Coffey, P. Saidha és T. Newe).

A negyedik és ötödik fejezet tartalmazza az általunk elvégzett és közleményekben publikált kutatások összefoglalását.

A negyedik fejezetben kerül bemutatásra az időfeloldó (*time-release*) problémakör története és a K-M-P1 protokoll, amelyet M. Kudo és A. Mathuria dolgozott ki, és elemzett a CSN-logikával 1999-ben. Munkánkban

ezt a protokollt vizsgáltuk tovább. A 4.3. fejezet bemutatja, hogy a protokoll futása során a passzív támadó képes lehallgatni a résztvevők közötti kommunikációt. A G4. tétel szerint: *Az E támadó - aki lehallgatja az üzeneteket - ugyanolyan információkkal rendelkezik, mint B a t_9 időpontban, a t_8 időpont után. Vagyis E szintén képes visszafejteni az időbizalmas üzenetet.*

Szintén bizonyítható, hogy az abszolút megbízható fél képes visszafejteni az időbizalmas adatokat a protokoll lefutása előtt (G5. tétel). Ezek nem az eredeti protokoll hibái, az nem köti ki az ilyen irányú védettséget. Kutatásaink során megvizsgáltuk a kommunikáció ilyen irányú védelmének lehetőségeit. A kidolgozott K-M-P2 és K-M-P3 protokollokról a G6. és G7. tételekben igazoltuk, hogy már megfelelnek az általunk kitűzött céloknak.

Ezt követően a 4.4. fejezetben az aktív támadás lehetőségeit vizsgáltuk a K-M-P1, K-M-P2 és K-M-P3 protokollokban. A vizsgálatot az AVISPA rendszer segítségével végeztük el. A protokollok leírását és a futási eredményeket a 7.2. fejezet tartalmazza. Összefoglalva elmondhatjuk, hogy a K-M-P1 protokoll támadható, a módosított K-M-P2 és K-M-P3 protokollok esetén viszont már nem mutathatók ki hasonló támadások. Ezek az eredmények a [145][147] közleményekben jelentek meg.

Az ötödik fejezetben tovább bővítettük a CSN-logika alkalmazási körét. A vezeték nélküli kommunikációs megoldások fejlődésével előtérbe kerültek a többcsatornás protokollok. Fő eredményünk az, hogy a CSN-logikát sikerült úgy bővítenünk, hogy az alkalmassá vált a többcsatornás protokollok formális vizsgálatára. Ez a bővítés az 5.3. fejezetben került bemutatásra. Az alkalmazhatóság igazolására a MANA protokollcsalád három protokollját elemeztük. A MANA I protokoll esetén sikerült igazolni a protokoll-célokat. A MANA II és MANA III protokollok esetén viszont olyan pontokat sikerült feltárnunk, amelyekkel sikerülhet megzavarni a protokoll működését. Az elemzés végén javaslatot tettünk a protokollok erősítésére. Ezek az eredmények a [146][148][150][149] közleményekben jelentek meg.

A dolgozat hatodik fejezete a protokollok vizsgálatának további le-

hetőségeit mérlegeli. A gondolatsor legfőbb eleme a protokollok egyre általánosabb használata, a protokollszerű megközelítési mód *„terjedése”*. Véleményünk szerint hasonló fejlődési szakaszok mutathatók ki a számítógépes hálózatok, a kriptográfiai protokollok és az orvos-szakmai irányelvek területén. A gazdasági- és biztosítási döntéshozatalnál eljárásrendeket, döntési-fákat igyekeznek kidolgozni. Kirajzolódni látszik egy általánosabb protokoll-elméleti megközelítés. Ezeket a tendenciákat, a belőlük levont sejtéseket nem állt módunkban igazolni, további vizsgálatok szükségesek igazolásukhoz, csupán gondolatébresztő szándékkal közöltük őket.

9. fejezet

Summary

The theme of this dissertation is to examine cryptographic protocols based on formal methods. In chapter one we survey the direction of our research work.

The aim of the second chapter is to review and clear the basic notions of cryptographic protocols. This part of the dissertation emphasizes crucial elements which are necessary to analyze cryptographic protocols.

We have to highlight the notation part of the chapter. We apply more notations in this chapter. One of the reasons is to follow the traditions of this scientific area. The other reason is the complexity of the applied logical system (CSN-logic). We apply the classical notation to describe basic notations. The more complex notation is used to describe our logical examinations.

The description of the basic protocols is a longer part of the dissertation. The reason of this is to enhance the variety of protocols. The outline presents that the protocols are based on each other. The faults of one protocol are corrected by another. It is also typical of protocols that tiny variances may cause vulnerable points and disorders. Different attack methods (interception, Man in the Middle attack, dictionary attack, etc.) are presented in this chapter.

These prepare the program and results of chapter four and five.

In chapter three the aim is to introduce the examination tools of the cryptographic protocols. Two main parts can be divided. The first one is the computability theory and the second one is the formal examination. The two methods seem to interlock these days. The process of this connection dates back to the common objects and aims of the methods: to construct trusty, secure, adequate protocols.

We apply two methods in the course of the introduction of formal examination. The second classification reflects the approach of our days.

As a result we can state that researches done all the end of 1990s can be separated from later ones. We can consider these periods I and II generation examination periods. Further basic change in the evolution of protocols is that wireless communication has become general. This situation has created new tools, protocols and examination methods. Here we should mention the interlocking of protocols, the multi-user and open-ended protocols. The fact the separated periods are factual evolutionary phases can only be confirmed in longer time perspective. This approach may be a new concept in this area of science but finding the final regularities demands further researches.

In chapter 3.2.2. we present the BAN-logic as the first significant system of the formal examinations of cryptographic protocols. Chapter 7.1. contains the description of the BAN-logic in details. The reason is the contrastability of the BAN-logic and the CSN-logic in chapter 3.2.3. The first description of the CSN-logic was published in 1997 and the creators of the logic (T. Coffey, P. Saidha amd T. Newe) extended it in 2003.

Chapters four and five contain the summary of our researches which we published in scientific papers.

We present the history of the time-release problem and the K-M-P1 protocol in chapter four. This protocol was worked out and analyzed with the CSN-logic by M. Kudo and A. Mathuria in 1999. Henceforth we analyzed this protocol in our research. Chapter 4.3. presents that the passive

attacker is able to intercept the communication between partners. By theorem $G4$: Attacker E - who eavesdrops messages - has the same information as B at time t_9 after the milestone t_8. Namely, E is able to decrypt the time-confidential messages too.

It also can be proved that the absolute reliable partner is able to decrypt the time-release messages before the end of the protocol (theorem $G5$). These are not the mistakes of the original protocol because it does not specify this kind of secrecy. We examine this kind of direction of the protection of the communication in our research. The protocols K-M-P2 and K-M-P3 elaborated by us meet our original requirements as we proved it in the theorem G6 and G7.

Next we examine the possibilities of the active attack in protocols K-M-P1, K-M-P2 and K-M-P3. The examination is achieved with the AVISPA system. The description of the protocols and the results of running the AVISPA code are in chapter 7.2. To sum up, we state that the protocol K-M-P1 can be attacked but we cannot detect similar attacks in the modified protocols K-M-P2 and K-M-P3. These results were published in [145][147].

In chapter five we extend the application range of the CSN-logic further. Multi-channel protocols come to the front by the development of wireless communication solutions. Our main results is that we can extend the CSN-logic to be able to carry out formal examinations of multi-channel protocols. It is presented in chapter 5.3. We apply the extended logic to verify validity of three protocols in the MANA protocol family. We have established that protocol MANA I is correct. We have disclosed such attack points in the protocol MANA II and MANA III with which the process of the protocols can be disturbed. We suggest reinforcing and expanding the protocols at the end of the analyses. These results were published in [146][148][150][149].

We study additional possibilities of protocol examination in chapter six. The main component of the chain of ideas that a new 'protocol aspect thinking' become general. In our opinion similar development phases can be detect in the areas of computer networks, cryptographic protocols and medical guidelines. Similar processes and decision-trees are being worked

out in economic and insurance decision-making. A general protocol theory approach seems to outline. We cannot verify these tendencies and conjectures from them. We need more examinations to verify them, so our intension is only thought-provoking.

Irodalomjegyzék

[1] M. Abadi and A. Gordon. A calculus for cryptographic protocols: The Spi calculus. Technical report, Digital Equipment Corporation, System Research Center, January 1998.

[2] M. Abadi and R. Needham. Prudent engineering practice for cryptographic protocols. Research Report 125, Digital Equipment Corp., System Research Center, Junius 1994. Később megjelent: IEEE Transactions on Software Engineering, 1996, vol. 22, pp. 122–136.

[3] M. Abadi and P. Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). In *In Proceeding of the IFIP Conference on Theoretical Computer Science*, LNCS 1872. Springer, 2000.

[4] M. Abadi and M. Tuttle. A semantics for a logic of authentication. In *Proceedings of the Tenth Annual ACM Symposium on Principles of Distributed Computing*, pages 201–206, 1991.

[5] G. Ács. Biztonságos útvonalválasztás ad hoc hálózatokban. Master's thesis, Budapesti Műszaki és Gazdaságtudományi Egyetem, Villamosmérnöki és Informtikai Kar, Híradástechnikai Tanszék, Budapest, 2005. Konzulens Dr. Buttyán Levente.

[6] G. Adámy. *A Biometrikus fehérkönyv*. Login Autonom Kft., 2000.

[7] N. Agray, W. van der Hoek, and E. de Vink. On BAN logics for industrial security protocols. In B. Dunin-Keplicz and E. Nawarecki,

editors, *From Theory to Practice in Multi-Agent Systems, LNAI 2296*. 2002.

[8] J. Alves-Foss and T. Soule. A weakest precondition calculus for analysis of cryptographic protocols. In *Proceedings of the DIMACS Workshop on Design and Formal Verification of Security Protocols*, 1997.

[9] AVANTSSAR. Automated validation of trust and security of service-oriented architectures. http://www.avantssar.eu. Visited: 2009.02.23.

[10] Team AVISPA. Automated validation of internet security protocols and applications, http:// avispa-project.org. Visited: 2009.02.17.

[11] Team AVISPA. Deliverable D2.1: The high level protocol specification language. http://avispa-project.org/delivs/2.3/d2-1.pdf, January 2003.

[12] Team AVISPA. Deliverable D2.3: The intermediate format. http://avispa-project.org/delivs/2.3/d2-3.pdf, August 2003.

[13] Team AVISPA. AVISPA v1.1 User manual. http://avispa-project.org/package/user-manual.pdf, June 2006.

[14] Team AVISPA. HLPSL tutorial - A beginner's guide to modelling and analysing internet security protocols. http://avispa-project.org/package/tutorial.pdf, June 2006. Document Version 1.1.

[15] M. Barr. Programming embedded systems in C and C++, embedded system glossary. http://www.netrino.com/Embedded-Systems/Glossary, 1999. A megjelölt könyvre alapozott Internetes közlemény.

[16] T. Bartha, Gy. Csertán, Sz. Gyapay, I. Majzik, A. Pataricza, and D. Varró. *Formális módszerek az informatikában*. TypoTex Kiadó, Budapest, 2004.

[17] G. Bella. *Formal Correctness of Security Protocols*. Springer-Verlag, 2007.

[18] G. Bella and L. Paulson. Using Isabelle to prove properties of the Kerberos authentication systems. In *Proceedings of the DIMACS Workshop on Design and Formal Verification of Security Protocols*, 1997.

[19] M. Bellare and S. Goldwasser. Encapsulated key-escrow. In *4th ACM Conference on Computer and Communications Security*, 1996. Earlier version appeared as MIT Laboratory for Computer Science Technical Report 688, April 1996.

[20] M. Bellare and S. Goldwasser. Verifiablepartial key escrow. In *Proceedings of the Fourth Annual Conference on Computer and Communications Security, ACM*, 1997. Earlier version was Technical Report CS95-447, Department of Computer Science and Engineering, University of California at San Diego, October 1995.

[21] M. Bellare and P. Rogaway. Entity authentication and key distribution. In D. Stinson, editor, *Advances in Cryptology - Crypto'93*, volume 773 of *Lecture Notes in Computer Science*. Springer-Verlag, 1994.

[22] S. M. Bellovin and M. Merritt. Encrypted key exchange: Password-based protocols secure against dictionary attacks. In *IEEE Computer Society Symposium*, pages 72–84, 1992.

[23] S. M. Bellovin and M. Merritt. Augmented encrypted key exchange: A password-based protocol secure against dictionary attacks and password file compromise. In *ACM Conference on Computer and Communications Security*, pages 244–250, 1993.

[24] S. M. Bellovin and M. Merritt. An attack on the interlock protocol when used for authentication. *I.E.E.E. Transactions on Information Theory*, 40.(1):273–275, January 1994.

[25] A. Bérczes, J. Folláth, and A. Pethő. On a family of collision-free functions. *to appear*.

[26] A. Bérczes and I. Járási. An application of index forms in cryptography. *Periodica Mathematica Hungarica*, 58:35–45, 2008.

[27] A. Bérczes, J. Ködmön, and A. Pethő. A one-way function based on norm form equations. *Periodica Mathematica Hungarica*, 49:1–13, 2004.

[28] G. Biczók, K. Fodor, B. Kovács, and Á. Szabó. Pervasive computing - Rejtett számítástechnika. *Híradástechnika*, Március 2003.

[29] P. Bieber. A logic of communication in a hostile environment. In *Proceedings of the Computer Security Foundations Workshop III*, pages 14–22. IEEE Computer Society Press, June 1990.

[30] I. F. Blake and A. C.-F. Chan. Scalable, server-passive, user-anonymous timed release public key encryption from bilinear pairing. http://eprint.iacr.org/2004/211, September 2007.

[31] A. Bogdanov and L. Trevisan. *Average-Case Complexity*, volume 2 of *Foundation and Trends in Theoretical Computer Science*. now Publishers Inc., 2006.

[32] D. Bohen and M. Franklin. Identity based encryption from the weil pairing. In *Advances in Cryptology - Crypto'01*, LNCS 2139, pages 213–229. Springer-Verlag, 2001.

[33] D. Bohen and M. Naor. Timed commitments. In *Advances in Cryptology - Crypto'00*, LNCS 1880, pages 236–254. Springer-Verlag, 2000.

[34] H. Zs. Bojané, G. Borsodi, J. Miskolczi, K. Tarnay, and L. Zs. Varga. Kommunikációs protokollok formális leírása és konformancia tesztelése. In *Magyar Informatikusok I. Világtalálkozója, Konferencia és Kiállítás*, pages 467–472. Gábor Dénes Műszaki Informatikai Főiskola, Augusztus 1996.

[35] D. Boneh, X. Boyen, and E.-J. Goh. Hierarchical identity based encryption with constant size ciphertext.

http://eprint.iacr.org/2005/015, May 2005. An extended abstract of this paper appears in R. Cramer, editor, Advances in Cryptology - EUROCRYPT 2005, volume of Lecture Notes in Computer Science, pages 440-456, Sprinegr, 2005.

[36] C. Boyd and W. Mao. On a limitation of BAN logic. In *Advances in Cryptology - Eurocrypt'93*, pages 240–247. Springer-Verlag, 1994.

[37] S. H. Brackin. A HOL extension of GY for automatically analyzing cryptographic protocols. In *Proceedings of the Ninth IEEE Computer Security Foundations Workshop*, pages 62–76, 1996.

[38] J. A. Buchmann. *Introduction to Cryptography.* Springer-Verlag, 2001.

[39] M. Burrows, M. Abadi, and R. Needham. Authentication: A practical study in belief and action. In Moshe Y. Vardi, editor, *Proceedings of TARK II (Theoretical Aspects of Rationality and Knowledge)*, March 1988.

[40] M. Burrows, M. Abadi, and R. Needham. A logic of authentication. *ACM Transactins on Computer Systems*, 8(1):18–36, February 1990.

[41] R. W. Buttler. What is formal methods, why is formal methods necessary?, May 2004.

[42] L. Buttyán. Formal methods in the design of cryptographic protocols (state of the art). Technical report, Swiss Federal Institute of Technology, Institute for computer Communications and Applications (ICA), November 1999.

[43] L. Buttyán. *Buliding blocks for secure services: Authenticated key transport and rational exchange protocols.* PhD thesis, École Polytechnique Fédérale de Lausanne, 2002.

[44] L. Buttyán, S. Staamann, and U. Wilhelm. A simple logic for authentication protocol design. In *Proceeding of the IEEE CS Computer Security Foundations Workshop*, 1998.

[45] L. Buttyán and I. Vajda. *Kriptográfia és alkalmazásai*. TypoTex, 2004.

[46] M. Cagalj, S. Capkun, and J-P. Hubaux J-P. Key agreement in peer-to-peer wireless networks. In *Proceedings of the IEEE (Special Issue on Security and Cryptography)*, volume 92, 2006.

[47] I. Cervasato, N. Durgin, P. Lincoln, J. Mitchell, and A. Scedrov. A metanotation for protocol analysis. In *Proceedings of 12th IEEE Computer Security Foundations Workshop*. IEEE Computer Society Press, June 1999.

[48] T. Coffey and P. Saidha. Logic for verifying public-key cryptographic protocols. *IEEE Proceedings Computers and Digital Techniques*, 144(1):28–32, 1997.

[49] T. Coffey, P. Saidha, and P. Burrows. Analysing the security of a non-repudiation communication protocol with mandatory proof of receipt. In *Proceedings of International Symposium on Information and Communication Technologies (ISICT03)*, pages 370–376, 2003. Trinity College, Dublin; ISBN 0-9544145-2-7.

[50] G. Di Crescenzo, R. Ostrovsky, and S. Rajagopalan. Conditional oblivious transfer and timed-release encryption. In *Advances in Cryptology - Eurocrypt'99*, LNCS 1592, pages 74–89. Springer-Verlag, 1999.

[51] Z. Csajbók. Orvosi protokollok információtechnológiai reprezentációja. Technical report, Debreceni Egyetem, Egészségügyi Kar, Egészségügyi Informatika Tanszék, 2008.

[52] Z. Csajbók, P. Takács, K. Bodnár, and Zs. Kristóf. E-learning rendszerek biztonsági jellemzőinek vizsgálata. In *II. Nyíregyházi Doktorandusz Konferencia, Nyíregyházi Főiskola rendezésében*, 2008. Megjelenés alatt.

[53] D. E. Denning and G. M. Sacco. Timestamps in key distribution protocols. *Communications of the ACM*, 24.(8.):533–536., August 1981.

[54] W. Diffie, P. van Oorschot, and M. Wiener. Authentication and authenticated key exchanges. *Design, Codes and Cryptography*, 2:107–125, 1992.

[55] R. Dojen, T. Coffey, and L. Tian. A security protocol specification language extension for modal logic-based verification. In *2007 IET China-Ireland International Conference on Information and Communications Technologies CIICT07*, pages 295–302, August 2007. Dublin, Ireland.

[56] D. Dolev, S. Even, and R. M. Karp. On the security of ping-pong protocols. *Information and Control*, 55(1-3):57–68, October/November/December 1982.

[57] D. Dolev and A. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, IT-29(2):198–208, March 1983.

[58] C. Ellison. Establishing identity without certification authorities. In *Proceedings of the Sixth Annual USENIX Security Symposium*, pages 67–76., October 1996.

[59] S. Even and O. Goldreich. On the security of multi-party ping-pong protocols. In *Proceedings of the 24th IEEE Symposium on the Foundations of Computer Science*. IEEE Computer Society Press, 1983.

[60] M. Ferenczi. *Matematikai logika*. Műszaki Könyvkiadó, 2002.

[61] J. Folláth, A. Huszti, and A. Pethő. Designin asymmetric authentication system. In *Proceedings of the 7th ICAI Conference, Eger*, number 1, pages 53–61., 2007.

[62] G. Fülöp. *Az információ*. Eötvös Loránd Tudományegyetem, Könyvtártudományi - Informatikai Tanszék, 1996.

[63] J. Garay and M. Jakobsson. Timed release of standard digital signatures. In *Financial Crypto'02*, LNCS 2357, pages 168–182. Springer-Verlag, 2002.

[64] J. Garay and C. Pomerance. Timed fair exchange of standard signatures. In *Financial Crypto'03*, LNCS 2742, pages 190–207. Springer-Verlag, 2003.

[65] L. Gavin. An attack on the needham-schroeder public key authentication protocol. *Information Processing Letters*, 56.(3.):131–136., November 1995.

[66] C. Gehrmann, C. J. Mitchell, and K. Nyberg. Manual authentication for wireless devices. *Cryptobytes*, 7(1):29–37, 2004.

[67] Y. Glouche, T. Genet, O. Heen, and O. Courtay. A security protocol animator tool for AVISPA (SPAN). In *ARTIS2 Workshop on Security Specification and Verification of Embedded Systems*, May 2006. Pisa.

[68] S. Goeman. Specification of prototypes - D11, IST - 2000 - 25350 - SHAMAN, Public Report. http://www.isrc.rhul.ac.uk/shaman/docs, March 2003. D11v2.pdf.

[69] O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity or all languages in np have zero-knowledge proof systems. *Journal of the Association for Computing Machinery*, 38(1):691–729, July 1991.

[70] P. Golle and M. Jakobsson. Reusable anonymous return channels. In *WPES'03*, 2003.

[71] L. Gong, R. Needham, and R. Yahalom. Reasoning about belief in cryptographic protocols. In *Proceedings of the IEEE CS Sysmposium on Research in Security and Privacy*, pages 234–248, 1990.

[72] L. Gong and P. Syverson. Fail-stop protocols: A new approach to design secure protocols. In *Proceedings of the 5th International Working*

Conference on Dependable Computing for Critical Applications, pages 44–55, 1995.

[73] O. Heen, T. Genet, S. Geller, and N. Prigent. An industrial and academic joint experiment on automated verification of a security protocol. In *IFIP Networking Workshop on Mobile and Networks Security*, May 2008. Singapore.

[74] N. Heintze and J. Tygar. A model for secure protocols and their compositions. *IEEE Transactions on Software Engineering*, 22:2–13, 1996. Proseedins of the IEEE CS Symposium on Research in Security and Privacy, 1994, pp. 2-13.

[75] J. Hintikka. *Knowledge and Belief. An Introduction to the Logic of the Two Notions.* Cornell University Press, 1962.

[76] J. Hintikka. *Részletek Jaakko Hintikka Tudás és Hit (Bevezetés a két fogalom logikájába) című művéből. A hiedelmek természete, szerveződése és szerepe a mindennapi tudatban című munkaértekezlet segédanyaga 7.* Fordítási részek. Fordította: Berkes Ildikó, 1975.

[77] C. M. Holloway. Why engineers should consider formal methods. In *16th Digital Avionics Systems Conference*, October 1997.

[78] D. Hristu-Varsakelis, K. Chalkias, and G. Stephanides. Low-cost anonymous timed-release encryption. In *Proceedings of the Third International Symposium on Information Assurance and Security*, pages 77–82. IEEE Computer Society, 2007.

[79] A. Huszti. A secure electronic voting scheme. *Periodica Polytechnica, Electrical Engineering*, 51(3-4):141–146., 2007.

[80] Clay Mathematics Institute. Millenium problems. http://www. claymath.org/millenium.

[81] Isabelle. http://www.cl.cam.ac.uk/research/hvg/Isabelle/index.html.

[82] A. Kehne, J. Schonwalder, and H. Langendorfer. A nonce-based protocol for multiple authentications. *Operating Systems Review*, 26(4):84–89, October 1992.

[83] A. Keromythis and J. Smith. Creating efficient fail-stop cryptographic protocols. Technical Report MS-CIS-96-32, University of Pennsylvania, December 1996.

[84] J. Ködmön. *Kriptográfia (Az informatikai biztonság alapjai, a PGP kriptorendszer használata).* ComputerBooks, 1999/2000.

[85] J. Ködmön and K. Bodnár. Biztonságosabb felhasználóazonosítás az egészségügyben. *IME*, VI.(9. szám):46–51, November 2007.

[86] Magyar Közlöny. 164/2006. (XII. 27.) az egyes főbb betegségcsoportok finanszírozási eljárásrendjének kidolgozása, szerkesztése és szakmai egyeztetése lefolytatásának egységes szabályairól. Magyar Közlöny 2006. 164. szám, 2006. december 27., 2006. http://kozlonykiado.hu/nkonline/MKPDF/hiteles/mk06059.pdf, Egészségügyi Minisztérium rendelete.

[87] Magyar Közlöny. 23/2006. (V. 18.) rendelet a vizsgálati és terápiás eljárásrend kidolgozásának, szerkezetének és szakmai egyeztetése lefolytatásának eljárásrendjéről. Magyar Közlöny 2006. 59. szám, 2006. május 18., 2006.
http://kozlonykiado.hu/nkonline/MKPDF/hiteles/mk06059.pdf, Egészségügyi Minisztérium rendelete.

[88] S. Kramer. Logical concepts in cryptography.

[89] Zs. Kristóf, Z. Csajbók, P. Takács, K. Bodnár, and J. Ködmön. Azonosítón alapuló kriptográfiai rendszerek alkalmazása e-learning környezetben. In *Multimédia a felsőoktatásban, Budapest*, 2008.

[90] M. Kudo and A. Mathuria. An extended logic for analyzing timed-release public-key protocols. In *Proceedings Information and Commu-*

nication Security, Second International Conference, ICICS'99, Sysdney, pages 9–11, November 1999.

[91] M. G. Kuhn and R. J. Anderson. Soft tempest: Hidden data transmission using electromagnetic emanations. In David Aucsmith, editor, *Information Hinding, Second International Workshop, IH'98*, number 1525 in LNCS, pages 124–142. Springer-Verlag, 1998.

[92] T. Kyntaja. A logic of authentication by Burrows, Abadi and Needham. In *Proseedings of Helsinki University of Technology, Seminar on Network Security*, 1995. Chapter 5.

[93] S. Laur and K. Nyberg. Efficient mutual data authentication using manually authenticated string. http://eprint.iacr.org/2005/424. Cryptology ePrint Archive, Report 2005/424, 2006. : Extended Version. A shorter more compact version was published at CANS 2006.

[94] Y. Li and T. Newe. On the security analysis of authenticated group key exchange protocols for low-power mobile devices. In *International Conference on Signal Processing and Communication Systems (ICSPCS'07)*, December 2007. Gold Coast, Australia; ISBN: 978-0-9756934-3-8; Paper Num 165.

[95] L. Lovász. *Algoritmusok bonyolultsága*. ELTE TTK, 1994. Nemzeti Tankönyvkiadó.

[96] G. Lowe. Breaking and fixing the Needham-Schroeder public key protocol using FDR. In *Proceeding of the TACAS*, pages 147–166, 1996.

[97] G. Lowe. A family of attacks upon authentication protocols. Technical report, Department of Mathematics and Computer Science, University of Leicester, 1997.

[98] W. Mao. An augmentation of BAN-like logics. In *Proceedings of th 8th IEEE Workshop on Computer Security Foundations*. IEEE Computer Society, 1995.

[99] W. Mao. Timed-release cryptography. Technical Report HPL-2001-37, Hewlett Packard Development Company, 2001.

[100] W. Mao and C. Boyd. Towards formal analysis of security protocols. In *Computer Security Foundations Workshop VI.*, pages 147–158. IEEE Computer Society Press, 1993.

[101] M. Marcos, G. Berger, A. ten Teije F. van Harmelen, H. Roomans, and S. Miksch. Using critiquing for improving medical protocols: harder than it seems. In *Proceedings of the 8th European Conference on Artifical Intelligence in Medicine (AIME-2001)*, pages 431–441. Springer-Verlag, 2001.

[102] M. Marcos, H. Roomans, A. ten Teije, and F. van Harmelen. Improving medical protocols through formalisation: a case study. In *Session on Formal Methods in Healthcare, 6th International Conference on Integrated Design and Process Technology (IDPT-02)*, 2002.

[103] M. Marcos, F. van Harmelen, and A. ten Teije. PROTOCURE: Improving medical protocols by formal methods. http://www.protocure.org/old/resources-publications.html, February 2002. Visited: 2009.02.23.

[104] T. May. Timed-release crypto. In Manuscript; http://www.hks.net/cpunks/cpunks-0/ 1460.html; Visited: 2009.02.18., 1993.

[105] C. A. Meadows. Applying formal methods to the analysis of a key management protocol. *Journal of Computer Security*, 1(1):5–35, 1992.

[106] C. A. Meadows. Formal verification of cryptographic protocols: A survey. In *ASIACRYPT: International Conference on the Theory and Application of Cryptology, LNCS, Springer-Varlag*, 1994.

[107] C. A. Meadows. Open issues in formal methods for cryptographic protocol analysis. In *Proceedings of DISCEX 2000*. IEEE Computer Society Press, 2000.

[108] C. A. Meadows. Formal methods for cryptographic protocol analysis: Emerging issues and trends. *IEEE Journal on Selected Areas in Communicatios*, 21:44–54, January 2003.

[109] H. A. Medve and K. Tarnay. A formális nyelvek szerepe a távközlési szoftverek fejlesztésében. In *Networkshop 2001*, 2001.

[110] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied cryptography.* CRC Press, 1996. http://www.cacr.math.uwaterloo.ca/hac.

[111] R. Merkle. Secure communication over insecure channels (1974). http://www.itas.fzk.de/marhp/weber/merkle.htm, January 2002. With an Interview from year 1995, edited by A. Weber.

[112] T. Mihálydeák. *Az informatika logikai alapjai.* University Debrecen, 2007. Egyetemi jegyzet.

[113] I. Mironov. Hash functions: Theory, attacks, and applications. Technical Report MSR-TR-2005-187, Microsoft Research, November 2005.

[114] S. Mödersheim, L. Viganó, and D. von Oheimb. Automated Validation of Security Protocols (AVASP). In *AVASP'05 EATPS 2005 Tutorial*, 2005.

[115] D. Monniaux. Analysis of cryptographic protocols using logics of belief: an overview. *Journal of Telecommunications and Information Technology*, 4:57–67, 2002.

[116] M. C. Mont, K. Harrison, and M. Sadler. The HP time vault service: Innovating the way confodential information in disclosed at the right time. In *12th International World Wide Web Conference*, pages 160–169. ACM Press, 2003.

[117] R. Needham and M. Schroeder. Using encryption for authentication in large networks of computers. *Communication of theACM*, 21(12):993–999, December 1978.

[118] R. M. Needham and M. D. Schroeder. Using encryption for authentication in large networks of computers. *Communicatins of the ACM*, 21(12):993–999, 1978.

[119] B. C. Neuman and S. Stubblebine. A note on the use of timestamps as nonces. *Operating Systems Review*, 27(2):10–14, April 1993.

[120] T. Newe and T. Coffey. Verifying a minimum-knowledge authentication protocol for use in power-line networks. In *Proceedings of ISPLC2000 (2000 International Symposium on Power-line Communications 2000)*,, pages 109–116, April 2000. Limerick, Ireland.

[121] T. Newe and T. Coffey. Hybrid mobile security protocol: Formal verification using a new modal logic. In *Proceedings of ICAI-2002*, December 2002. Puerto De La Cruz, Spain.

[122] T. Newe and T. Coffey. Formal verification logic for hybrid security protocols. *International Journal of Computer Systems Science & Engineeing*, pages 17–25, 2003.

[123] T. Newe and T. Coffey. *Recent Advances in Communications and Computer Science*, chapter On the Logical Verification of Key Exchange Protocols for Mobile Communications, pages 76–81. WSEAS Press, 2003. ISBN:960-8052-86-6.

[124] I. Osipkov, Y. Kim, and J. H. Cheon. Timed-release public key based authentiocated encryption. http://eprint.iacr.org/2004/231, 2004.

[125] C. H. Papadimitriou. *Számítási bonyolultság*. Egyetemi tankönyv, Novadat Bt., 1999.

[126] M. Peleg, S. Tu, J. Bury, P. Ciccarese, J. Fox, R. Greenes, R. Hall, P. Johnson, N. Jones, A. Kumar, S. Miksch, S. Quaglini, A. Seyfang, E. Shortliffe, and M. Stefanelli. Comparing computer-interpretable guideline models: A case-study approach. *JAMIA*, 10, 2003.

[127] A. Pethő. Paraméterválasztás nyilvános kulcsú kriptográfiai rendszereknél. http://www.szatki.hu/ sztibor/eszigno/kriptografia.html, Május 2002. Kriptográfia és alkalmazásai szeminárium, SZTAKI, Budapest. Visited: 2009.02.23.

[128] P. Pleva. Anonim digitális pénz mobilkörnyezetben. Master's thesis, Debreceni Egyetem, 2008.

[129] A. Rácz. Orvos-szakmai protokollok formális ellenőrzése az egészségügyben. Master's thesis, Debreceni Egyetem, Egészségügyi Kar, Egészségügyi Informatika Tanszék, 2008.

[130] R. Rivest, A. L. Shamir, and D. A. Wagner. Time-lock puzzles and timed-release crypto. Technical Report 684, MIT Laboratory for Computer Science, 1996.

[131] A. Roscoe. Modelling and veryfying key exchange protocols using CSP and FDR. In *Proceedings of the IEEE CS Symposium on Research in Security and Privacy*, pages 98–107, 1995.

[132] P. Ryan, S. Schneider, M. Goldsmith, G. Lowe, and A. W. Roscoe. *The Modelling and Analysis of Security Protocols.* Pearson Education Limited, 2001.

[133] B. Schneier. *Applied Cryptography: Protocols, Algorithms and Source Code in C.* John Wiley & Sons, Inc., 1996.

[134] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal of Computer Security*, 27:379–423 and 623–656, July and October 1948.

[135] R. Sips, L. Braun, and N. Roos. Applying intention-based guidelines for critiquing. In *Workshop on AI techniques in healthcare: evidence-based guidelines and protocols*, August 2006. Riva del Garda, Italy. Held in conjunction with ECAI-06 - European Conference on Artifical Intelligence.

[136] E. Snekkenes. *Formal Specification and Analysis of Cryptographic Protocols*. PhD thesis, University of Oslo, Norway, 1995.

[137] SPORE. Security Protocols Open REpository, http://www.lsv.ens-cachan.fr/software/spore. Visited: 2009.02.17.

[138] T. Storer. BAN logic analysis of the UK rolling registration and postal voting systems. Technical report, University of St. Andrews, June 2003.

[139] C. A. Sunshine. Formal methods for communication protocol specification and verification. Technical Report N-1429-ARPA/NBS, Defense Advanced Research Projects Agency and the National Bureau of Standards, November 1979.

[140] P. Syverson. Adding time to a logic of authentication. In *Proceeding of the ACM Conference on Computer and Communications Security*, pages 97–101, November 1933.

[141] P. Syverson. Limitations on design principles for public key protocols. In *Proceedings of the IEEE CS Symposium on Research in Security and Privacy*, pages 62–73, 1996.

[142] P. Syverson and C. Meadows. A logical language for specifying cryptographic protocol requirements. In *Proceedings of the IEEE CS Symposium on Research in Security and Privacy*, pages 165–177, 1993.

[143] P. Syverson and P. van Oorschot. On unifying some cryptographic protocol logics. In *Proceedings of the IEEE CS Symposium on Research in Security and Privacy*, pages 14–28, 1994.

[144] M. D. Szabó. Biometrikus azonosítás és adatvédelem. *Acta Humana*, 1., 2004.

[145] P. Takács. The additional examination of the Kudo-Mathuria time-release protocol. *Journal of Universal Computer Science*, 12(9):1373–1384, 2006. Submitted: 31/12/05, accepted: 12/05/06, appeared: 28/09/06.

[146] P. Takács. The extension of CSN-logic for multi-channel protocols. In *Proceedings of the 7th ICAI Conference, Eger*, pages 147–154, 2007. Reviewed by Zentralblatt für Mathematik.

[147] P. Takács. A Kudo-Mathuria protokoll vizsgálata az AVISPA protokollellenőrző rendszerben. In *II. Nyíregyházi Doktorandusz Konferencia*. Nyíregyházi Főiskola, Nyíregyházi Főiskola, November 2008. Megjelenés alatt. Lektorálta: dr. Ködmön József és Vályi Sándor.

[148] P. Takács and S. Vályi. Többcsatornás kriptográfiai protokollok vizsgálata a bővített CSN-logika eszközeivel. In *I. Nyíregyházi Doktorandusz Konferencia, DE-EK*, December 2007. Megjelenés alatt.

[149] P. Takács and S. Vályi. An extension of protocol verification modal logic to multi-channel protocols. *Tatra Mountains Mathematical Publications*, 41:153–166, 2008.

[150] P. Takács and S. Vályi. Javaslat a MANA II kriptográfiai protokoll korrekciójára. In *Informatika a felsőoktatásban 2008*, Augusztus 2008.

[151] L. Tóthfalusi. Az evidence based medicine statisztikai alapjairól. *Cardiologia Hungarica*, 33(1):55–59, 2003.

[152] M-J. Toussaint. Deriving the complete knowledge of participants in cryptographic protocols. In *Advances in Cryptology - CRYPTO'91*, pages 24–43. Springer-Verlag, 1992.

[153] M. Vuagnoux and S. Pasini. Compromising electromagnetic emanations of wired and wireless keyboards. Technical report, École Polytechnique FédÉrale de Lausanne, School of Computer and Communication Sciences, Security and Cryptography Laboratory, 2008.

[154] M. Weiser. Some computer science issues in ubiquitous computing. *CACM*, 1993. Reprinted as "Ubiquitous Computing". Nikkei Electronics; December 6, 1993; pp. 137-143.

[155] J. Wessels. Applications of BAN-logic. Technical report, IPA, April 2001. IPA Spring Days on Security, Kapellerput, Heeze, April 18-20, 2001.

[156] P. Windirsch. Security for mobile systems beyond 3G - Presentations and posters of the IST - 2000 - 25350 - SHAMAN Worshop, 2002. http://www.isrc.rhul.ac.uk/shaman/docs, 2002.

[157] F-L. Wong and F. Stajano. Multi-channel protocols. In *Proceeding of Security Protocols, 13th International Workshop, Cambridge, UK*, volume 4631 of *Lecture Notes in Computer Science*. Springer-Verlag, April,20-22 2005.

[158] F. L. Wong and F. Stajano. Multichannel security protocols. *Pervasive computing*, pages 31–39, October-December 2007.

[159] T. Woo and S. Lam. A semantic model for authentication protocols. In *Proceedings of the IEEE CS Symposium on Research in Security and Privacy*, pages 178–194, 1993.

[160] R. Yahalom, B. Klein, and T. Beth. Trust relationships in secure systems: A distributed authentication perspective. In *Proceedings of the 1993 IEEE Symposium on Security and Privacy*, pages 150–164. IEEE Computer Society Press, May 1993.

[161] J. Yan, A. Blackwell, R. Anderson, and A. Grant. The memorability and security of passwords - some empirical results. Technical report, University of Cambridge, Computer Laboratory, September 2000.

[162] S. Yang and X. Li. A limitation of BAN logic analysis on a man-in-the-middle attack. *Journal of Informmation and Computing Science*, 1(3):131–138, 2006.

[163] Complexity Zoo. http://qwiki.stanford.edu/wiki/complexity_zoo.

Publikációs lista

1. P. Takács *The Additional Examination of the Kudo-Mathuria Time-Release Protocol*, Journal of Universal Computer Science, vol 12, no.9 (2006), 1373-1384. Submitted: 31/12/05, accepted: 12/05/06, appeared: 28/09/06.

2. P. Takács, *The extension of CNS-logic for multi-channel protocols.* Proceedings of the 7th ICAI Conference, Eger, 2007, 147-154.

3. P. Takács, Zs. Kristóf, *The investigation of the development of programming languages*, Proceedings of the 7th ICAI Conference, Eger, 2007, 327-332.

4. P. Takács, S. Vályi, *An extension of protocol verification modal logic to multi-channel-protocols*, Tatra Mountains Mathematical Publications - TATRACRYPT 2007. Editors: O. Grosek, K. Nemoga, M. Vojvoda. Vol. 41. (2008), 153-166.

5. Takács P. *A Windows NT biztonsági jellemzői*, Informatika a Felsőoktatásban'99 Konferencia kiadvány, Debrecen, 1999.

6. Takács P. *Bevezetés az Internet használatába*, Fejezet a Bevezetés az alkalmazott kutatásmódszertanba című tankönyvben, Pro Educatione Alapítvány, Nyíregyháza, 2001. ISBN 963 00 7697 7

7. Takács P. *A kriptográfia időtényezőiről.* Informatika a felsőoktatásban'05 Konferencia kiadvány, Debrecen 2005.

8. Takács P., *Hálózati alapismeretek I., II.* Távoktatási jegyzetek. HEFOP-3.5.1-K-2004-10-0001/2.0 országos pályázat keretében, a Nyíregyházi Regionális Képző Központ vezetésével, 2006. Nyelvi lektor: Takács Ferencné; Szakmai lektorok: Molnár Gábor (Györgyi Gyula, Szemcsák Imre - NYRKK belső lektorok).

9. Takács P., *Adatbázis-kezelés I., II.* Távoktatási jegyzetek. HEFOP-3.5.1-K-2004-10-0001/2.0 országos pályázat keretében, a Nyíregyházi Regionális Képző Központ vezetésével, 2006. Nyelvi lektor: Takács Ferencné; Szakmai lektorok: Máté István (Györgyi Gyula, Szemcsák Imre - NYRKK belső lektorok).

10. Takács P., Vályi S. *Javaslat a MANA II kriptográfiai protokoll korrekciójára*, Informatika a felsőoktatásban'08, Konferenciakiadvány, Debrecen 2008.

Előadások

1. Takács P., Ködmön J., *Egészségügyi informatika a DOTE Egészségügyi Főiskolai Karán*, Informatika a Felsőoktatásban '96 - Networkshop '96, Debrecen, 1996.

2. Ködmön J., Takács P., *Hálózatbiztonsági technikák az egészségügyben*, Informatika a Felsőoktatásban '96 - Networkshop '96, Debrecen, 1996.

3. Takács P., *Adatvédelem és egészségügy*, XX. Neumann Kollokvium, A számítástechnika orvosi és biológiai alkalmazásai, Veszprém, 1996.

4. Komoróczy T., Takács P., *DOM - Digitális OrvosMúzeum* Networkshop '97, 6. Országos konferencia és kiállítás, Keszthely, 1997.

5. Takács P., *A Windows NT biztonsági jellemzői.* Informatika a Felsőoktatásban '99, Debrecen, 1999.

6. Dr. Iszlai É., Dr. Ágoston S., Dr. Rácz F., Dr. Kapin M., Takács P., Dr. Szerafin L., *Szabolcs-Szatmár-Bereg megyei Helicobacter pylori (H.p.) seroepidemiológiai szűrésen részt vettek további vizsgálata (gastroscopia, szövettan, anti-CagA ea.)*, Magyar Gasztroenterológiai Társaság Endoszkópos Szekciójának ülése, Gödöllő, 2001. aug. 31. - szept. 01.

7. Kántor I., Gaál Zs., Takács P., Dicső F., Valenta B., *Halmozottan előforduló diabetes egy családon belül - MODY?*, Gyermek-

diabetalogiai Kongresszus, Dobogókő, 2002.11.25-26.

8. Kántor I., Gaál Zs., A.T. Hattersley, Stenszky V., Takács P. *'MODY 2' betegek utánkövetéses vizsgálata*, Gyermekdiabetologiai Kongresszus, Szeged, 2003.

9. Kántor I., Gaál Zs., A. T. Hattersley, Stenszky V., Takács P., *'MODY 2' betegek utánkövetéses vizsgálata (esetismertetés)* Sz.-Sz.-B. Megyei Önkormányzat Jósa András Kórház Tudományos Bizottságának Tudományos Ülése - 2003. Évi "Jósa András Pályázat" díjnyertes pályamű. Nyíregyháza, 2004.

10. Takács P., *A kriptográfia időtényezőiről*, Informatika a felsőoktatásban'05, Debrecen 2005.

11. Takács P., Ködmön J., *Az egészségügyi szervező képzés Nyíregyházán*, Informatika a felsőoktatásban'05, Debrecen 2005.

12. P. Takács, *On time-dependent cryptographic protocols and it's allpications*, ISBIS'05 Győr - International Syposium on Business Information Systems, 2005.

13. P. Takács, *On examine of Multi-channel Protocols*, NyírCrypt - 6th Central European Conference on Cryptography. Nyíregyháza, 2006.

14. P. Takács, *The Extension of CSN-logics: On Examine of Multichannel Protocols* ICAI'07 - Eger. 2007.

15. P. Takács, S. Vályi, *On Verification of the MANA Protocol Family*, 7th Central European Conference on Cryptology, Smolenice, 2007.

16. S. Vályi, P. Takács, J. Ködmön, *Algorithmic aspecs of some protocol verification logics*, 7th Central European Conference on Cryptology, Smolenice, 2007.

17. Takács P., Vályi S. *Többcsatornás kriptográfiai protokollok vizsgálata a bővített CSN-logika eszközeivel*, I. Nyíregyházi Doktorandusz Konferencia, DE-EK, 2007.

18. Takács P., Vályi S. *Javaslat a MANA II kriptográfiai protokoll korrekciójára*, Informatika a felsőoktatásban '08, Debrecen, 2008.

19. Kristóf Zs., Csajbók Z., Takács P., Bodnár K., Ködmön J. *Azonosítón alapuló kriptográfiai rendszerek alkalmazása eLearning környezetben*, Multimédia a felsőoktatásban '08 konferencia, Budapest, 2008.

20. Takács P. *A Kudo-Mathuria protokoll vizsgálata az AVISPA protokollellenőrző rendszerben.*, II. Nyíregyházi Doktorandusz Konferencia, Nyíregyházi Főiskola, 2008.

Acknowledgements

I am grateful to professor Attila Pethő for taking on the tasks of the supervisor. He supported my progress in many cases and he let me try myself in other areas.

I would like to say thanks to Tamás Mihálydeák for his help o compile my first publication and for his support thereafter.

Thanks to Sándor Vályi for the common work.

Thanks to Laurent Vigernon who helped without knowing me forming HLPSL codes and using the AVISPA system.

I wish to thank my colleagues for their encouragement and assistance.

Last, but not least I would like to thank my wife, my children, my father and mother, my family for their endless patience and love which they helped my work with.

Köszönetnyílvánítások

Köszönöm Pethő Attilának, hogy elvállalta a témavezetői feladatokat, sok esetben utat mutatott a továbblépéshez, türelemmel elviselte, hogy más területeken is kipróbáltam magam.

Szeretném megköszönni Mihálydeák Tamásnak az első közlemény összeállítása során nyújtott segítségét, és azt követő precíz szakmai tanácsokat.

Köszönöm Vályi Sándornak a közös munkát, az együttgondolkodást.

Köszönet Laurent Vigernon-nak, hogy ismeretlenül is segített elindulni a HLPSL kódolásban, és az AVISPA rendszer használatában.

Köszönöm munkatársaimnak a biztatást és a segítségnyújtást.

Végül, de nem utolsó sorban, köszönöm feleségemnek, gyermekeimnek, édesapámnak, édesanyámnak és családomnak azt a végtelen türelmet és szeretet, amellyel segítették munkámat.

Kriptográfiai protokollok formális vizsgálata a CSN logikai rendszer bővítésével

Értekezés a doktori (Ph.D.) fokozat megszerzése érdekében az informatika tudományágban.

Írta: Takács Péter okleveles matematika, fizika és számítástechnika tanár.

Készült a Debreceni Egyetem Informatikai Tudományok Doktori Iskolája (Digitális kommunikáció programja) keretében.

Témavezető: Prof. Dr. Pethő Attila

A doktori szigorlati bizottság:

elnök: Dr. . . . . .

tagok: Dr. . . . . .

Dr. . . . . .

A doktori szigorlat időpontja: 200. . . . . . .

Az értekezés bírálói:

Dr. . . . . .

Dr. . . . . .

Dr. . . . . .

A bírálóbizottság:

elnök: Dr. . . . . .

tagok: Dr. . . . . .

Dr. . . . . .

Dr. . . . . .

Dr. . . . . .

Az értekezés védésének időpontja: 200. . . . . . .