Multimedia Security

Media Access Security

jean.lefeuvre@telecom-paristech.fr

Transport Security

- **Confidentiality**
  - Cipher algorithms
  - Key distribution
- **Authentication**
  - Client has to authenticate server
  - Server may require client to authenticate
- **Integrity**
  - Detect data modifications
- **Protection against attacks**
  - Replay
  - DoS/DDoD
  - Man-in-the-middle
- **Performances:**
  - Bitrate overhead
  - Error propagation
  - Computing Times

AAA Protocols

- **Authentication**
  - Check identity of user/device (both client and server)
- **Authorization**
  - What are the rights given to the user ?
- **Accounting**
  - Gather Information during the session
  - Logs for billing
  - Traffic statistics
  - …
- **Examples**
  - RADIUS
    - Remote Authentication Dial In User Service
    - IETF, RFC 2865
  - DIAMETER
    - IETF, RFC 3588 …
    - Used in 3GPP IMS

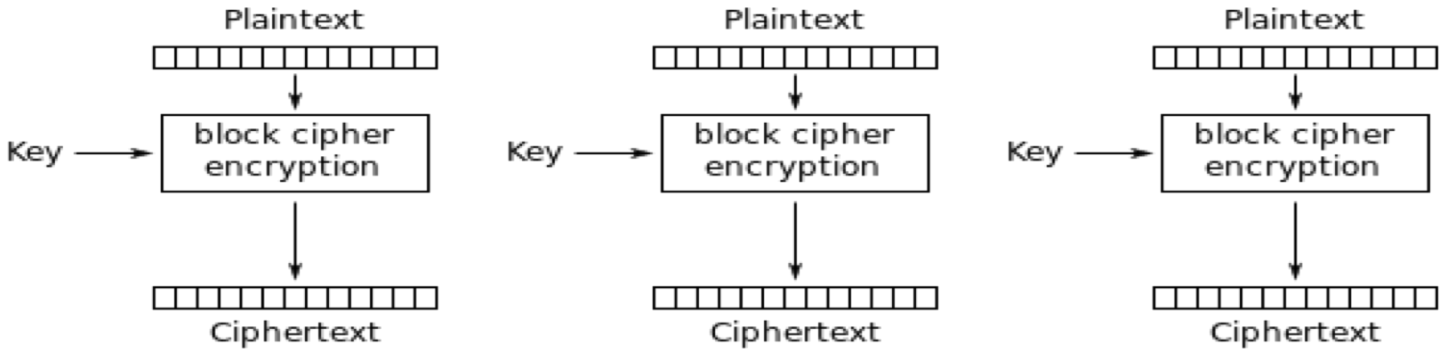
Cipher types (refresher)

- **Symmetrical Cipher**
  - Same key used by both peers in the communication
  - Advantage: robust, quite fast
  - Disadvantage: key needs to **be shared**
- **Asymmetrical Cipher**
  - Each client C_i has a unique pair of keys (K_i, P_i) such that
    - $MSG = P_i(K_i(MSG)) = K_i(P_i(MSG))$
    - $K_i(K_i(MSG)) \neq MSG$ and $P_i(P_i(MSG)) \neq MSG$
    - Usually called *public and private keys*
  - Advantage
    - If one key is kept secret, successful decipher validates origin
    - Possibility to cipher
      - for all peers (using one's private key)
      - For a given peer (using its public key)

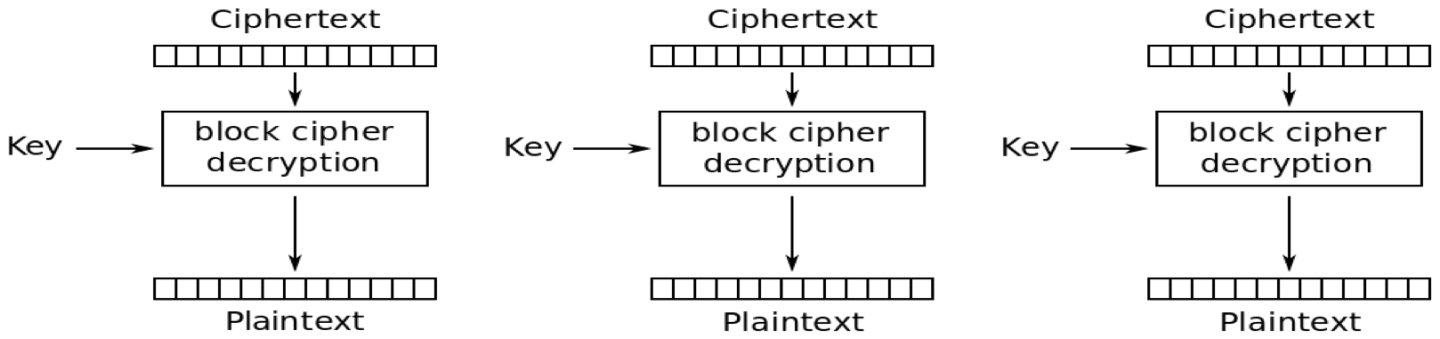
AES: Advanced Encryption Standard

- **Characteristics**
  - 128, 192 or 256 bit bloc cipher
  - Symmetrical cypher (key must be shared)
  - Used almost everywhere (TLS, IPsec, media protection)
    - Hardware acceleration in intel core i7
- **Mode of operation or “chaining modes”**
  - Block ciphers encrypts only K bits
    - If several identical input K-bit patterns, attack can be performed by analyzing output
  - Solutions
    - Use random number to XOR the first bloc
      - Initialization Vector (IV) or Start Variable (SV)
    - chain the output of a bloc to the input of the next
    - Mix of both

No chaining: Electronic Codebook (ECB)



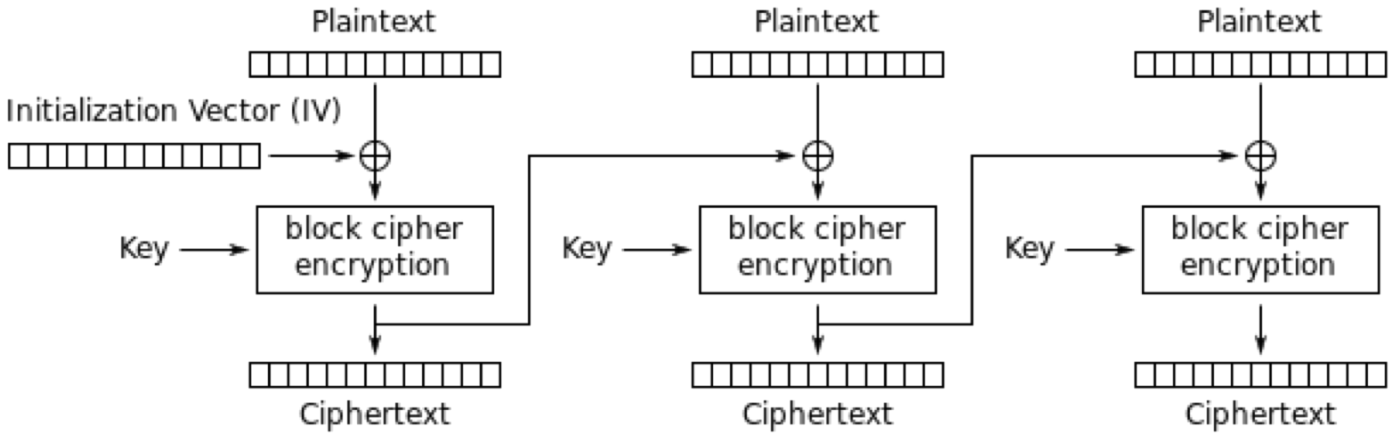
Electronic Codebook (ECB) mode encryption



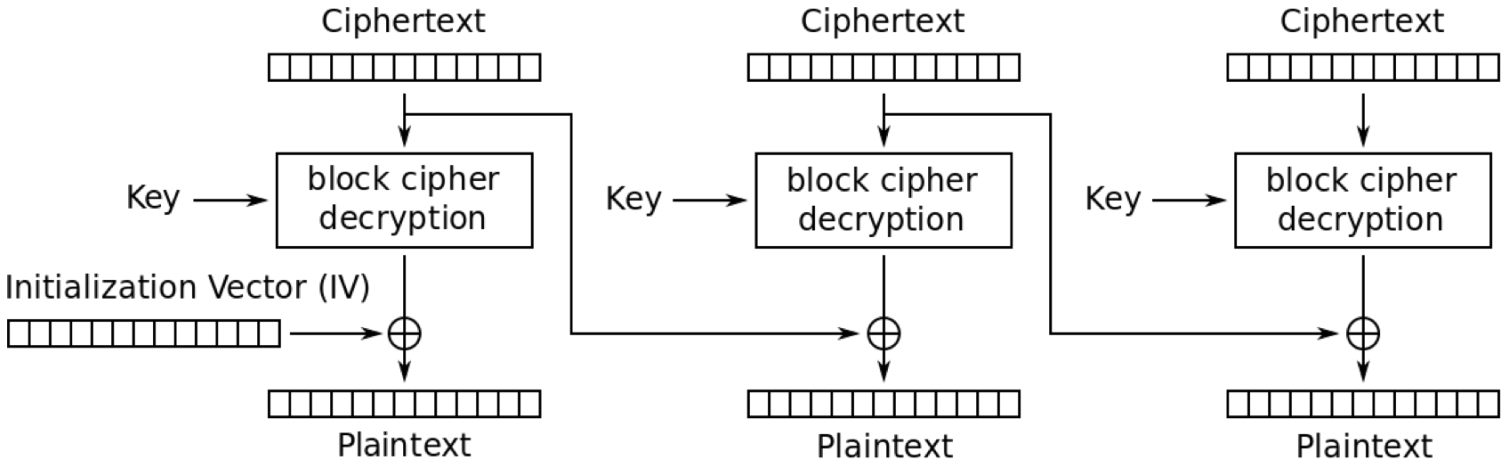
Electronic Codebook (ECB) mode decryption

© Wikipedia

Cypher Block Chaining: CBC



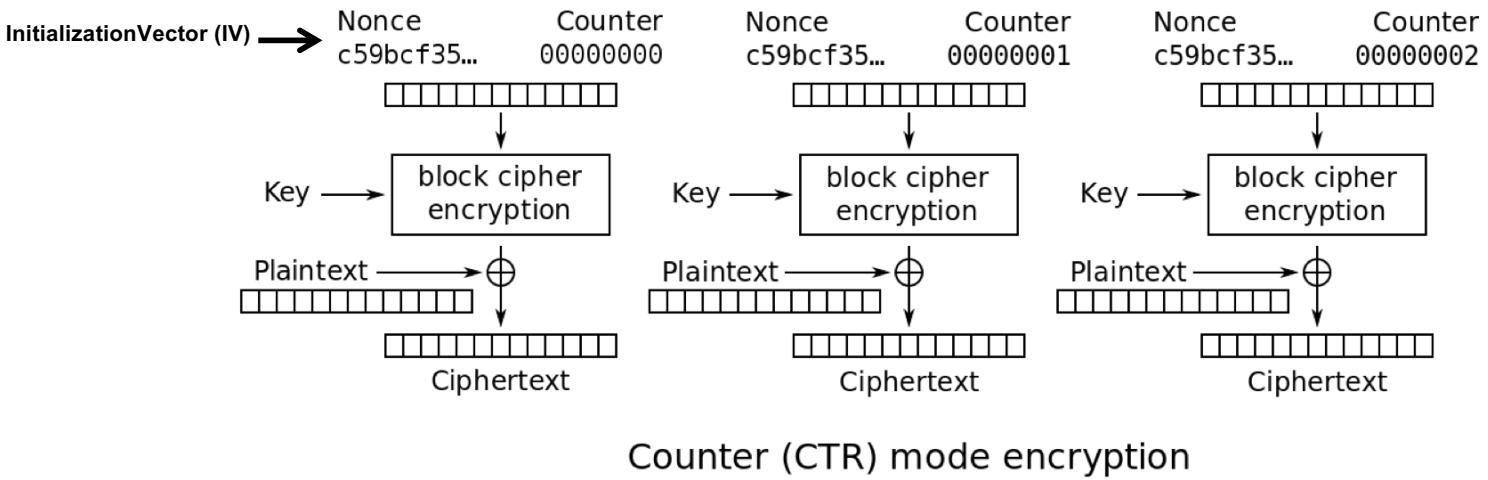
Cipher Block Chaining (CBC) mode encryption



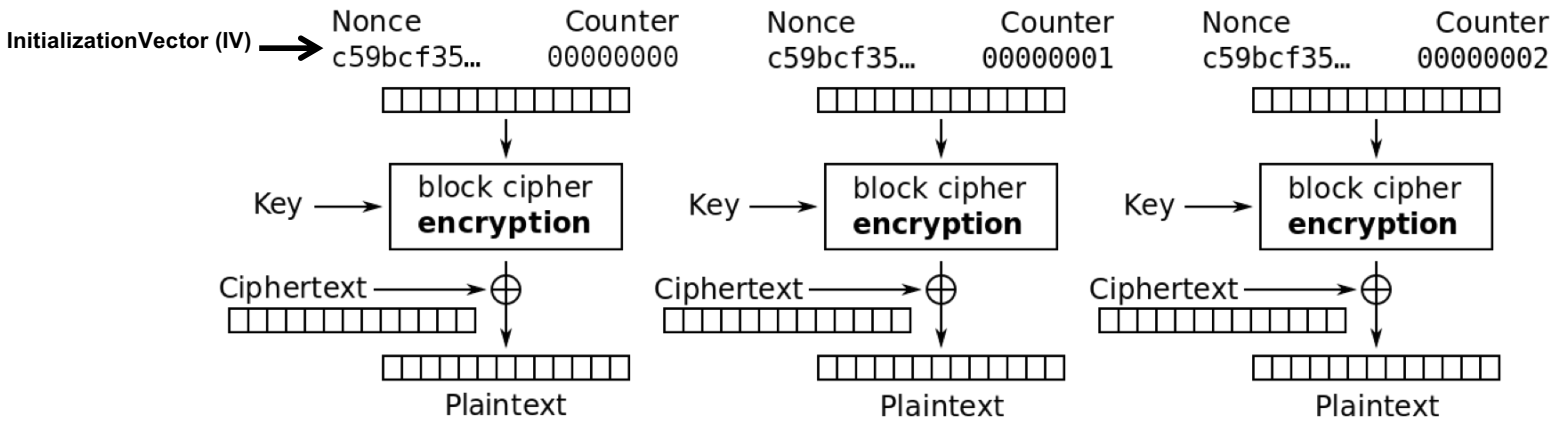
Cipher Block Chaining (CBC) mode decryption

© Wikipedia

Stream Counter mode: CTR



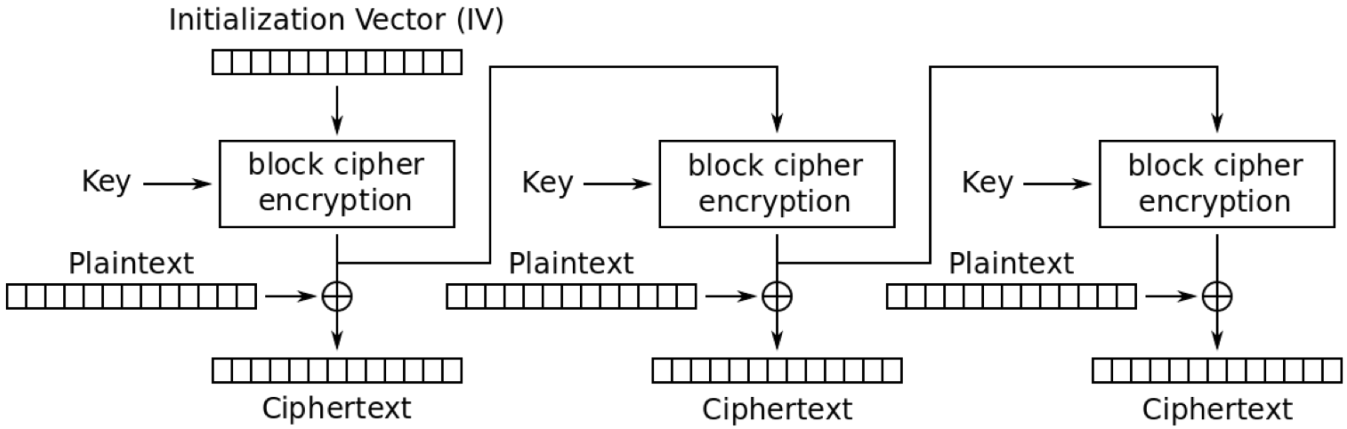
Counter (CTR) mode encryption



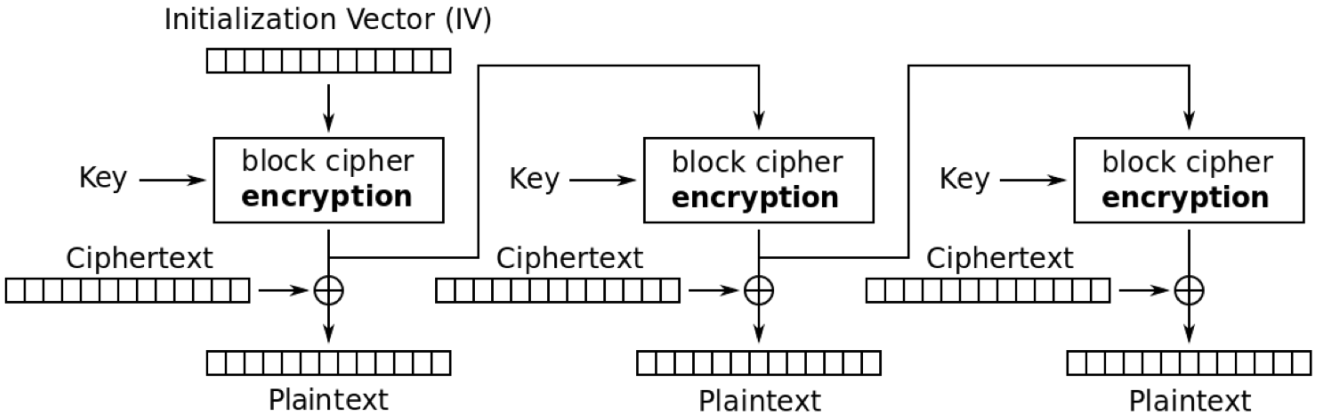
Counter (CTR) mode decryption

© Wikipedia

Output feedback mode: OFB or FB



Output Feedback (OFB) mode encryption



Output Feedback (OFB) mode decryption

- **Symmetrical**
  - Encryption⇔ decryption

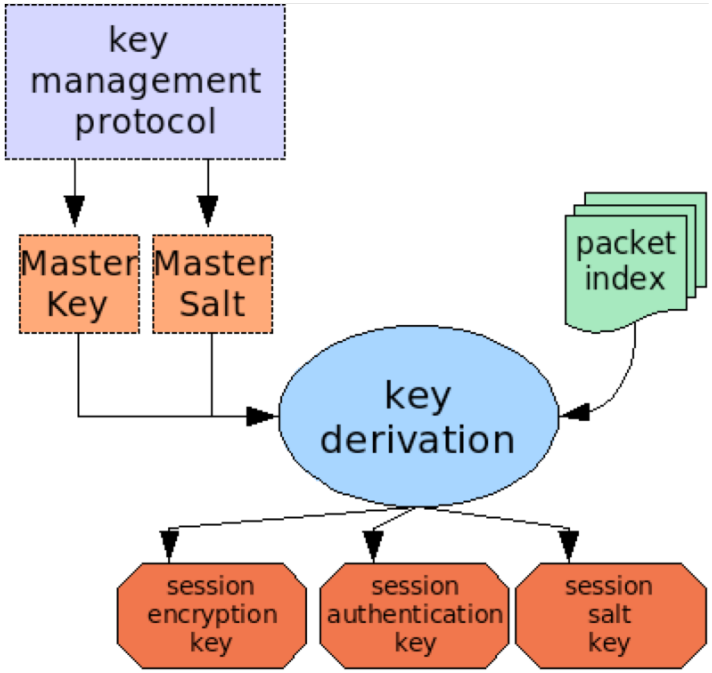
© Wikipedia

Padding

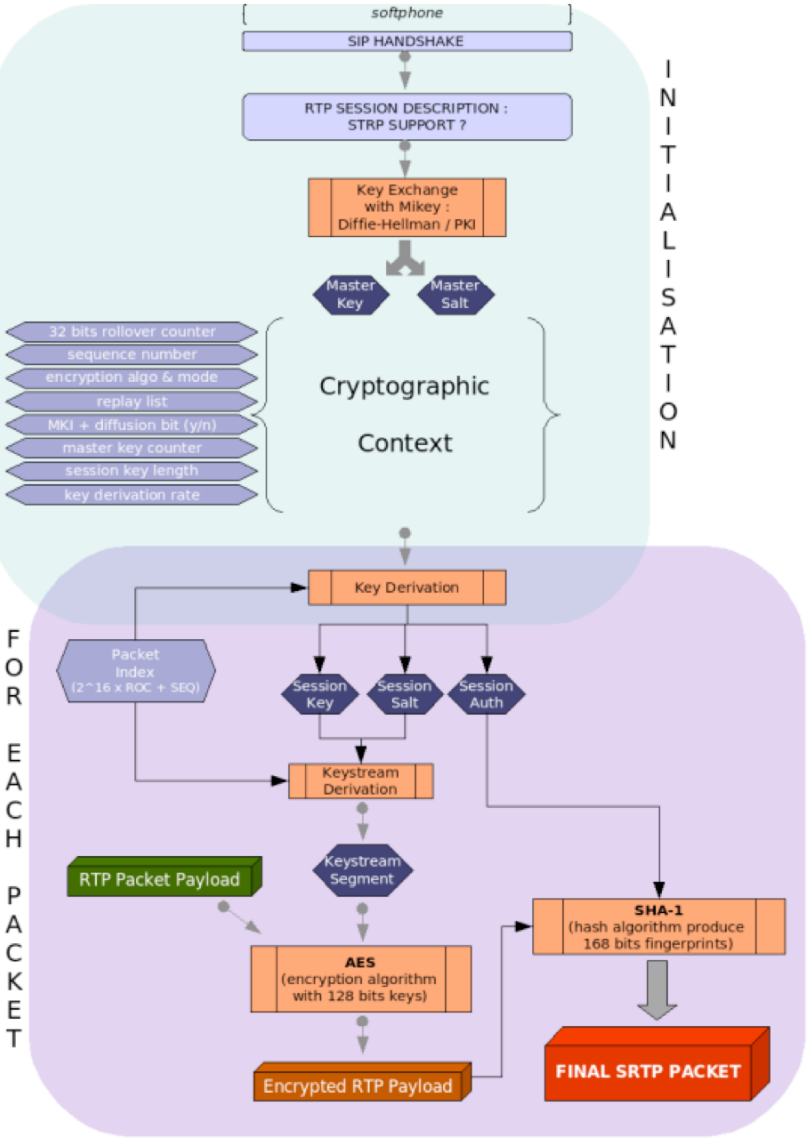
- **Ciphers work on K bits**
  - If plaintext is the input to the cipher, need N*K-bits
    - CBC
  - If plaintext is XORed with output of cipher, no padding needed
- **Padding strategies with CBC**
  - Signal number of n padded bytes
    - No signaling
      - Set remaining bits to 0 in the last block
      - simple but not always convenient if no framing (size indication) in the plaintext
    - Pad the last block with n bytes of value n
      - May require an extra block to signal plaintext was multiple of K!
  - No padding
    - Only cipher a multiple of K bits, leaving (K/8 – n) bytes in the clear

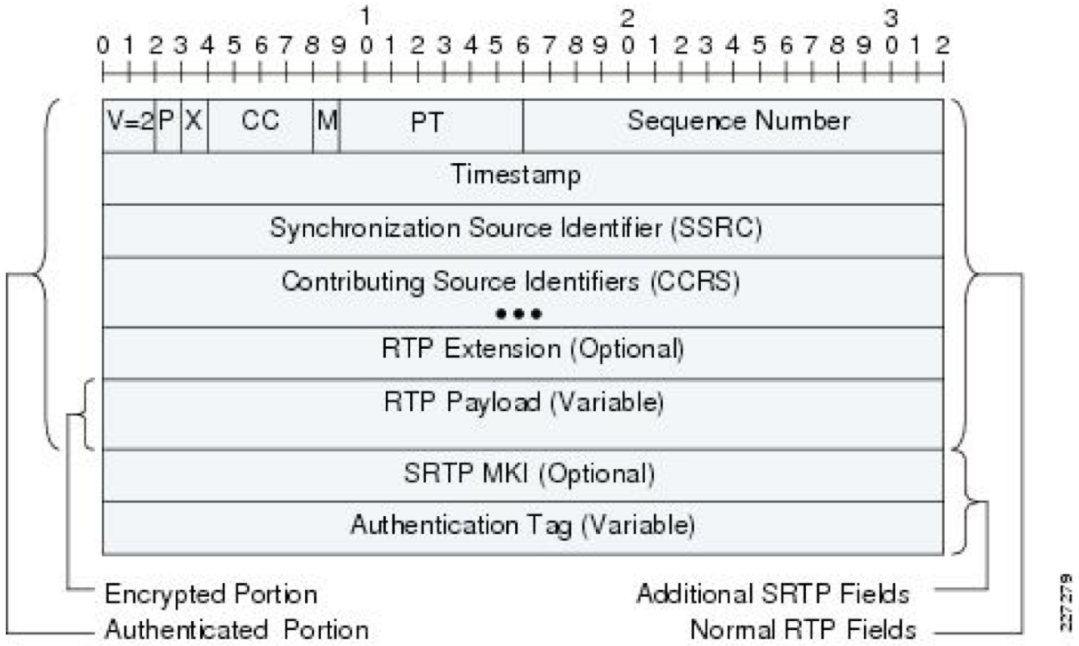
Secure RTP

- **SRTP**
  - RFC 3711
  - Confidentiality and authentication for RTP and RTCP
- **Confidentiality**
  - Ciphering of SRTCP and SRTP (optional)
    - AES-128 mode CBC or OFB
    - NULL-cipher (no ciphering)
  - No payload expansion during ciphering
  - Header not encrypted
    - allows header compression techniques
  - Key change through
    - MKI: Master Key Index
    - From-To: pre-computed key changes
- **Integrity**
  - Authentication of packets
    - Mandatory for SRTCP, optional for SRTP
    - HMAC-SHA1 over payload and some headers
      - including packet seq number
    - 10 or 4 bytes long
  - Protection against replay
    - HMAC of SRTP
    - Internal list of received packets
- **Key exchange**
  - Out of scope
    - VoIP: often using SIPS (SIP over TLS)
  - One Master Key is exchanged
    - Derivation of SRTP and SRTCP session key, session salt and auth key



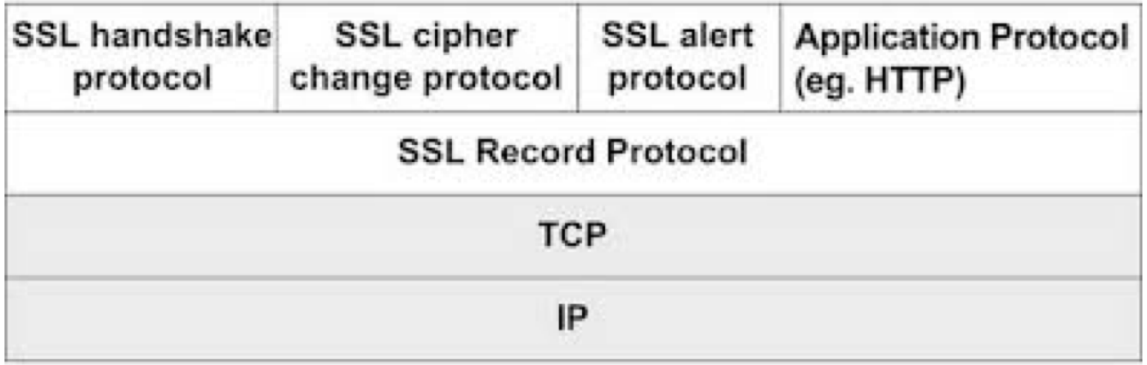
Secure RTP


softphone
SIP HANDSHAKE
RTP SESSION DESCRIPTION : STRP SUPPORT ?
Key Exchange with Mikey : Diffie-Hellman / PKI
Master Key
Master Salt
32 bits rollover counter
sequence number
encryption algo & mode
replay list
MKI + diffusion bit (y/n)
master key counter
session key length
key derivation rate
Cryptographic Context
INITIALISATION
Key Derivation
Packet Index (2^16 x ROC + SEQ)
Session Key
Session Salt
Session Auth
Keystream Derivation
Keystream Segment
RTP Packet Payload
AES (encryption algorithm with 128 bits keys)
SHA-1 (hash algorithm produce 168 bits fingerprints)
Encrypted RTP Payload
FINAL SRTP PACKET
FOR EACH PACKET


0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2
V=2
P
X
CC
M
PT
Sequence Number
Timestamp
Synchronization Source Identifier (SSRC)
Contributing Source Identifiers (CCRS)
RTP Extension (Optional)
RTP Payload (Variable)
SRTP MKI (Optional)
Authentication Tag (Variable)
Encrypted Portion
Authenticated Portion
Additional SRTP Fields
Normal RTP Fields
227279

HTTPS and TLS

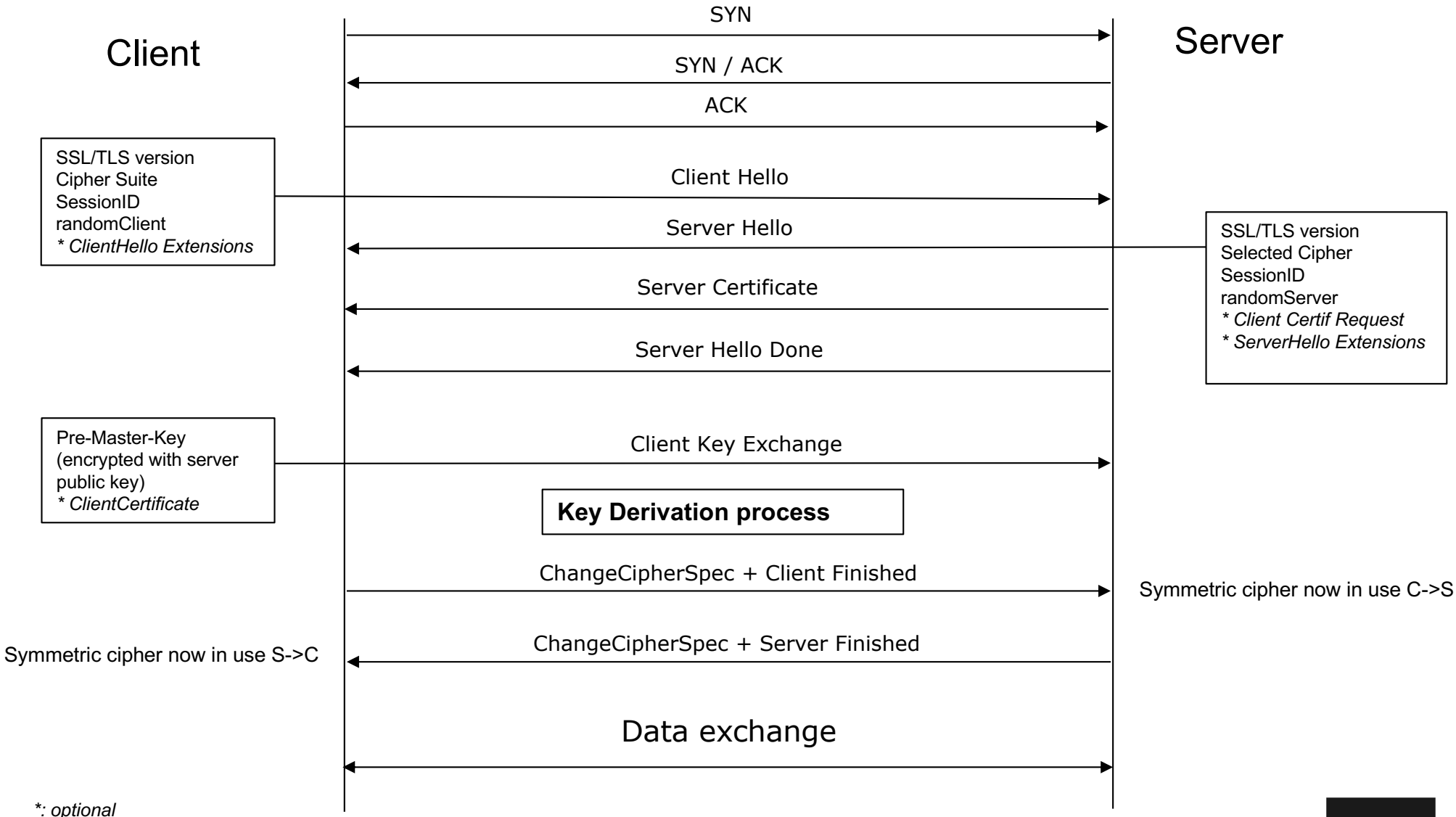
- **TLS**
  - Transport Layer Security
    - Previously SSL (Secure Socket Layer)
  - Over TCP
  - Two protocols
    - Handshake protocol: initial setup
    - Record protocol: during session
- **Features**
  - Privacy
    - Symmetrical ciphers
  - Authentication
    - Asymmetrical ciphers
    - Usage of server certificates to authenticate public key
  - Integrity
    - Authentication hash (HMAC-SHA1) in messages
  - Key distribution
  - Reusable across different connections
    - SessionID
- **CipherSuite**
  - List of options for
    - Key exchange
    - Cipher (AES, RC4, RC2, DES, 3DES, IDEA, Fortezza)
    - CipherType (stream ou block)
    - MAC (MD5 ou SHA1)



TLS Keys

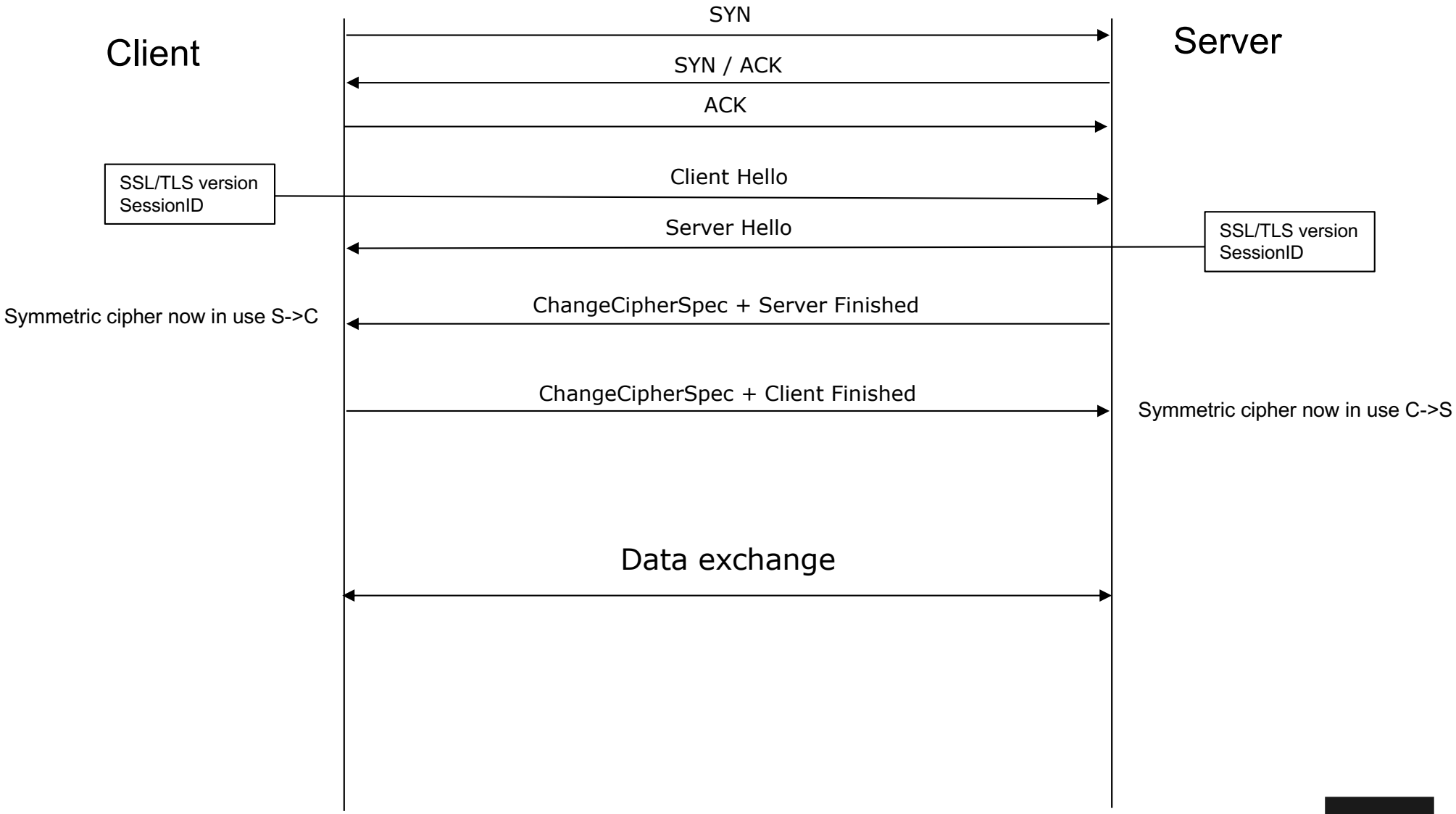
- **Derived from**
  - Random values picked by client and server
  - *pre_master_key* computed by client
    - Send encrypted using server's certificate
- **Authentication Keys**
  - 1 for Client , 1 for Server
  - Used for message integrity
- **Cipher keys**
  - 1 key and 1 IV for client
  - 1 key and 1 IV for server
  - Each peer uses its own key for ciphering messages it sends

TLS setup

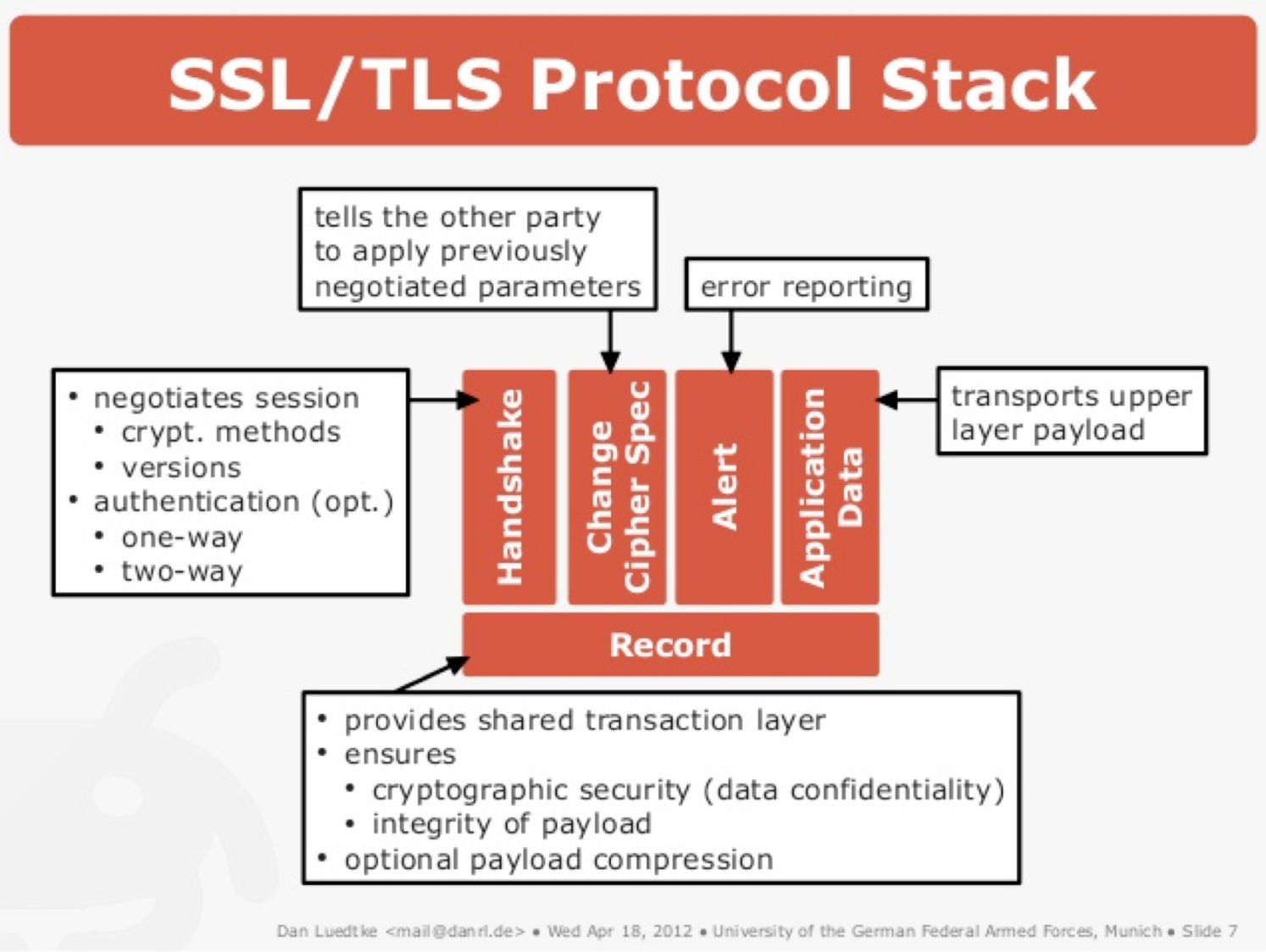


*: optional

TLS resume


Client
Server
SYN
SYN / ACK
ACK
SSL/TLS version
SessionID
Client Hello
Server Hello
SSL/TLS version
SessionID
Symmetric cipher now in use S->C
ChangeCipherSpec + Server Finished
ChangeCipherSpec + Client Finished
Symmetric cipher now in use C->S
Data exchange

TLS Record Protocol

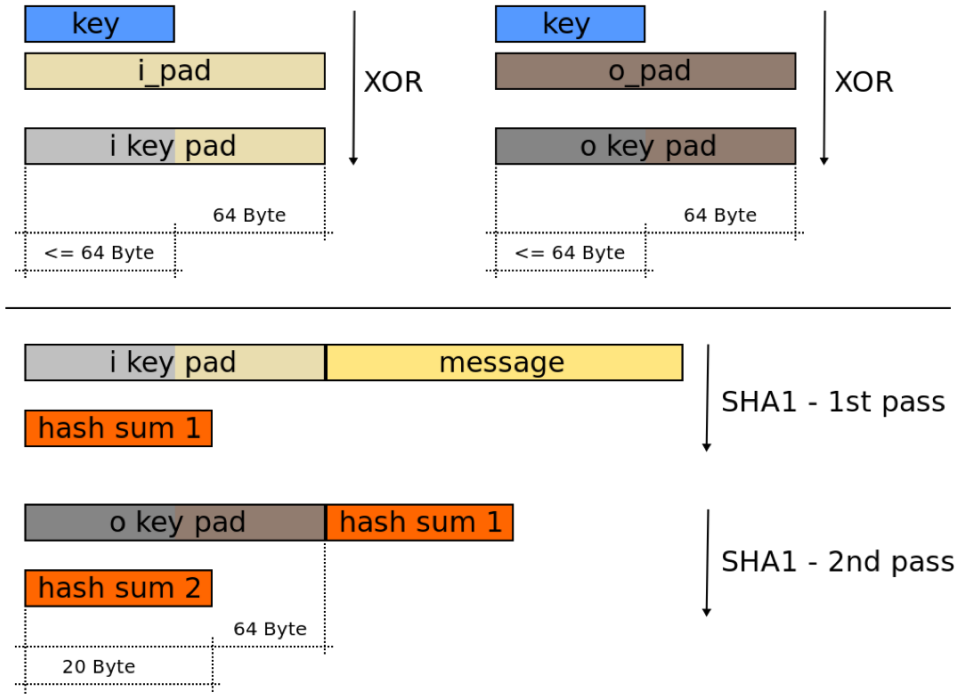


TLS message authentication

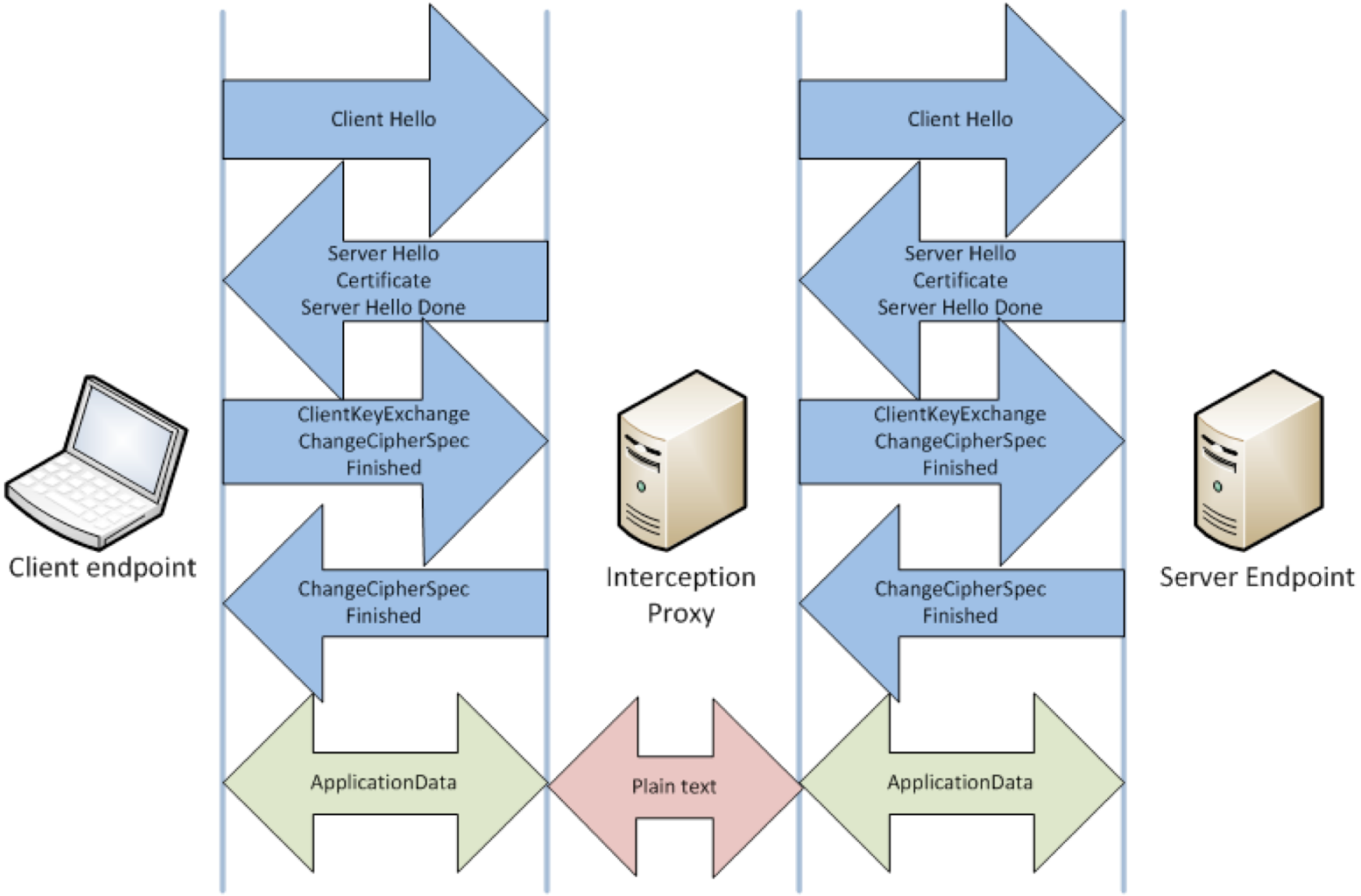
- **Key-HASHed Message Authentication Code**
  - HMAC-MD5 ou HMAC-SHA1

$$HMAC(K, m) = H\Big((K \oplus opad) \;||\; H\big((K \oplus ipad) \;||\; m\big)\Big)$$

  - K key, m message, || concatenation, $\oplus$ XOR
  - opad=0x36, ipad=0x5C

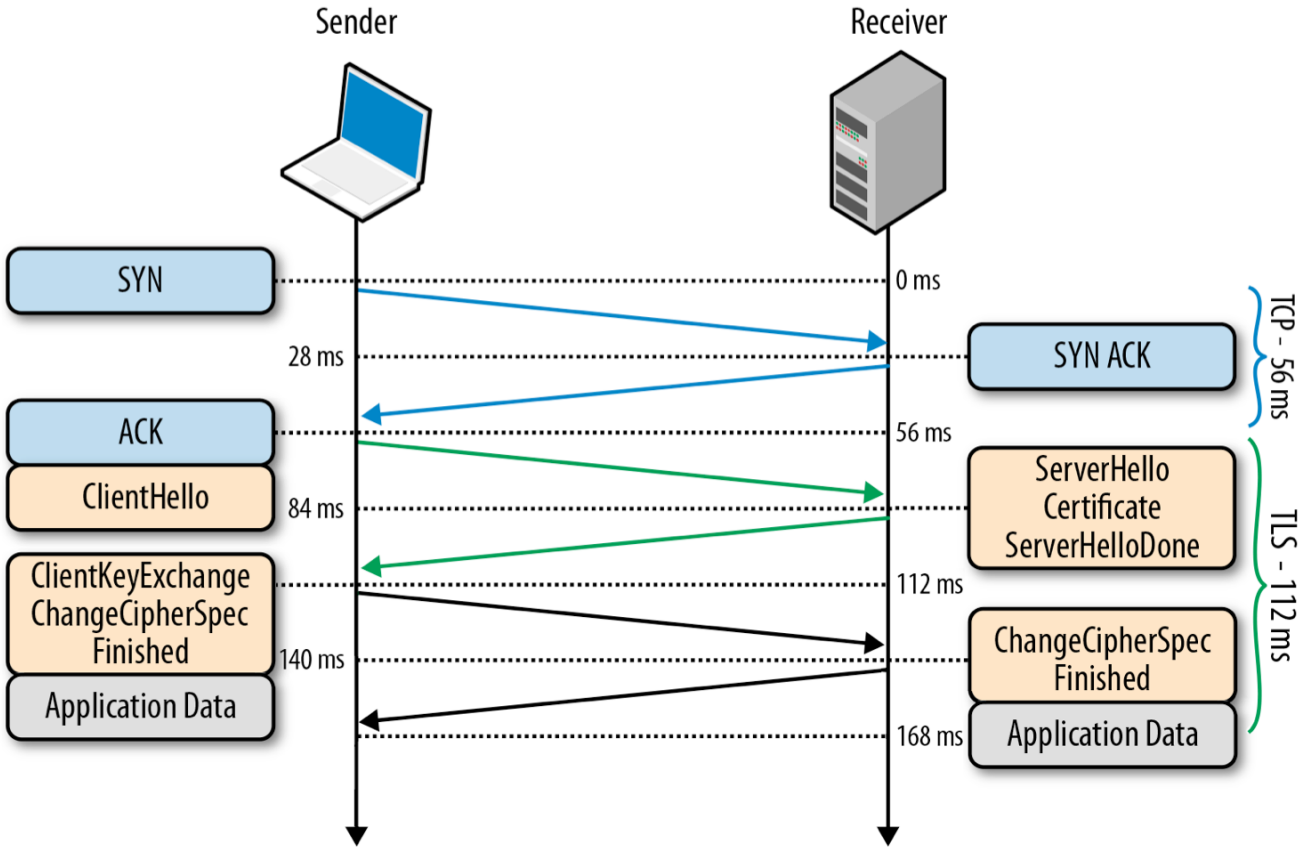


TLS and Man-in-the-middle



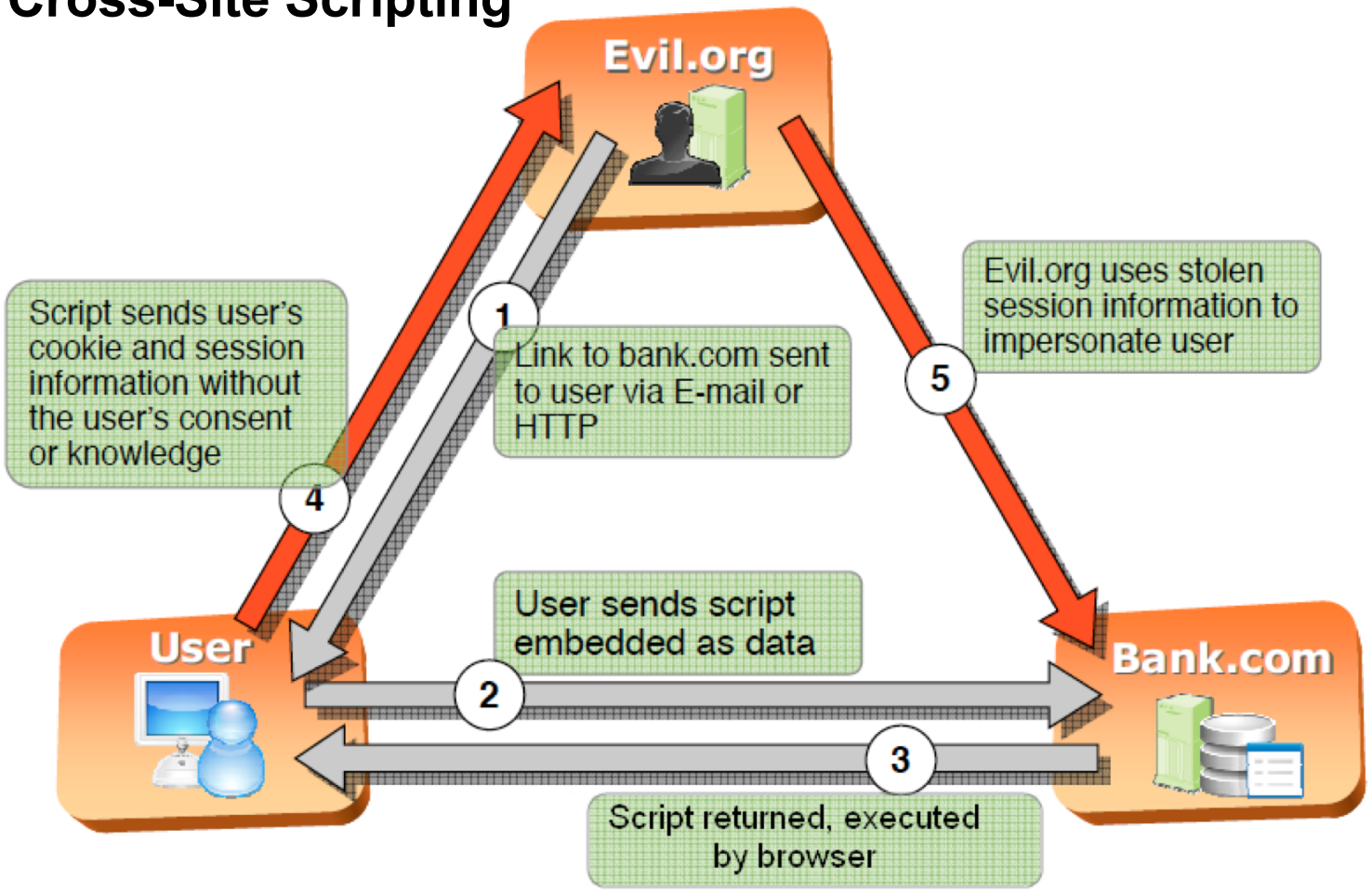
TLS and latency

- **Additional RTTs**
  - 1 for Client Hello / Server Hello
  - 1 for client finished / sever finished
  - Opt, 1 or 2 for certificate exchange
- **TLS False Start**
  - Embed HTTP (or other) client request in 2nd exchange



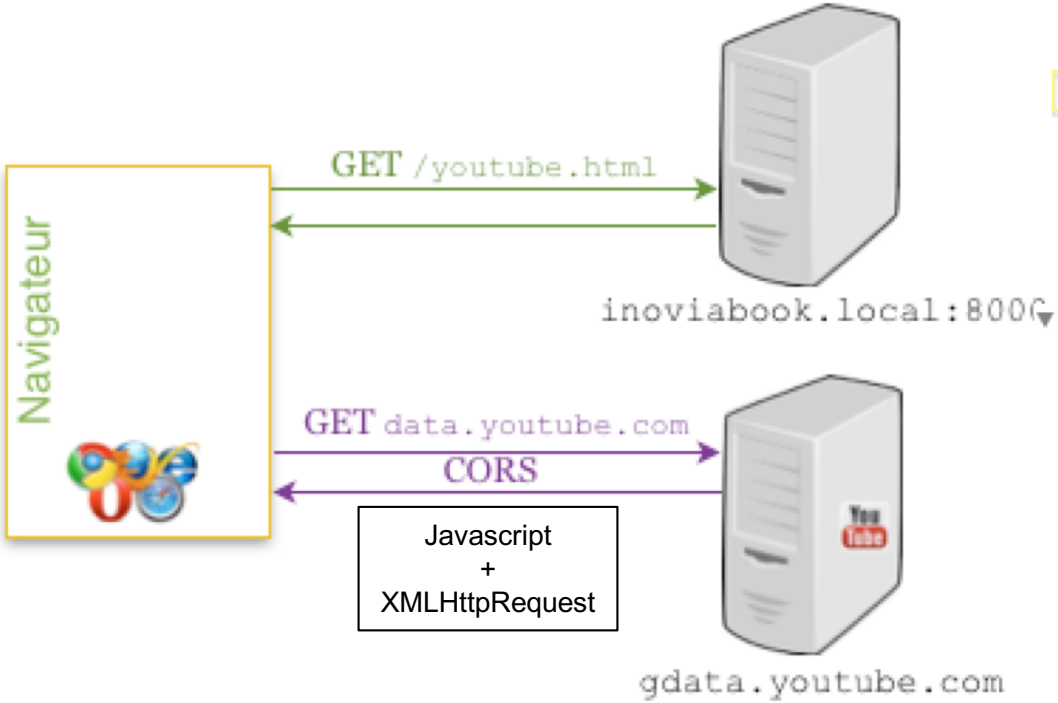
Web security

- Cross-Site Scripting



Web Security

- **HTTP Cross-origin resource sharing**
  - Server A explicitly allows client to use its data within JavaScript downloaded from server B
  - Impact on HTTP Streaming
    - JS based players
    - Players are usually distributed through different servers than media



▼Request Headers

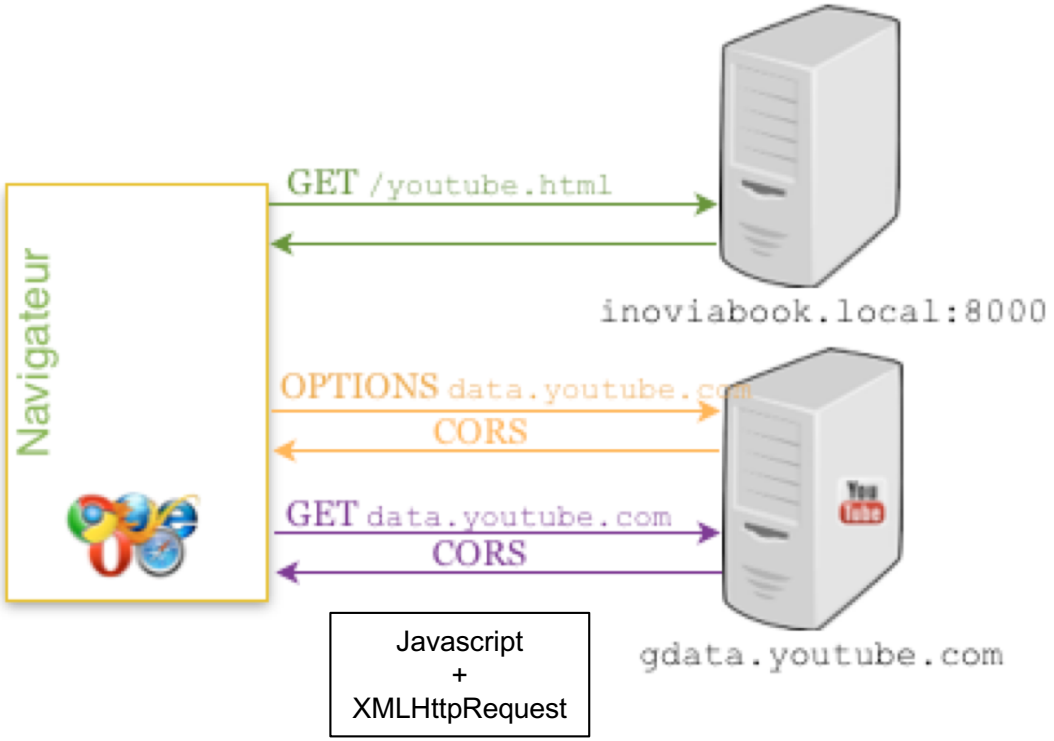
:authority: gdata.youtube.com
:method: GET
:path: /feeds/api/standardfeeds/most_popular?v=2&alt=json
:scheme: https
accept: application/json, text/javascript, */*; q=0.01
accept-encoding: gzip, deflate, sdch
accept-language: fr-FR,fr;q=0.8,en-US;q=0.6,en;q=0.4
cache-control: no-cache
origin: http://inoviabook.local:8000
pragma: no-cache
referer: http://inoviabook.local:8000/youtube.html
user-agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4)

▼Response Headers

access-control-allow-origin: http://inoviabook.local:8000
access-control-expose-headers: Cache-Control,Content-Encoding,
cache-control: private, max-age=300, no-transform
content-encoding: gzip
content-type: application/json; charset=UTF-8
date: Wed, 15 Apr 2015 14:58:58 GMT
etag: W/"A04HSXg7cCp7I2A9XRRaEU0."
expires: Wed, 15 Apr 2015 14:58:58 GMT

CORS and preflight

- **Preflight goal**
  - Detect old servers without cross-origin awareness
    - Will complain about “OPTION” request
  - Find global policies for a domain
  - Impact on session
    - One additional RTT
    - Can be cached




```
OPTIONS
/feed/api/most_popular?v=2&alt=json
HTTP/1.1
Host: gdata.youtube.com
origin: http://inoviabook.local:8000
access-control-request-method: GET
access-control-request-headers:
accept, content-type
```

Multimedia Security

Media Content Protection

Rights Management

- **Problematic**
  - Ensure content is protected outside of transport layer
  - Prevent content redistribution
    - Lock the content (Secure storage)
    - Watermark the content (identification of leaks)
- **Rights Management**
  - Distribution
  - Identification of rights and content
  - Secure framework for rights requests
- **Rights Exploitation**
  - Content Access Control
  - Content (re)distribution Control

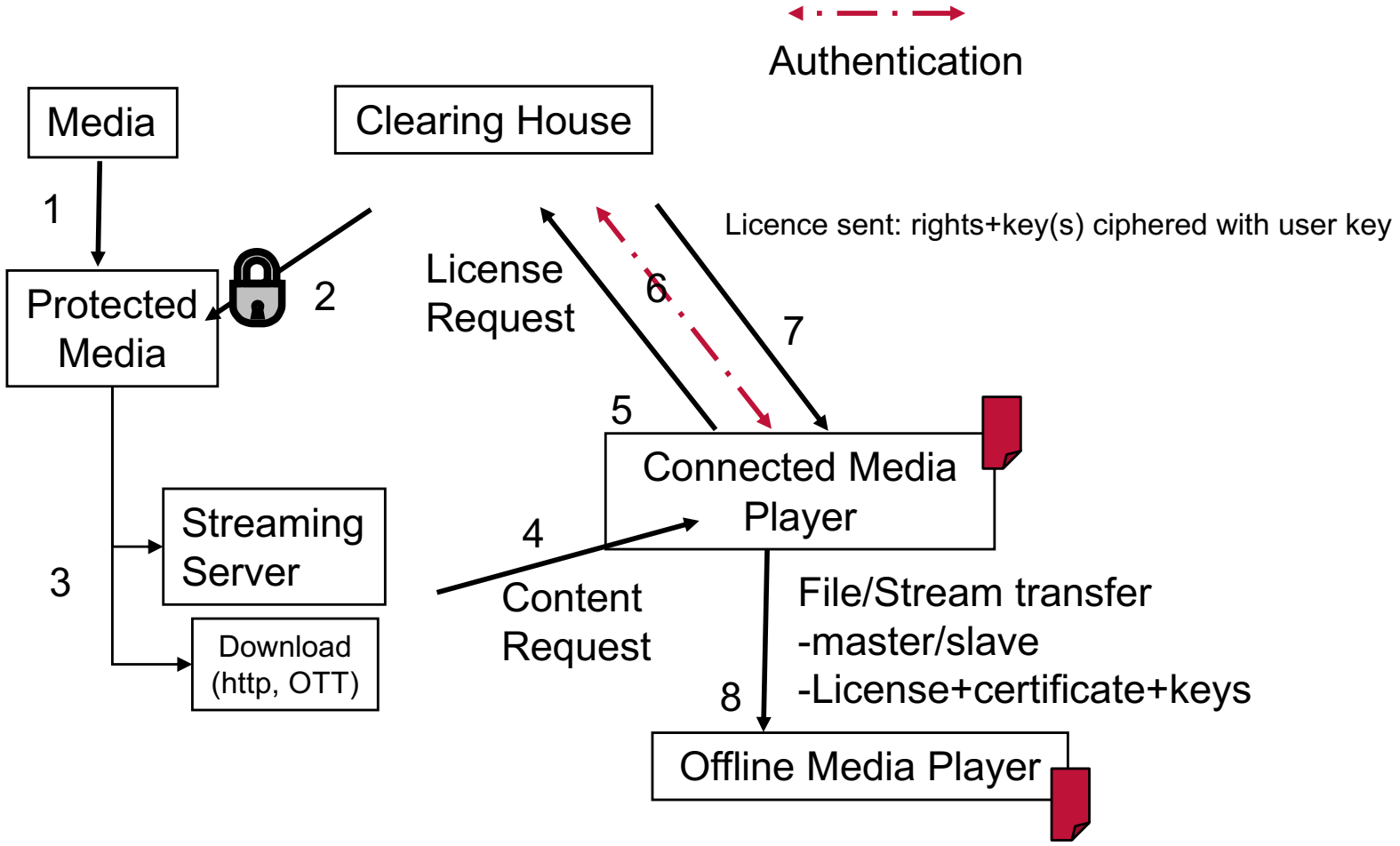
Who is involved ?

- Content owners
  - Universal, TF1, Disney, …
- Rights Owner
  - Producers, artists, authors, compositors, …
- Standardization Bodies
  - 3GPP, OMA, MPEG, OSI, …
- Industry Group
  - Qualcomm, Intel, Sony, Technicolor, Sagem, Apple, Philips, …
  - Integrate DRM solutions in CE devices in trusted way
- DRM Technical Solution Providers
  - Microsoft (PlayReady), Apple (FairPlay), Google (Widevine) Nagra, …
- DRM Providers
  - DRM Technical Solution Providers
  - Rights-only providers (buyDRM, …)
- Final User

DRM Principles

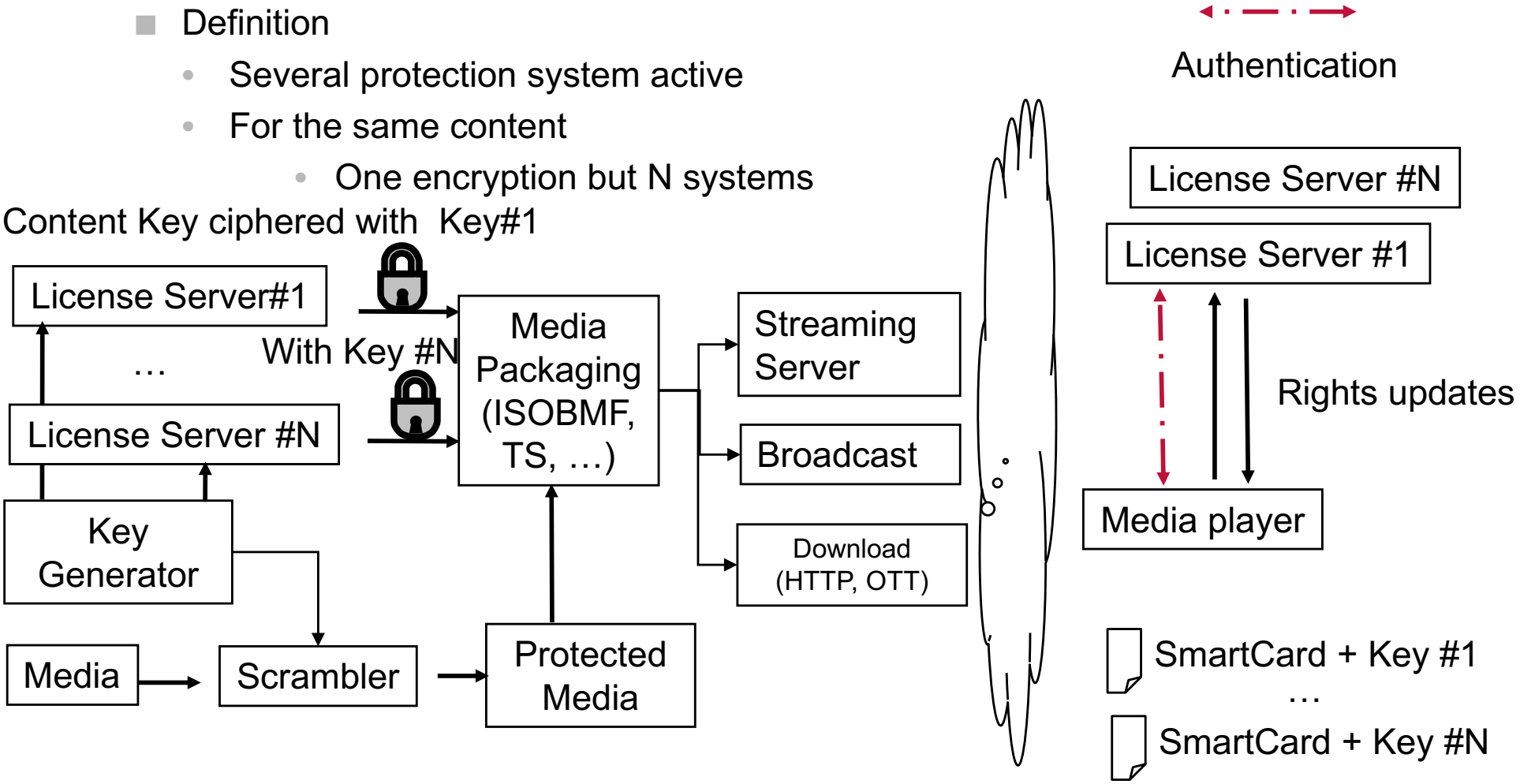
- **Content Protection**
  - Content is ciphered
    - Symmetrical cipher (faster)
    - Single or multiple cipher types for N streams
  - Keys have to be distributed
    - **Cipher the keys**
  - Ensure perfect synchronization key / media payload
    - If delay, decoding is broken
- **Access Rights**
  - Keys may be sent to everybody
    - Avoids one cipher version per user
    - Allows offline cipher of content
  - Rules describing cipher rights
    - Based on service subscription
    - Based on content provider logic
      - Geo-localization
      - Based on subscription models
      - Based on physical access type
      - Other….

Typical DRM Architecture



SimulCrypt Architecture

- Definition
  - Several protection system active
  - For the same content
    - One encryption but N systems



DVB CAS

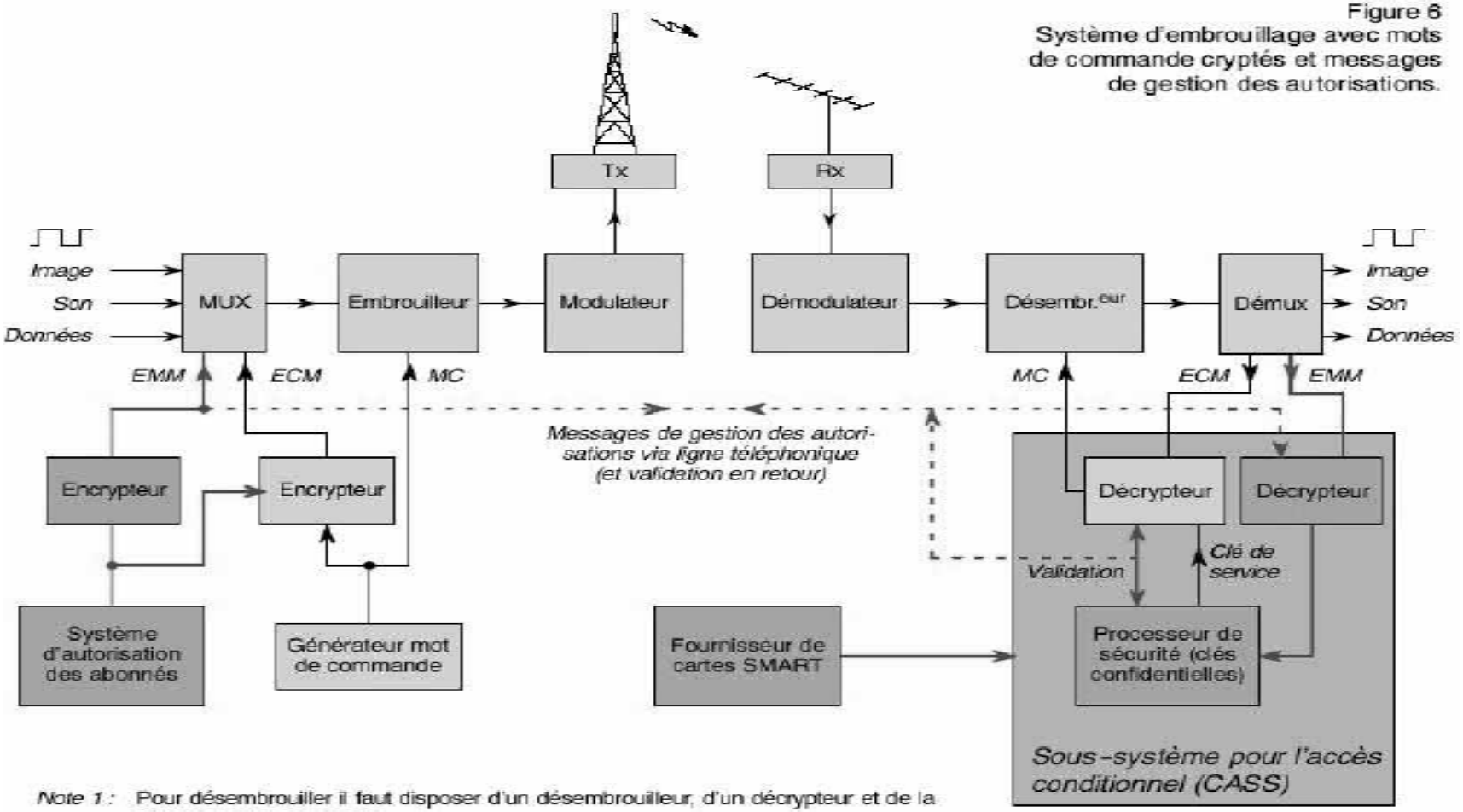
- **Conditional Access System**
- **DVB-CSA (Common Scrambling Algorithm)**
  - Common to all DVB services
  - Secret (NDA)
  - Algos: DVB CSA, CSA3
  - Implemented in hardware
- **DVB-CI (Common Interface)**
  - Interface STB / CAM
    - Data Decipher
    - Message Exchanges (config, status, etc)
  - Conditional Access Modules
    - All-in-one: removable hardware module
      - ~= PCMCIA
    - Removable SmartCard + certified hardware in TV/STB
  - DVB-CI+
    - Security after deciphering
    - Multi-stream
    - IPTV extensions

MPEG-2 TS and DVB Content Protection

- **Principles**
  - 2 keys (“control words”) for decipher available
  - Send ciphered in ECM (Entitlement Control Message)
    - Updated frequently (2 to 10 s)
  - Decipher & key swap based on rights
  - Rights update through EMM (Entitlement Management Messages)
  - Transport via MPEG-2 TS sections
- **CA_descriptor Signaling:**
  - Indicates where (PID) ECM/EMM are sent
  - In CAT (PID=1): information related to the whole crypto system, not to the media stream (system-wide): **EMM**
  - CA in the PMT: info per program and per crypto system: **ECM**

DVB CAS Architecture

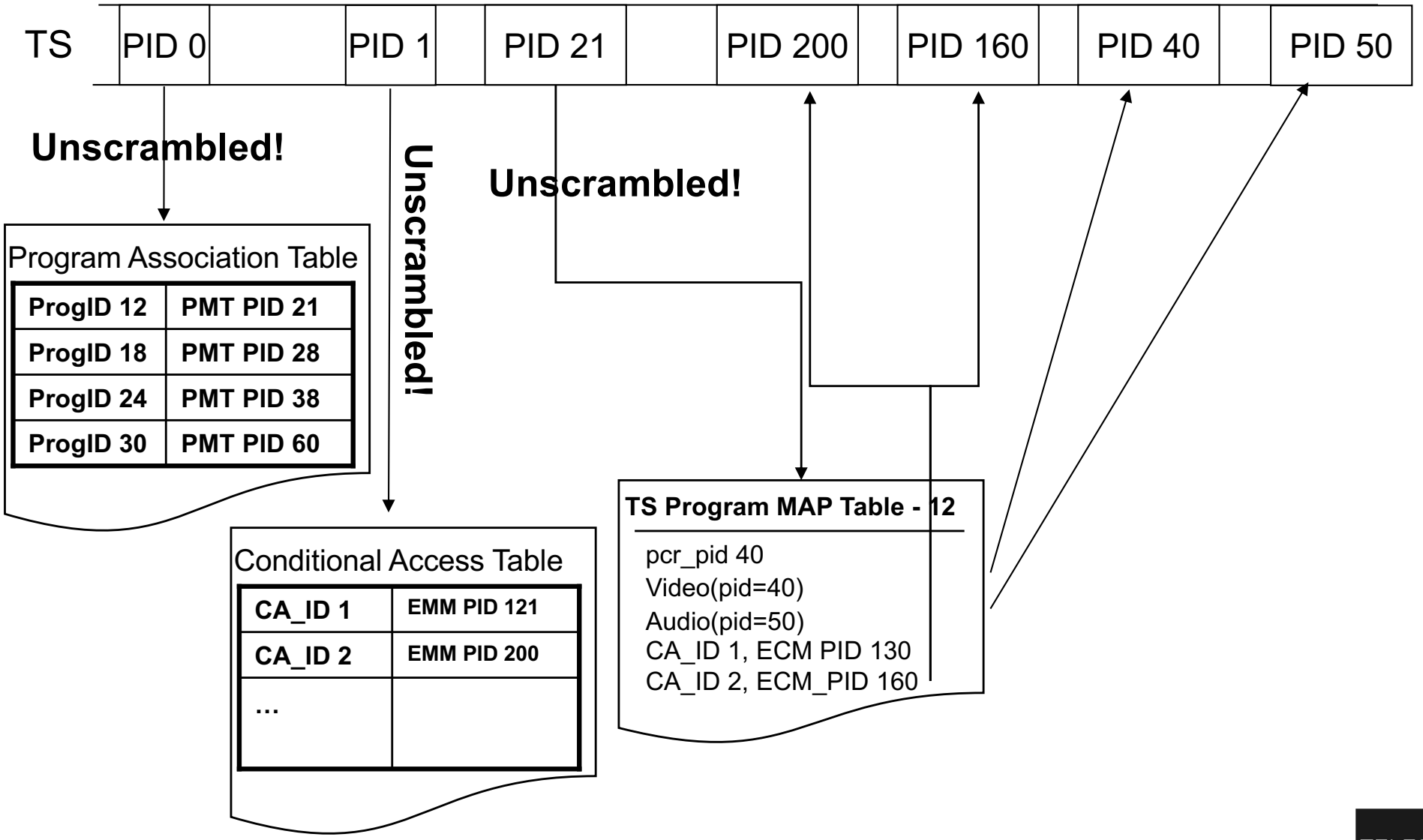
Figure 6
Système d'embrouillage avec mots de commande cryptés et messages de gestion des autorisations.



Note 1 : Pour désembrouiller il faut disposer d'un désembrouilleur, d'un décrypteur et de la clé de service concernée.

Note 2 : Pour décrypter il faut disposer des messages de gestion des autorisations (EMM) pour le programme donné. En général cela requiert des clés confidentielles mémorisées dans un sous-système pour l'accès conditionnel (CASS).

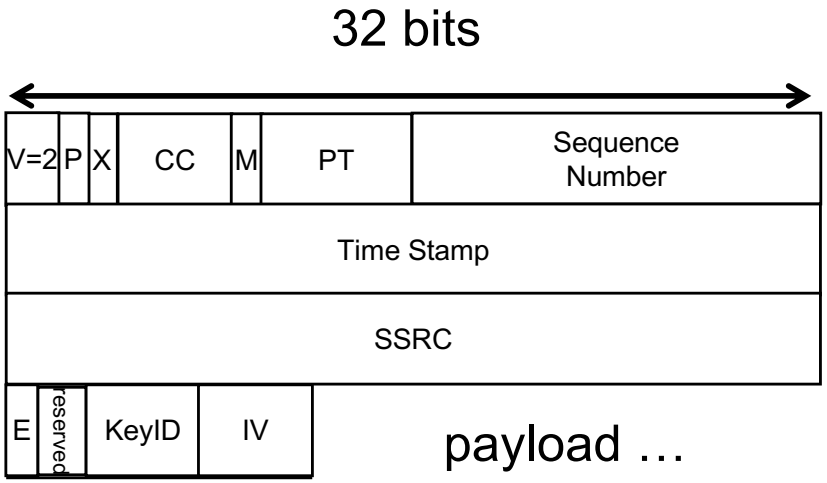
MPEG-2 TS: tune-in and scrambling



ISMACryp

- **Requirements**
  - RTP Streaming of secured content
    - != secure streaming of RTP content
  - Key rolling and key renewal
- **Problematic**
  - SRTP: single key per stream
  - Ensure media/key sync if several keys are used
- **Solution**
  - Clearing House for key exchanged signaled as URI in SDP
  - Cipher: AES-128 CTR mode
    - CTR incremented by (Nb Bytes in stream)/16
    - Needs to signal IV used to start deciphering
  - Additional header after RTP header:
    - Cipher/non-cipher bit
    - Key index
    - CTR IV for the packet

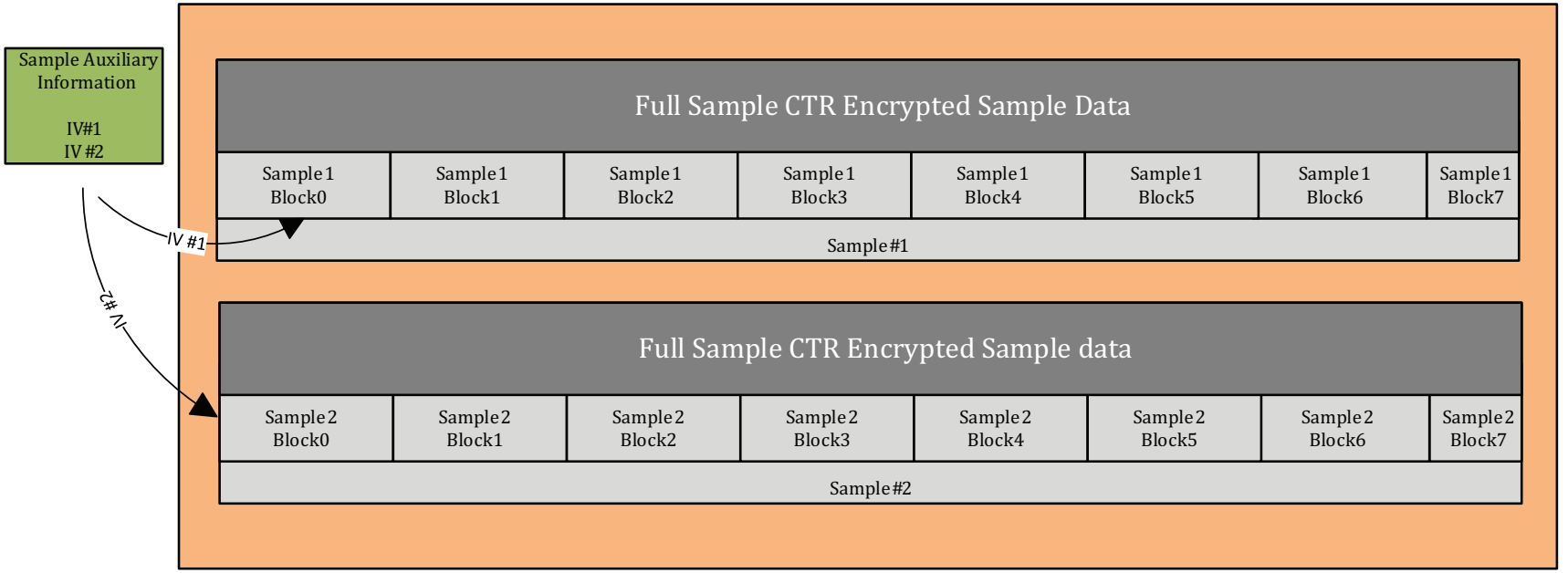
ISMACryp Packet



ISOBMF Common Encryption (CENC)

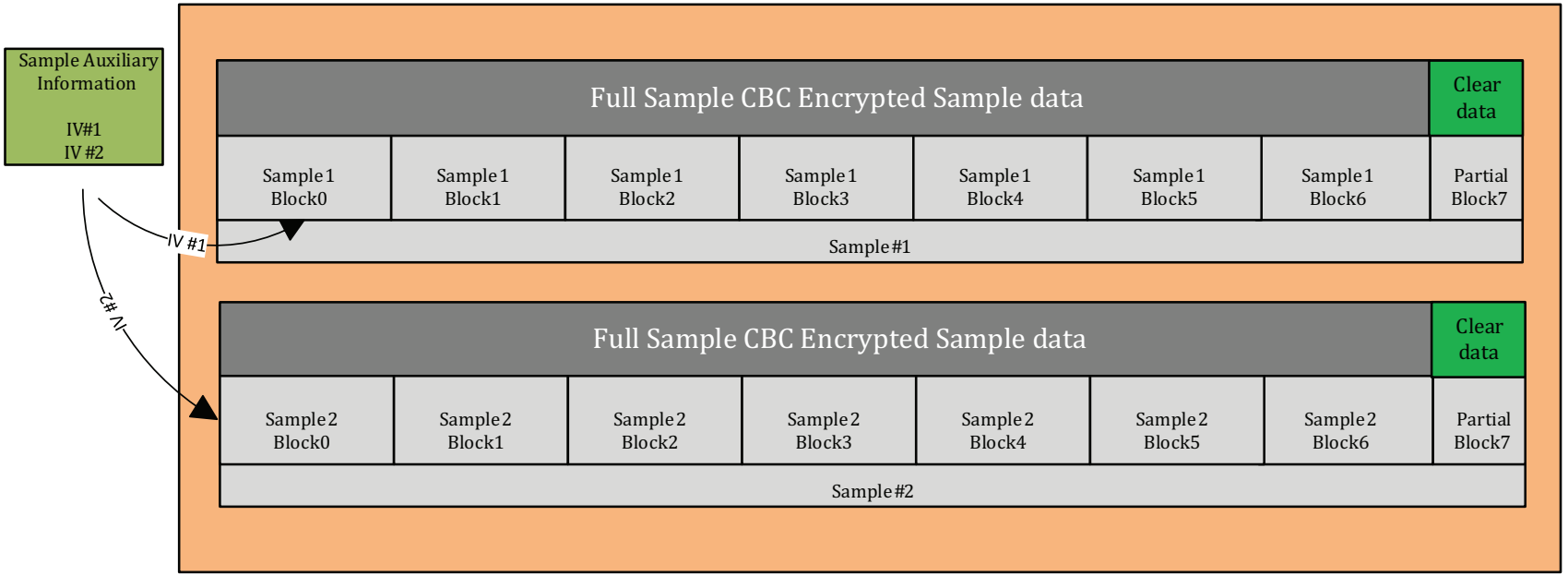
- **Multiple DRM for a single ISO file**
  - Private CAS data in box « pssh »
    - Stored in moov (file) or moof (fragment)
  - Several CAS in file possible via pssh.SystemID
  - Data linked to
    - the complete CAS via systemID (cf EMM)
    - Subset of key IDs (per key/track config of CAS)
  - Cipher used
    - AES-128 CTR mode
    - AES-128 CBC mode
    - Initialization Vector
      - 64 or 128 bits
- **Principle:**
  - One IV for each sample
    - Side data (auxiliary) for each sample
  - One default key for each track
  - Key Rotation possible by tagging samples with other keys through sample groups

Full sample encryption in AES-128 CTR mode



- Remaining bytes in last block (7) are discarded during file writing

Full sample encryption in AES-128 CBC mode



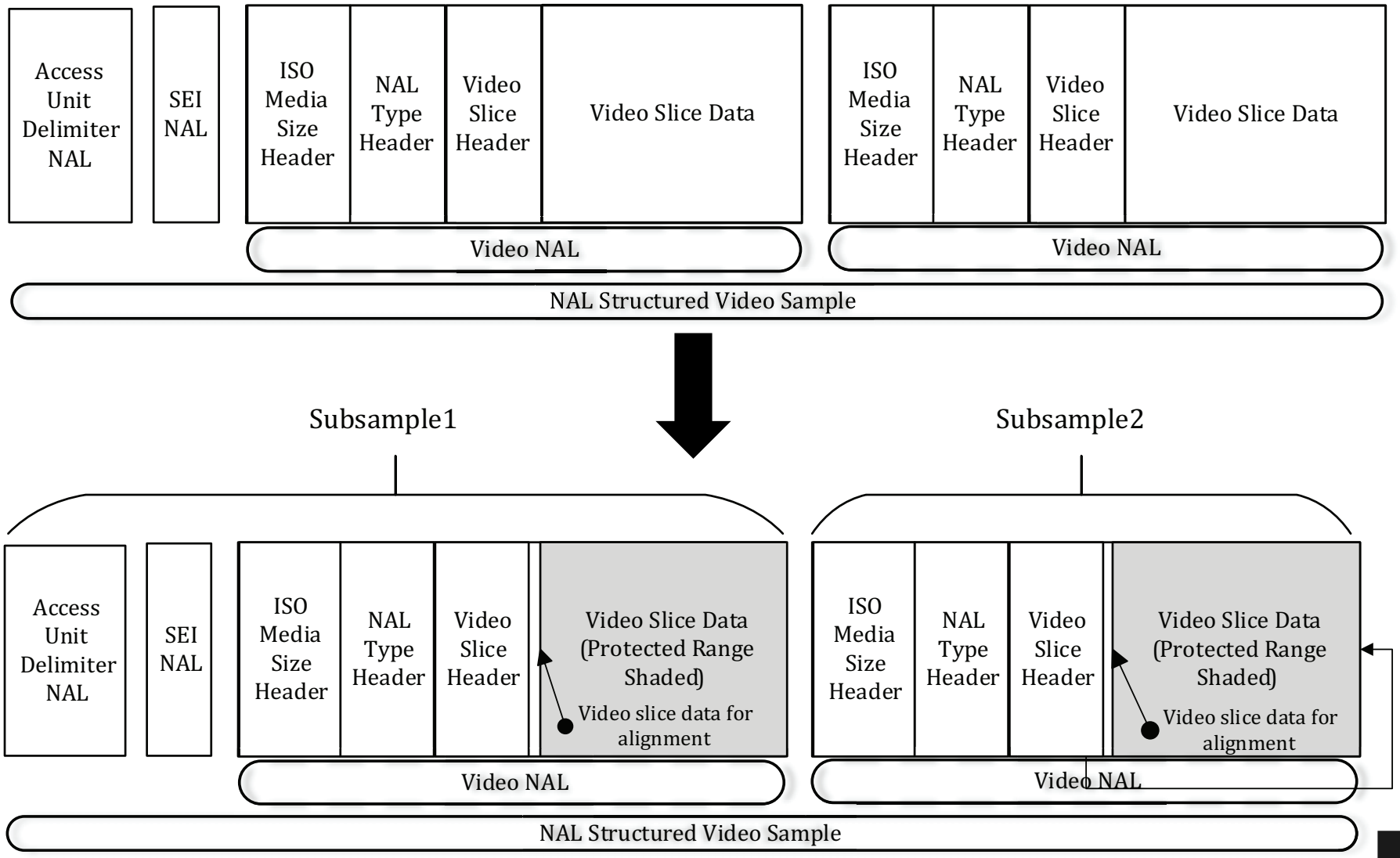
- Last block if not encrypted if not a multiple of 16 bytes

CENC Partial Encryption

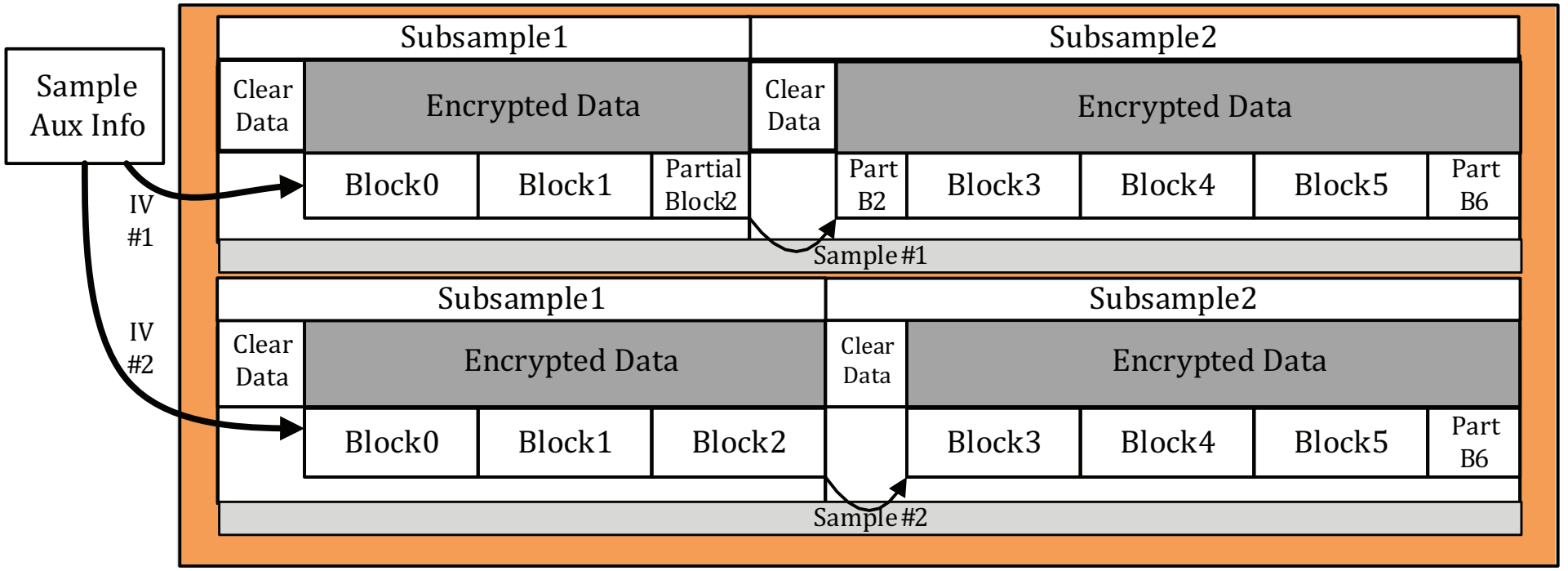
- **Use cases**
  - Some transport layers may inspect NAL header for AVC or HEVC
    - Random access type
    - Temporal / spatial scalability
  - Some transport systems work at NAL level, not AU level
    - cf RTP packetization
  - Slice header may contain useful information:
    - Parallel decoding type in HEVC
      - Wavefront processing instructions
      - Tile base
    - Picture number
      - Error recovery when losses and multiple slices in pictures
  - Ciphering entire AU hides this information
    - Partial encryption leaving slice header in the clear
- **Subsamples**
  - Logically breaks the sample into smaller data chunks
    - Sample data is still contiguous in the file
  - In side data carrying IV
  - Indicates list of subsamples, with
    - BytesOfClearData
    - BytesOfProtectedData

```
aligned(8) class CencSampleAuxiliaryDataFormat
{
   unsigned int(Per_Sample_IV_Size*8) InitializationVector;
   if (sample_info_size > Per_Sample_IV_Size )
   {
      unsigned int(16) subsample_count;
      {
        unsigned int(16) BytesOfClearData;
        unsigned int(32) BytesOfProtectedData;
      } [subsample_count ]
   }
}
```

NAL Video Structure exemple

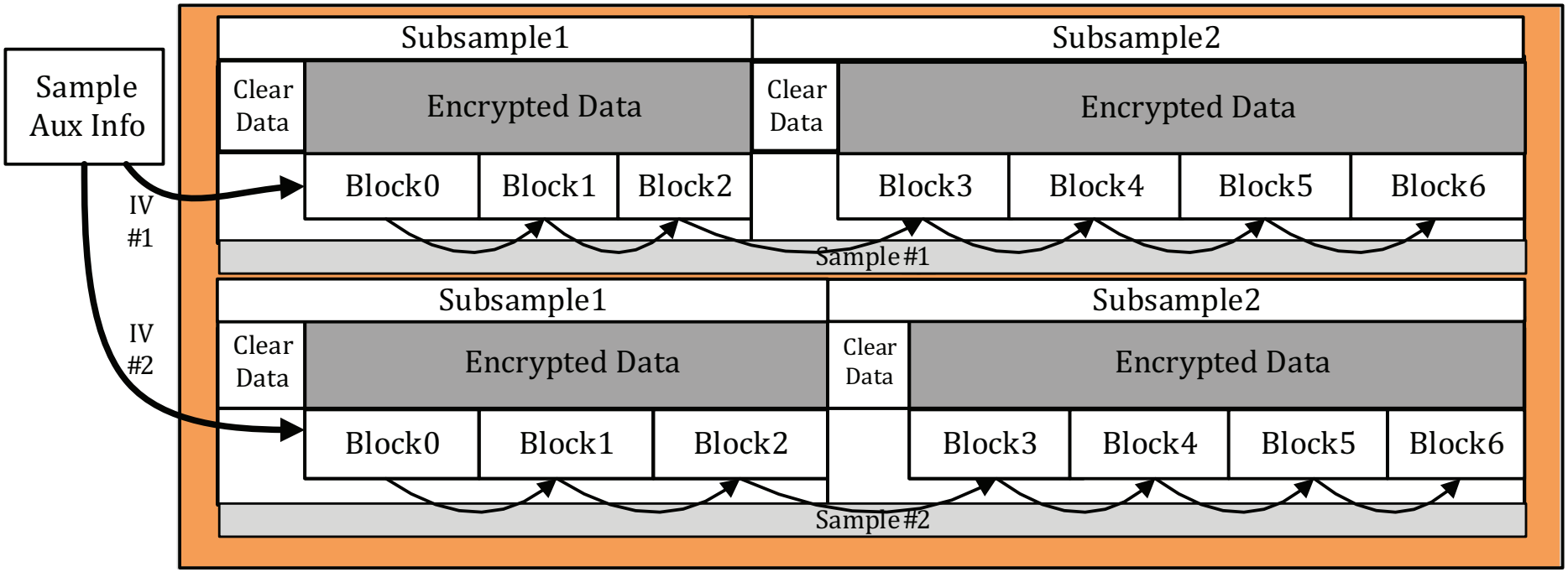


AES-128 CTR for NAL-based videos (AVC|H264, HEVC)



- Block counter only incremented for ciphered parts
  - Avoids un-necessary resync of block counter after clear data

AES-128 CBC for NAL-based videos (AVC|H264, HEVC)

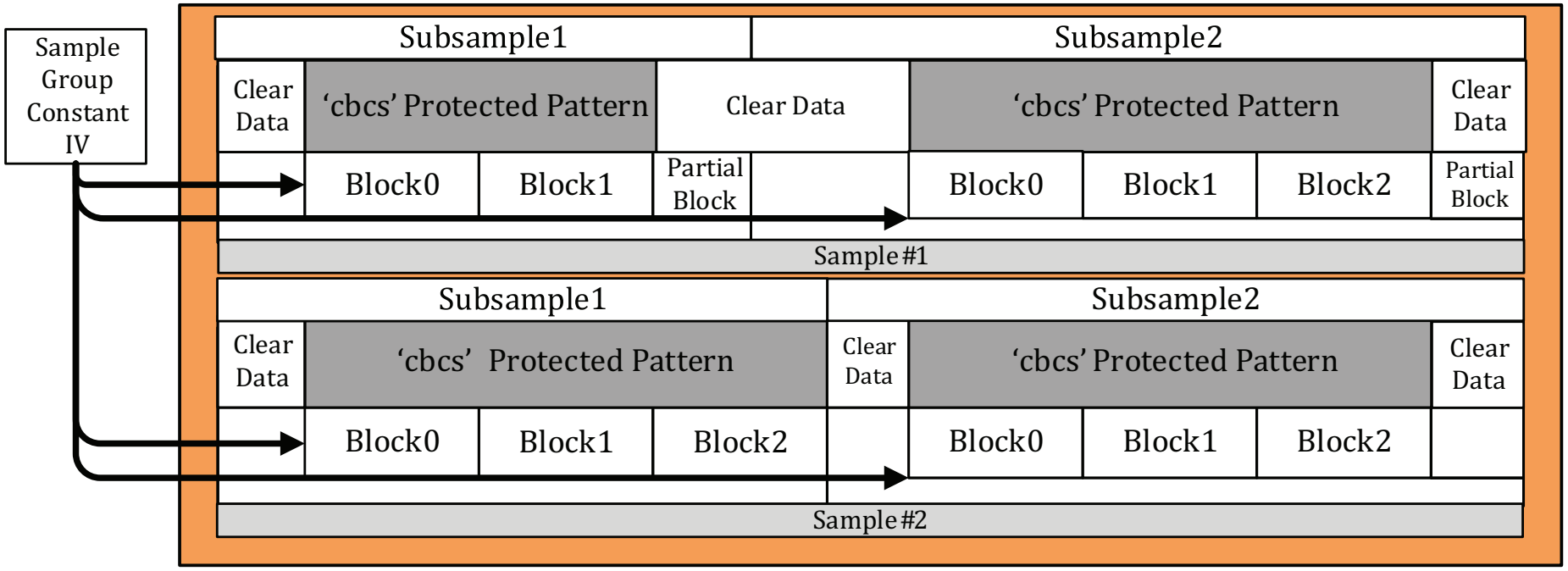


- If NAL < 16 octets, not encrypted
- Each protected part = Nx16 bytes
  - No partial block at the end of the subsample
  - Adjustment of first clear part size
- Mode ‘cbc1’

CENC Patern Encryption

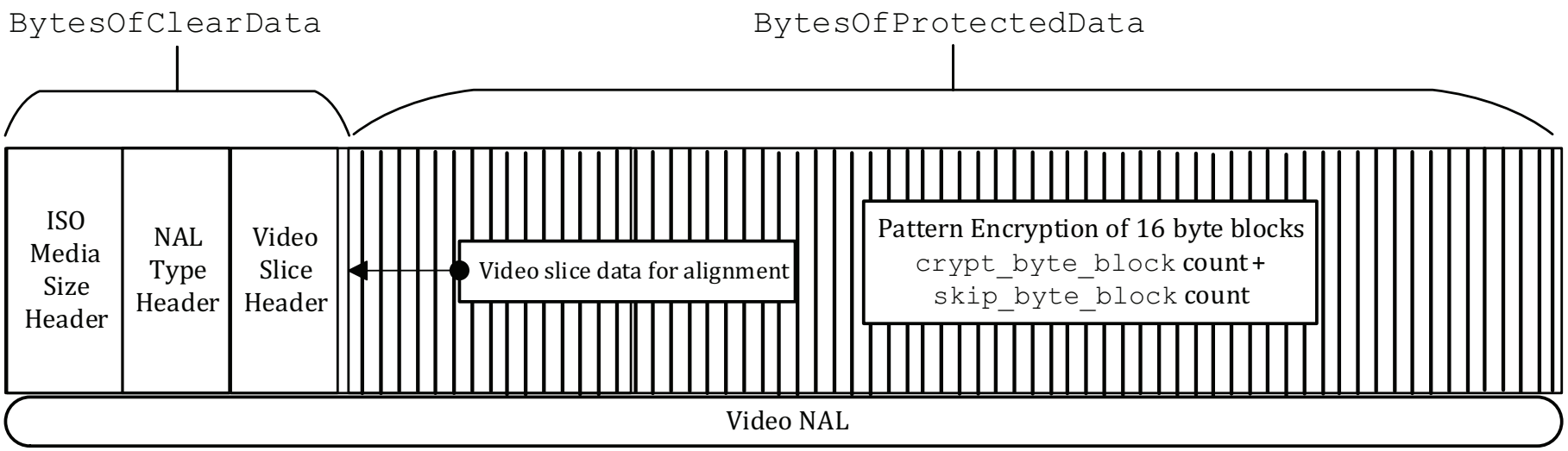
- **Cost Problematic**
  - Size of signaling for each AVC/HEVC sample
    - clear_data / protected_data
    - Initialization Vector
      - CBC may still be robust with a constant IV for all samples
  - Processing
    - Corrupting only parts of the video result in unusable video
    - Only the slice header has to be in the clear, but size bounded
- **Solution**
  - Describe pattern of {C clear bytes clear + P protected bytes)
  - Constant IV or not
  - No signaling required per sample

AES-128 CBC for NAL-based videos (AVC|H264, HEVC) with pattern



- If NAL < 16 octets, not encrypted
- Each protected part <= Nx16 bytes
  - Pattern of encrypted/unencrypted blocks
  - If last pattern exceeds the protected size, last block shall be clear
- Mode 'cbcs'

AES-128 CBC pattern mode exemple



DASH Segment Encryption and Authentication

- **Encryption**
  - Entire segment is ciphered
  - AES-128 CBC mandatory
    - Remaining block is padded
    - Nb bytes padded with value of NbBytes:
      - 0x01
      - 0x02 0x02
      - 0x04 0x04 0x04 0x04
      - …
- **Key description**
  - Keys and IVs signaled in MPD via templates
    - Same substitution rules than for segments
  - Rules to associate {key,IV} and segments
- **Not ciphered**
  - InitializationSegment, Index, BitstreamSwitchingSegment

Apple HLS Stream Encryption

- **Per segment**
  - Each TS file ciphered in AES 128 CBC
  - Last block is padded as previously
- **Per Stream**
  - PID audio or video ciphered using AES 128 CBC
  - Dedicated streamType for protected stream
    - Not compatible with standard CAS
  - Similar to CENC:
    - NALU header clear, then Nx16 bytes ciphered then end in clear
    - Same principles for AC3 and AAC

Apple Sample Encryption AVC Example

- **Signaling**
  - AVC|H264 PID type 0xdb
  - Private_data_indicator ‘zavc’
- **PES Payload format:**
  - Start code 0x00000001
  - Then encrypted NAL unit

```
Encrypted_NAL_Unit () {
  NAL_unit_type_byte // 1 byte
  unencrypted_leader // 31 bytes
  while (bytes_remaining() > 16) {
    protected_block_one_in_ten // 16 bytes
  }
  unencrypted_trailer // 1-16 bytes
}
```

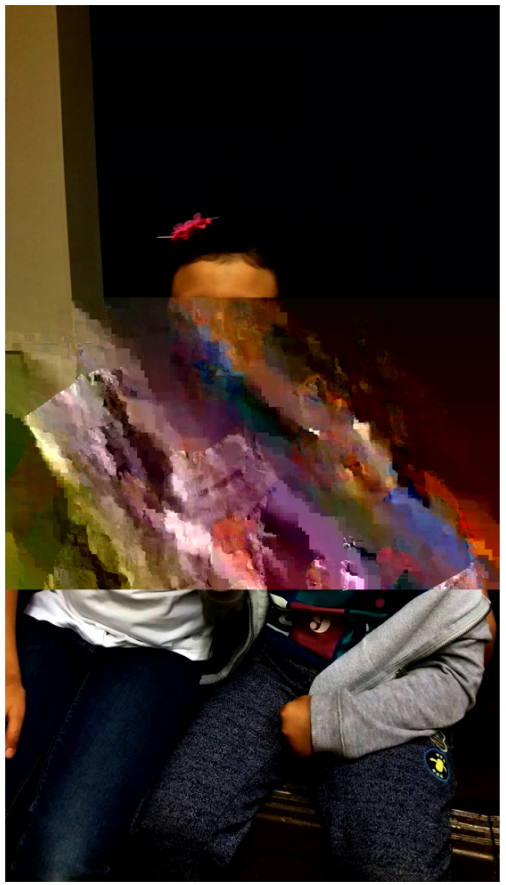
Apple Sample Encryption AAC Example

- **Signaling**
  - AAC ADTS PID type 0xcf
  - Private_data_indicator ‘aacd’
- **PES Payload format:**
  - Encrypted AAC frame

```
Encrypted_AAC_Frame () {
  ADTS_Header // 7 or 9 bytes
  unencrypted_leader //16 bytes
  while (bytes_remaining() >= 16) {
    protected_block // 16 bytes
  }
  unencrypted_trailer // 0-15 bytes
}
```

Selective Encryption

- **Principles**
  - Encrypt only parts of the bitstream
  - Break the reconstructed image
  - Keep the bitstream syntax intact
  - Combination with motion-constrained slices/tiles
    - Parts of the image properly decode
    - Avoids error propagation in ME
- **Applications**
  - Privacy management
  - Base layer in clear, enhancement protected



Selective Encryption: AVC example

- **Entropy coding results**
  - Words of variable length
  - Length is derived from most significant bits
- **Encrypt less significant bits :**

	Mb_QP_Delta value	Code-word
0	0	1
1	1	010
2	-1	011
3	2	00100
4	-2	00101
5	3	00110
6	-3	00111
7	4	0001000
8	-4	0001001
…	….	…

	Mb_QP_Delta value	Code-word Encrypted bits
0	0	1
1	1	01***0***
2	-1	01***1***
3	2	001***00***
4	-2	001***01***
5	3	001***10***
6	-3	001***11***
7	4	0001***000***
8	-4	0001***001***
…	….	…

  - CBC/other block cyphers:
    - Blocks of input data XORed then encrypted
    - **Impossible to parse the content**
  - CTR/other stream cyphers
    - Nounce is encrypted, input is XORed with nounce
    - Possible to parse bitstream and only do XOR on protected bits

Media security in browsers

- **W3C Encrypted Media Extensions**
  - Javascript model for key setup in decoding pipeline
    - Extends HTMLMediaElement
    - Optional for HTML5 conformance
- **Principles**
  - System query
    - One mandatory predefined system ClearKey "org.w3.clearkey »
    - Other systems (widevine, playready, fairplay, …) depend on the browser/OS
  - System configuration
    - License exchange and validation
  - Persistent data secure storage
  - Setup of media keys
    - Single key
    - Multiple keys for key rotation (usually for live content)

Encrypted Media Extension Architecture