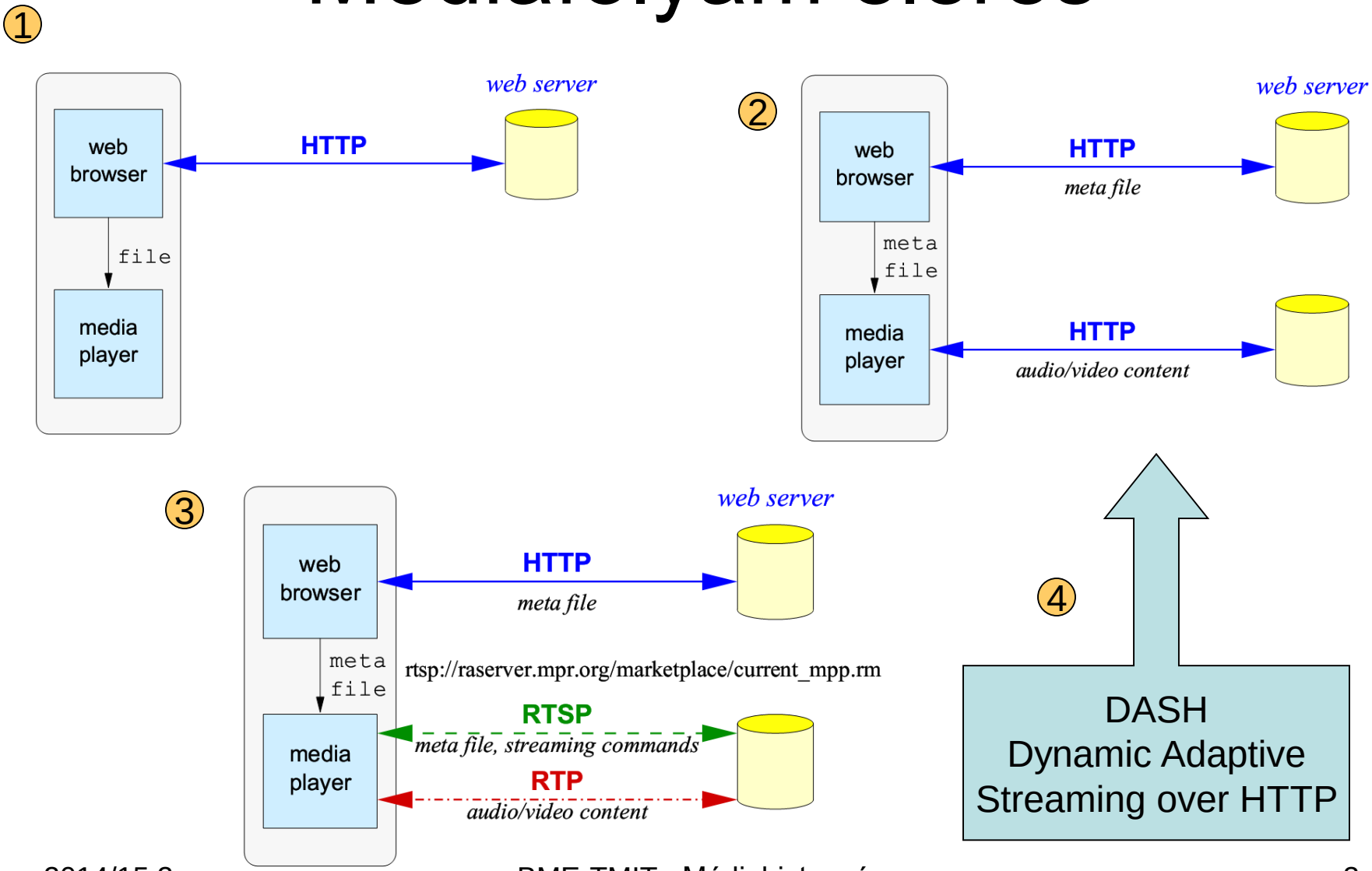
Médiafolyamok
titkosítása
SRTP – MIKEY - ZRTP

BME - TMIT
Médiabiztonság
feher.gabor@tmit.bme.hu

Médiafolyamok

- Médiafolyam valós idejű továbbítása Interneten
  - Videó becsomagolása
  - A csomagok valós idejű küldése
    - RTP – Real-time Transport Protocol
  - Videófolyamok vezérlése
    - RTSP – Real-Time Streaming Protocol

Médiafolyam elérés



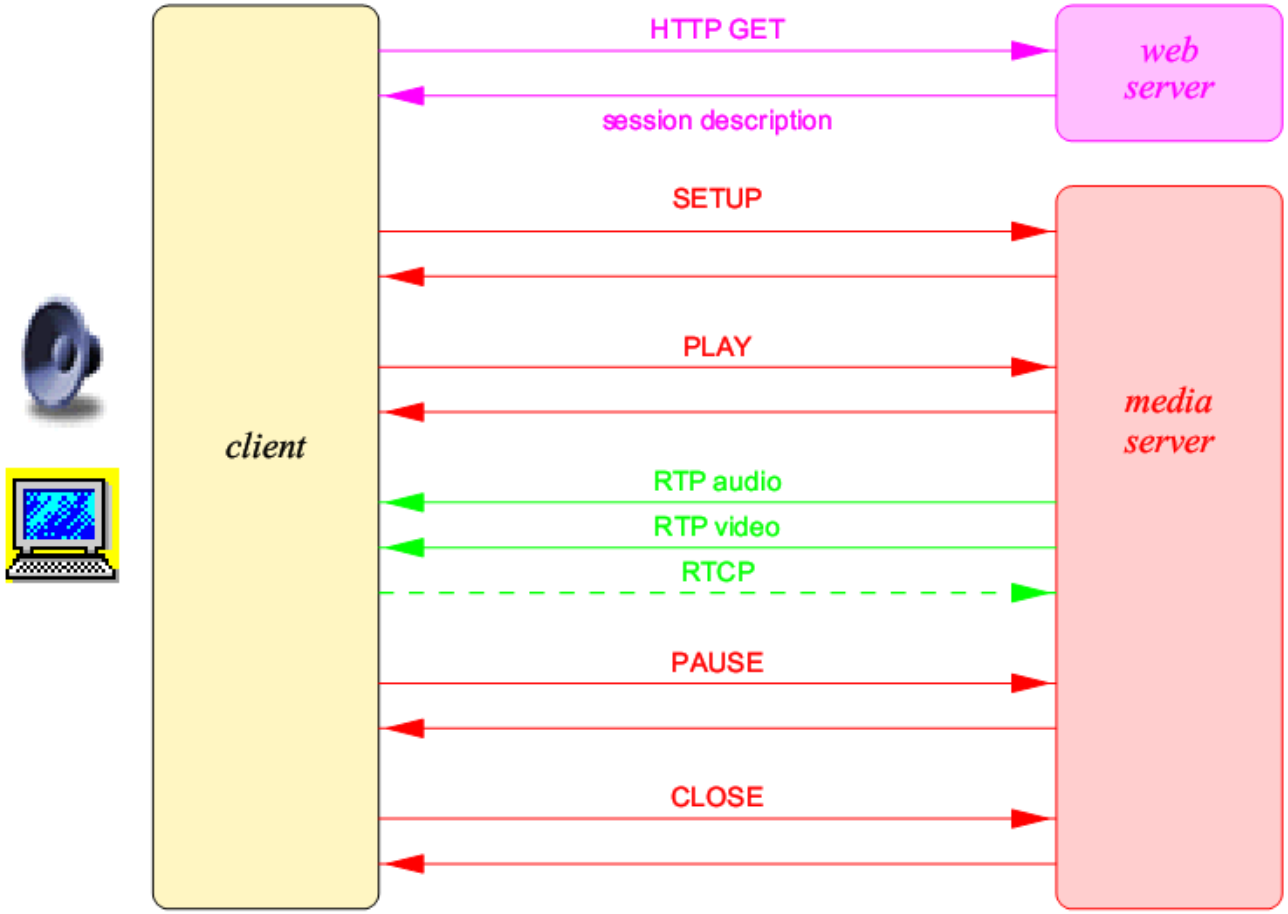
RTSP / Real Time Streaming Protocol

- Durva szinkronizáció
- Teljesítmény elosztás
- Viszonyleírás
- Egység vezérlés (pl.: pan, zoom, tilt)
- Cache

- Nagyon hasonló a HTTP-hez
  - Szöveg + MIME fejlécek
  - Requsest / response
  - Státusz kódok
  - URL
  - HTTP kiegészítések használata (pl. hitelesítés, SSL)

  - DE! : szerver->kliens, adatok a protokollon kívül

RTSP működése

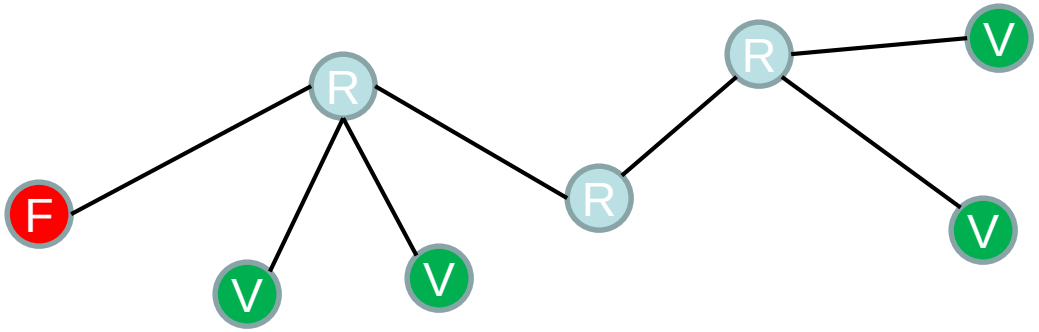


RTSP parancsok

- OPTIPONS, SETUP, ANNOUNCE, DESCRIBE, PLAY, RECORD, REDIRECT, PAUSE, SET_PARAMETER, TEARDOWN

Multicast

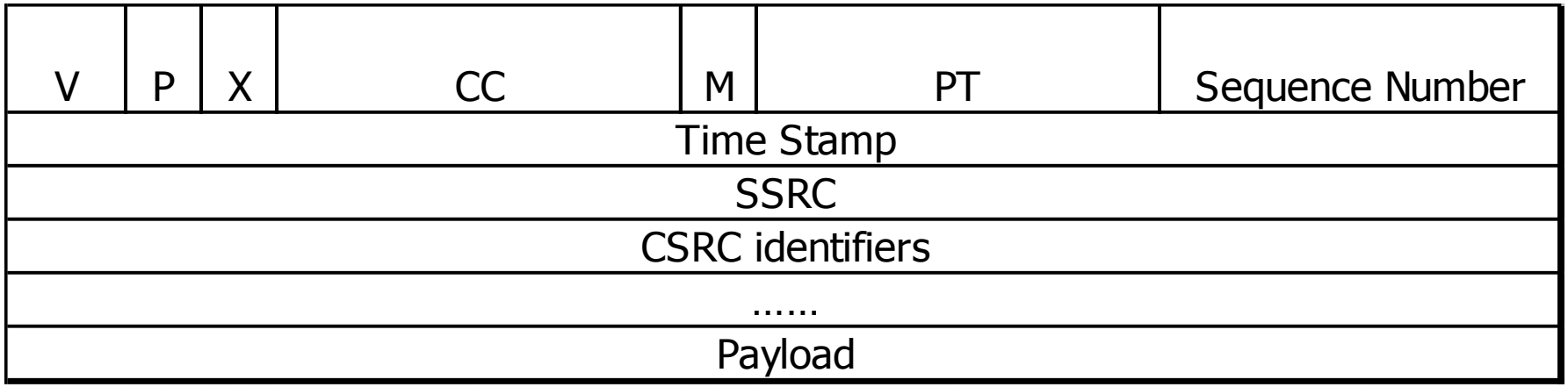
- IP folyam: több küldő – több vevő
  - Speciális címzés (Class D) 224.x.x.x – 239.x.x.x
  - Csatlakozás / leiratkozás: IGMP (Internet Group Management Protocol)
  - A routerek állítják össze a multicast fát, amely alapján az IP csomagok haladnak (multicast protokollok)
  - TCP nem alkalmas a visszajelzés miatt. Helyette általában UDP. De van megbízható multicast is..
  - IPTV használja (1 forrás, több nyelő szituáció)



RTP / Real-time Transport Protocol

- Multimédia adatok valósidejű átvitele
  - UDP csomagokban
    - Vagy akár UDPLite, DCCP (Datagram Congestion Control Protocol)
  - RTCP – Real-Time Control Protocol
  - Unicast és multicast esetek

RTP formátum



- V: verzió – 2 bit
- P: kitöltés – 1 bit
- X: kiterjesztés – 1 bit
- CC: CSRC szám – 4 bit
- M: Jelölő – 1 bit
- PT: RTP adat típusa – 7 bit
- Seq. number – 16 bit
- Időbélyeg – 32 bit
- SSRC – 32 bit
- CSRC litsa

RTP mezők

- Kitöltés
  - Blokk titkosító esetén kitöltést kell használni. P értéke 1
  - A kitöltés méretét a kitöltés utolsó bájtja jelöli
- Kiterjesztés
  - A fejléc után egy másik kiterjesztett fejléc található
- Jelölő mező
  - Fontos események jelzése
- SSRC
  - Forrás azonosítása
  - Véletlen érték
- CSRC mezők
  - Amennyiben a folyam egy mixeren megy keresztül, akkor a mixelésben résztevő források azonosítója. Csak az első 15 ilyen forrás jelölhető meg

RTP mezők 2.

- Sorszám
  - Sorszámozás szabályai
    - Minden csomag sorszámozva van
    - A sorszám mindig eggyel nő
    - A kezdeti sorszám véletlen érték
  - Sorszámok felhasználása
    - Csomagvesztés jelzése
    - Csomagsorrend helyreállítása
    - Többszálas feldolgozás
- Időbélyeg
  - A média mintavételezési idelye
  - Véletlen értékről indul

RTP mezők 3.

- Az RTP fejlécben nincs közvetlenül hossz jelzés
  - Nem szükséges, hiszen a szállított adat megegyezik az alatta lévő réteg adatával
  - Ha padding van, akkor az utlsó bájt a padding hossza
- RTP fejléc hossza
  - Az RTP fejléc meglehetősen hosszú
    - Hang esetén pl.: 40 bájt fejléc, 33 bájt adat
  - Tömöríteni kell a fejlécet! (csak L2 szinten tudjuk)
    - ROHC (akár 2 bájtos fejléc)

RTP adat típusok

- Hang
  - 0: PCMU
  - 8: PCMA
  - 9: G722
  - 4: G723
  - 15: G728
  - 18: G729
- Videó
  - 31: H261
  - 34: H263

- RFC 2250 - RTP Payload Format for MPEG1/MPEG2 Video
- RFC 3984 - RTP Payload Format for H.264 Video
- RFC 5215 - RTP Payload Format for Vorbis Encoded Audio

Több média esetén

- RTP esetén minden média külön kerül átvitelre
  - Nincs multiplexelés
  - Pl.: kép és hang külön utazik, azok szinkronizálására csak a vevőnél kerül sor
    - Szinkronizálás a sorszámok alapján
  - Nem feltétlenül külön port

RTCP – Real-Time Control Protocol

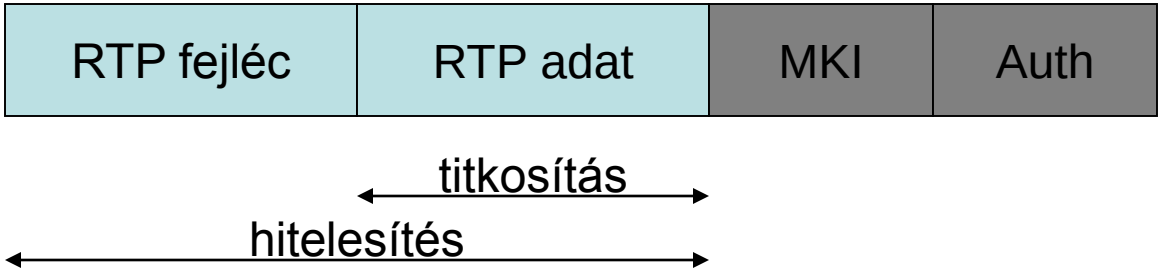
- Visszajelzés az RTP használatáról
  - SR: sender report
    - Az aktív résztvevők küldik
  - RR: receiver repot
    - A többi résztvevő küldi
  - SDES: source description
    - A végpont szöveges azonosítása (CNAME)
    - Opcionális mezők: név, email, telefon, …
  - BYE
    - RTP küldés vége
  - APP
    - Új alkalmazások és funkciók

SRTP - Secure RTP

- RFC 3711 / 2004
- Célok:
  - RTP és RTCP csomagok titkosítása, integritás védelme és védelem a visszajátszás ellen
  - Egyetlen kulcs és salt

SRTP - Secure RTP

- Az RTP csomagok titkosítása
  - Az RTP fejléc tömörítésre kerül, titkosítása így nem megoldható
  - Az RTP multimédia adat része titkosítható
  - Az RTP fejléc is hitelesíthető



SRTP mezők

- MKI: master Key Identifier
  - Azonosítja a kulcsot, amivel a csomagot titkosították
  - Növeli a csomag méretét
- Auth:
  - Integritásvédelem a csomagnak

- Belső paraméterek - kontextus
  - ROC: Rollover counter (32 bit)
    - Számolja, hogy a 16 bites SEQ hányszor fordult körbe
    - Így az SRTP sorszám már 48 bites
  - Master key:
    - A viszonykulcs előállításához szükséges titok
    - Alapértelmezésben 128 bit hosszú a viszonykulcs
  - Master salt:
    - A viszonykulcs melletti kulcs
    - Véletlen érték, de lehet publikus is
    - Alapértelmezésen 112 bit hosszú a viszonyban a salt

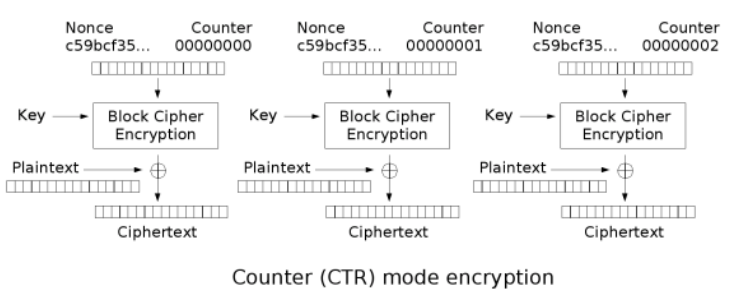
SRTP titkosítás

- AES blokk titkosító használata
  - A jelenlegi legbiztonságosabbnak és egyben
  - Leghatékonyabbnak tartott blokk titkosító
- Blokk titkosító átállítása folyam titkosítóvá
  - Nagyobb hatékonyság
    - Nincsenek blokkok
    - Nincs kitöltés
  - Hibaterjedés megállítása
    - Bithibából bithiba lesz

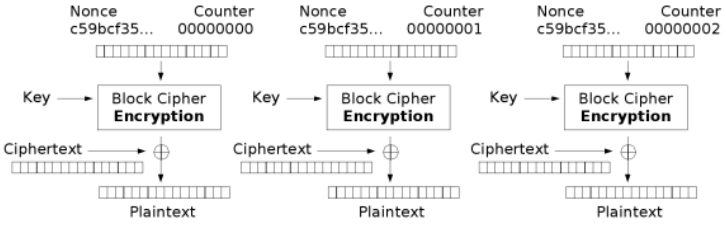
AES-CTR

- Counter (CTR) mode operation
  - A blokktitkosítót egy számláló vezérli
    - A számláló mellett van titkos rész is: IV
  - Kimenetként egy kulcsfolyam áll elő
  - A kulcsfolyam és a nyílt szöveg XOR kapcsolata adja a titkosított szöveget

- i = ROC << 16 + SEQ
- IV = (salt << 16) XOR (SSRC << 64) XOR (i << 16)



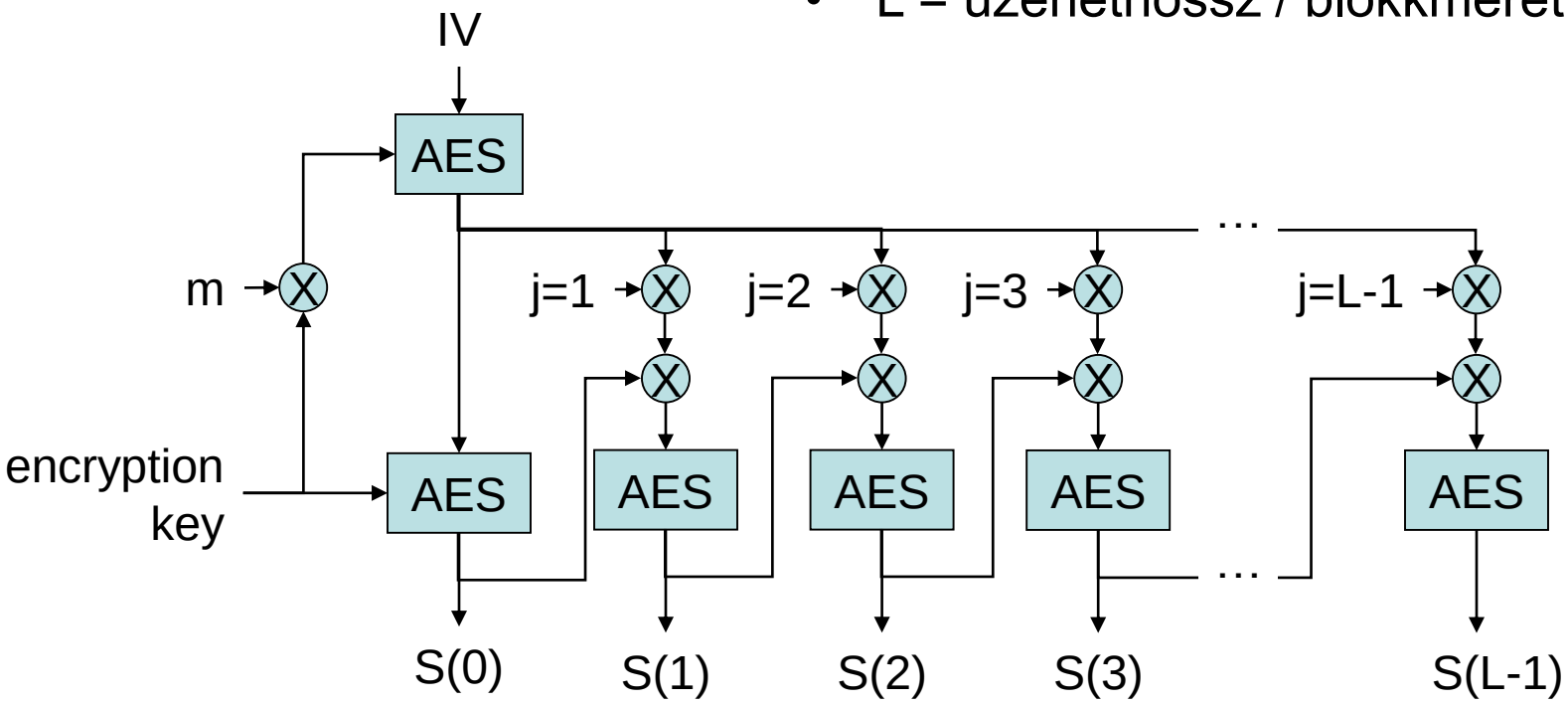
Counter (CTR) mode encryption



Counter (CTR) mode decryption

AES-F8

- F8 mode operation
  - UMTS esetén használják
- m = salting key || 010101…..01
- IV = 0x00 || M || PT || SEQ || TS || SSRC || ROC
- L = üzenethossz / blokkméret

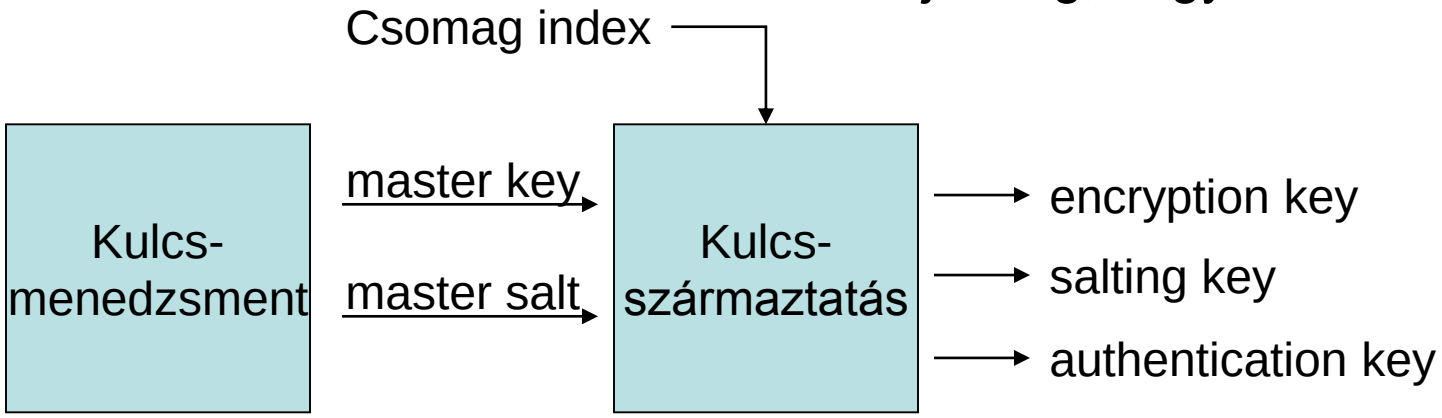


SRTP hitelesítés

- A hitelesítéshez a HMAC-SHA1 kulcsos hash függvényt használják
  - SHA-1: Secure Hash Algorithm
    - 160 bites kimenet, 512 bites blokkméret
  - HMAC:
    - HMAC(h,k,m) = h((k ⊕ opad) || h((k ⊕ ipad) || m))
      - ipad = 0x363636…36 A blokkméret hosszában
      - opad = 0x5c5c5c…5c A blokkméret hosszában
- A kulcs a viszonyhoz tartozó hitelesítő kulcs
- A kimenetet gyakran csonkolják (baloldali bitek)
  - Kisebb csomagok (akár csak 80 bites hitelesítő)

Kulcsok származtatása

- A encryption, salting és authentication kulcsokat újraszámolják
  - A kezdeti kulcshoz kötelező
  - Utána a kulcsszármaztatási ráta szabja meg, hogy mikor



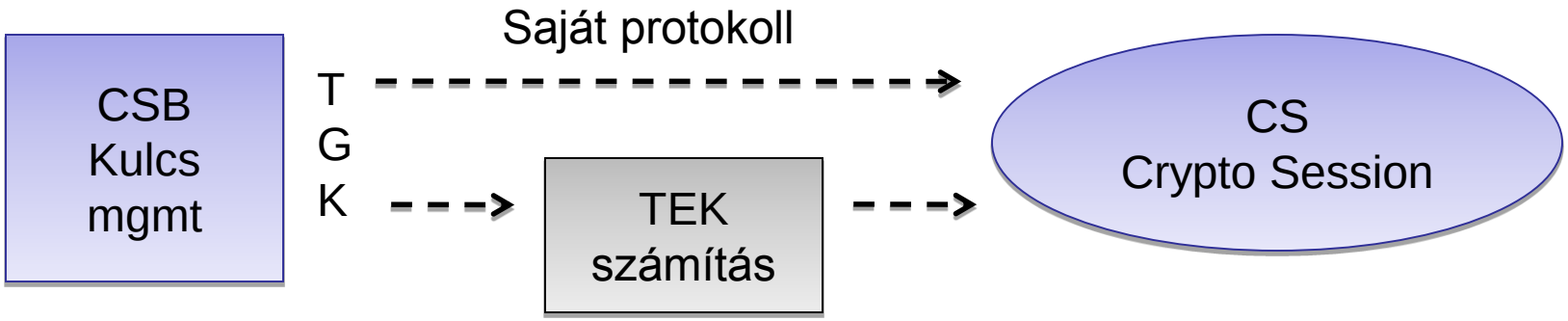
- Az alapértelmezett kulcsszármaztató algoritmus az AES-CTR
  - A szükséges kulcsok a kimenet MSB bitjei
  - Ha szükséges, akkor több kört fut az algoritmus

Multimedia Internet Keying (MIKEY)

- A SRTP működéséhez elengedhetetlen a megfelelő kulcsok kölcsönös ismerete
  - Kulcsmenedzsment protokollok
  - A MIKEY egy kulcsmenedzsment protokoll
- MIKEY (RFC 3830 / 2004) tervezési szempontok
  - End-to-end biztonság
    - Csak a végpontok ismerik a titkot
  - Egyszerűség
  - Hatékonyság
    - Kis sávszélesség, kis számítási teljesítmény, tömör kód, kevés üzenetváltás
  - Integrálhatóság
    - A MIKEY integrálható más protokollokba: SDP és RTSP
  - Nem függ a hálózat egyéb biztonságától

Multimedia Internet Keying (MIKEY)

- Kulcsok generálása
  - CS: Crypto Session
    - Védett kapcsolat
  - CSB: Crypto Session Bundle
    - Több CS összefogása
    - Közös TGK, de saját TEK
  - TEK: Traffic Encryption Key
  - TGK: TEK Generation Key



MIKEY kulcscserék

- Feladatuk a TGK meghatározása
- 3 lehetséges módszer
  - Közös titok
  - Publikus-privát kulcsok
  - Diffie-Hellman kulcscsere

MIKEY kulcscsere – közös titok

- Közös titok segítségével
  - A közös titokból származtatják a TGK-t
  - Skálázhatósági problémák
    - Előre egyeztetni kell a közös titkot
    - Publikus kulcsú módszerekkel egyeztethető a közös titok
  - A leghatékonyabb módszer
    - Csak szimmetrikus titkosítás
    - Rövid üzenetek
  - Kötelező az implementáció
- A skálázhatósági problémák miatt leginkább csak szerver->kliens kapcsolat

MIKEY kulcscsere – közös titok 2.

- A kezdeményező fél küldi a kulcsot a közös titokkal titkosítva
  - HDR: MIKEY fejléc. Jelzi, hogy a kezdeményező kér-e megerősítést -> kölcsönös hitelesítés
  - T: Időbélyeg
    - Véd visszajátszás ellen, illetve a titkosításban is szerepet játszik (IV)
  - RAND: álvéletlen érték a kulcs frissesség miatt (min 128 bit)
  - IDi és IDr az azonosítók
    - Ha ismert vagy számítható, akkor nem szükséges
  - SP: Security Policy: A titkosított kapcsolathoz szükséges paraméterek

HDR, T, RAND, [IDi],[IDr], {SP}, KEMAC

HDR, T, [IDr], V

KEMAC = $E_{\text{titkosító kulcs}}(\{TGK\})$ || MAC

V = MAC

A ttikosító kulcs és a hitelesítő kulcs a közös titokból származik

MIKEY kulcscsere - PKC

- Nyilvános kulcs mód (PKC)
  - Publikus/privát kulcsú titkosítás használata
    - PKI (Public Key Infrastructure) szükséges
  - Skálázható megoldás
  - Erőforrás-igényes
  - Kötelező az implementáció

MIKEY kulcscsere – PKC 2.

- A kezdeményező fél küldi a kulcsot a üzenet kulccsal titkosítva
  - CERTi: A kezdeményező tanúsítványa
  - CHASH: Publikus kulcs kijelölése, amennyiben a több is van
  - PKE: Titkosított üzenet kulcs (envelope key)
    - PKE = E_{PKr}(üzenet kulcs) - Titkosítás a címzett publikus kulcsával
  - SIGNi: Az egész üzenet aláírása

HDR, T, RAND, [IDi|CERTi], [IDr], {SP},
KEMAC, [CHASH], PKE, SIGNi

HDR, T, [IDr], V

KEMAC = $E_{\text{üzenet kulcs}}$(IDi || {TGK}) || MAC
V = MAC

A ttikosító kulcs és a hitelesítő kulcs az üzenet kulcsból származik

Diffie-Hellman kulcsegyeztetés

- Diffie-Hellman kulcsegyeztetés alap
  - p prím és g generator, $2 \leq g \leq p-2$
    - (1) $A \rightarrow B : g^x \bmod p$
    - (2) $A \leftarrow B : g^y \bmod p$
    - x és y véletlen, $1 \leq x, y \leq p-2$
    - A kialakuló kulcs $K = (g^x)^y \bmod p = (g^y)^x \bmod p$
  - Aktív támadás ellen nem véd!
  - Hiányzik belőle a hitelesítés
    - Közbeékelődéses támadás

MIKEY kulcsegyeztetés - DH

- Diffie-Hellman algoritmus hitelesítéssel
  - Két akár ismeretlen fél közös kulcsot hoz létre
  - A kulcsot együttesen hozzák létre, egyik sem irányítja
    - Csoportos kulcscserére alkalmatlan
  - Digitális aláírással hitelesítik a résztvevőket
    - DH érzékeny a közbeékelődéses támadásra
  - Leginkább erőforrás-igényes
- Nem kötelező implementálni

MIKEY kulcscsere – DH 2.

- A kezdeményező fél küldi a kulcsot a közös titokkal titkosítva
  - DHi és DHr a két DH paraméter
    - DHi = g^{xi} mod p
    - DHr = g^{xr} mod p
    - A generátor kulcs kiszámítása: TGK = $g^{xi \cdot xr}$ mod p

HDR, T, RAND, [IDi|CERTi],[IDr],
{SP}, DHi, SIGNi

HDR, T, [IDr|CERTr], IDi, DHr, DHi, SIGNr

Kulcsok leszármaztatása

- PRF – Pseudo Random Function
  - P(s, label, m) = HMAC-SHA1(s, a_1 || label) || HMAC-SHA1(s, a_2 || label) || … HMAC-SHA1(s, a_m || label)
  - outkey = MSB(PRF(inkey, label) = P(s_1, label, m) XOR P(s_2, label, m) XOR … P(s_n, label, m))

  - label: az előállítandó kulcstól függ
  - a_i = HMAC-SHA1(s, a_{i-1}), a_0 = label
  - n: inkey hossza / 256 (az SHA-1 belső méretének a fele), felfelé kerekítve
  - s_i: Az inkey i. blokkja (256 bites blokkok)
  - m: outkey hossza / 160 (az SHA-1 kimenetének hossza), felfelé kerekítve

Label számítása

- TEK generálás:
  - s: TGK
  - label:
    konstans || cs_id || csb_id || RAND
- Konstansok:
  - 0x2AD01C64: TGK
  - 0x1B5C7973: hitelesítő
  - 0x15798CEF: titkosító
  - 0x39A2C14B: salt

- Kulcs generálás kulcscsere közben
  - s: közös titok, vagy üzenet kulcs (PKC esetén)
  - label:
    konstans || 0xff || csb_id || RAND
- Konstansok:
  - 0x150533E1: titkosító
  - 0x2D22AC75: hitelesítő
  - 0x29B88916: salt

MIKEY szállítása

- Az üzeneteket más protokollokba is be lehet építeni
  - Internet telefónia
    - SIP/SDP
  - Video on Demand
    - RTSP

- MIKEY integrálsása SDP és RTSP protokollba (RFC 4567 / 2006)

```
⊞ Media Attribute (a): crypto:1 AES_CM_128_HMAC_SHA1_80 inline:1Ppc7iQYrJ7xUiY/lNIXIcO2ELx+o2iB5NfbJs8m
⊞ Media Attribute (a): crypto:2 AES_CM_128_HMAC_SHA1_32 inline:+fY6XR7tfHeIo6eX49cZ5bDqEvE3iuaO0D6SReui
```

MIKEY és SRTP alkalmazása

- SIP protokoll

SIP Invite + SDP(MIKEY)

SIP Ringing

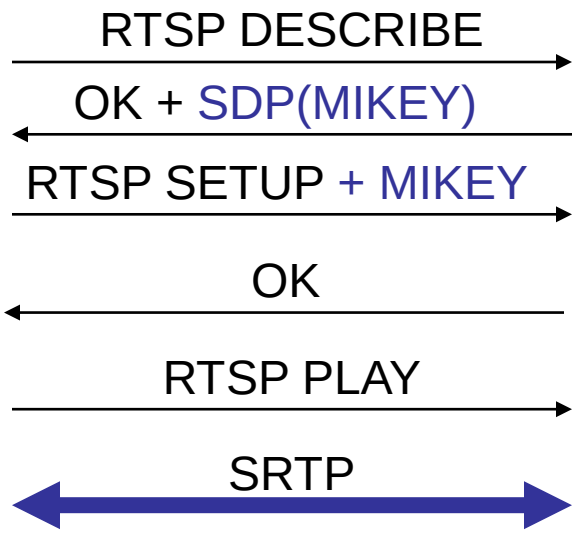
SIP OK + SDP(MIKEY)

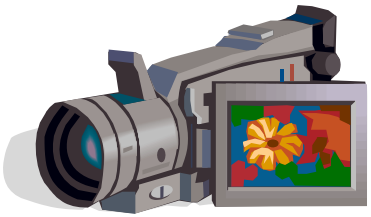
SIP Ack

SRTP

MIKEY és SRTP alkalmazása

- RTSP protokoll
  - Hálózati távirányító





ZRTP

- ZRTP: Media Path Key Agreement for Unicast Secure RTP
  - Philip R. Zimmermann (PGP is hozzá fűződik)
  - RFC 6189 (2011 április)

- Kulcs egyeztetés SRTP hang kapcsolat számára
  - A kulcs egyeztetése az RTP kapcsolaton belül történik.
    - Létezik SDP a ZRTP-hez, de ha nincs, akkor is fel lehet építeni a kapcsolatot

```
⊟ Media Attribute (a): zrtp-hash:1.10 a158c2ab3f51860e98dd593e1565509ee3d560927e070f6d3aeeead158eeeeb2
    Media Attribute Fieldname: zrtp-hash
    Media Attribute Value: 1.10 a158c2ab3f51860e98dd593e1565509ee3d560927e070f6d3aeeead158eeeeb2
```

  - Független a jelzésprotokolltól
  - Saját RTP payload típus

ZRTP kulcscsere

- Kulcscserék
  - DH kulcscsere, így nem függ PKI-tól (De MiTM veszély!)
  - Preshared mode, egy előző viszony közös titkát használva
    - Key continuity (Sikeres felépítés esetén tárolja a kialakult kulcsot, hogy később felhasználja hitelesítéshez)
  - Multistream mode, több RTP kapcsolat osztozik a közös kulcson
  - De lehet PKI is

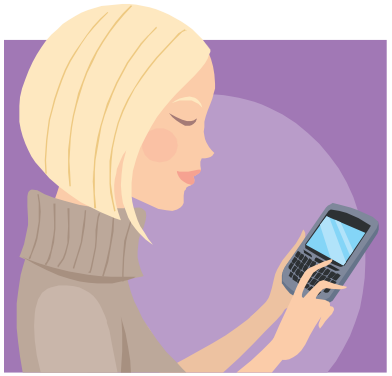
- Védelem a közbeékelődés ellen: a másik fél hangját kell ismerni
  - Short Authentication String (SAS)
    - Származása: üzenet tartalmak hash értékei + ZRTP végpont azonosítók
    - 16 bites: PGP szólistából 2 szó
    - 20 bites: base32 kódolás
  - Hash commitment
    - A kezdeményező előre kiszámolja és elküldi publikus DH érték hasht. A támadó így nem tudja azt megváltoztatni. A támadó még rövid SAS mellett sem lehet hatékony
    - Elküldi a másik fél korábbi Hello üzenet hash-ét is
      - hvi = hash(initiator's DHPart2 message || responder's Hello message)

- Támadható, de csak nagy költségekkel

ZRTP működése

- Hello és HelloACK üzenetek mindkét féltől
  - Benne: ZID: 96 bites egyedi azonosító, ami telepítéskor generálódik, egyeztetés a közös paraméterekben.
  - Ha van integritásvédelem, akkor küldhet egy üzenet hasht, ami további védelmet nyújthat
  - A hash értékek összekötik az üzeneteket, a hamis ZRTP üzenetek kiszűrhetőek
- ZRTP Commit
  - Ha megegyeztek az algoritmusokban, indul a kulcscsere. Ezt bármikor indíthatják („Go Secure”). Bármelyik fél indíthatja.
  - Már tartalmazza a jövőbeli DHpart2 és múltbéli Hello üzenetek hash értékét. (Hash commitment)
- DHpart1 és DHpart2
  - DH paraméterek cseréje. Ha már volt korábbi kapcsolat, akkor az abból származó kulcsok hash-eit is elküldjük.
- SAS számítás és egyeztetés (ha nincsen korábbi kulcs)
  - Egyeztetés: ZRTP protokollon kívül, szóban
  - SAS értéke az eddigi üzenetek hash étékeiből valamint a ZRTP azonosítókból. 32 bit hosszú, ebből 16 vagy 20 bitet használnak.
- Confirm
  - Egyeztetés befejezése, opcionális digitális aláírás (nem csak PKI alapon!), ha nem szöveges SAS van
- Conf2ACK vagy első érvényes SRTP üzenet

ZRTP működése



Hello (ver, opt, A ZIP)

HelloACK

Hello (ver, opt, B ZIP)

HelloACK

Commit (B ZID, opt, hash)

DHpart1 (pvr, sec hash)

DHpart2 (pvi, sec hash)

Hash commitment

Itt „B” a kezdeményező

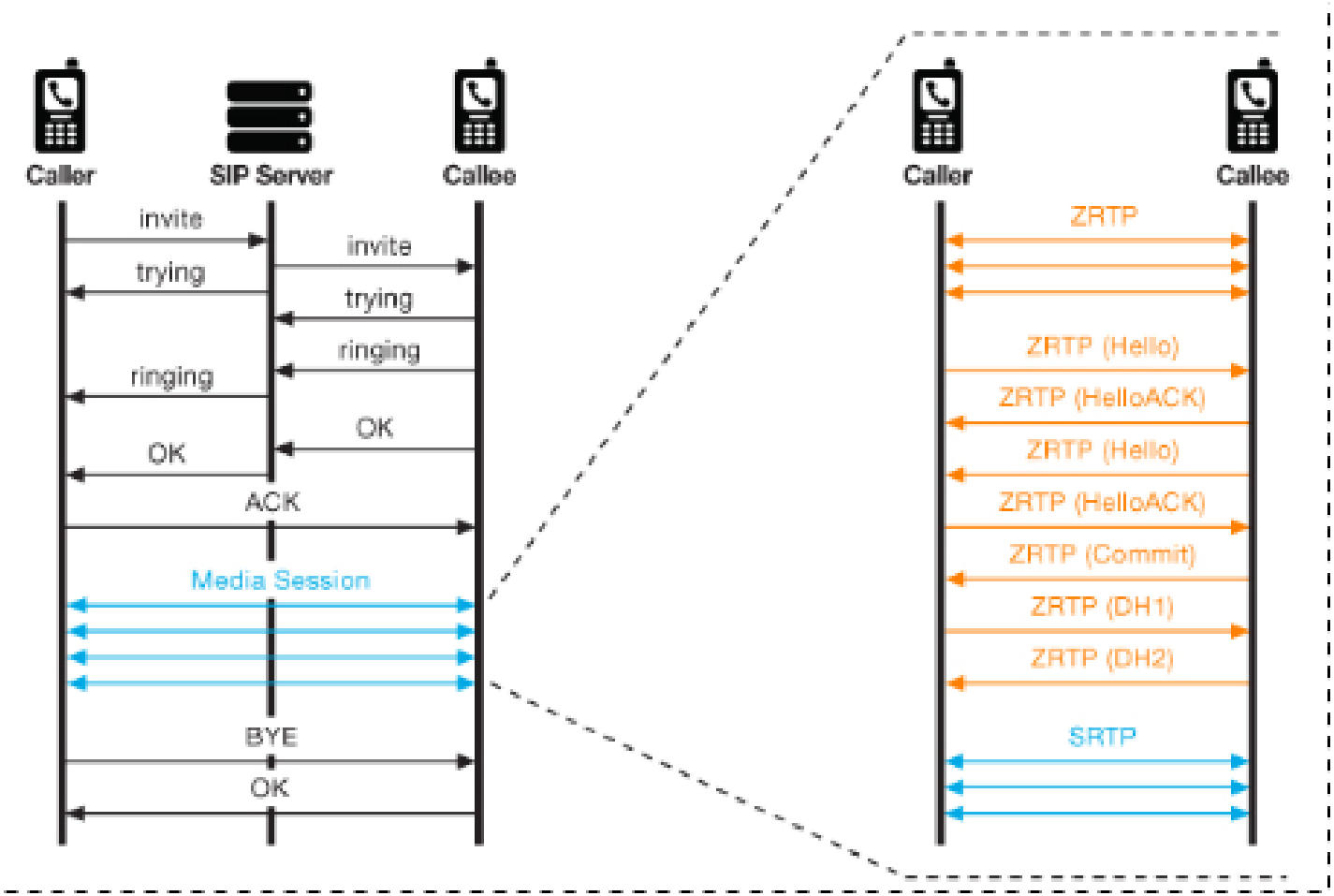
SAS ellenőrzés

Confirm1 (MAC, flags, *sig*)

Confirm2 (MAC, flags, *sig*)

Conf2ACK

ZRTP és SIP


Caller
SIP Server
Callee
invite
trying
invite
trying
ringing
ringing
OK
OK
ACK
Media Session
BYE
OK
Caller
Callee
ZRTP
ZRTP (Hello)
ZRTP (HelloACK)
ZRTP (Hello)
ZRTP (HelloACK)
ZRTP (Commit)
ZRTP (DH1)
ZRTP (DH2)
SRTP