# TLS/DTLS AES-CTR

draft-modadugu-tls-ctr-00

Nagendra Modadugu

Eric Rescorla

# AES-CTR Overview

- Works like a stream cipher, e.g. RC4
  - XOR keystream with plain text:

    `CT[i] := PT[i] ⊕ AES(CTR(i))`

  - Increment `Counter`

- `Counter` encrypted to generate keystream
  - `Counter` MUST never be re-used (with same key)
- No harm if `Counter` is public
  - But MUST be initially unpredictable

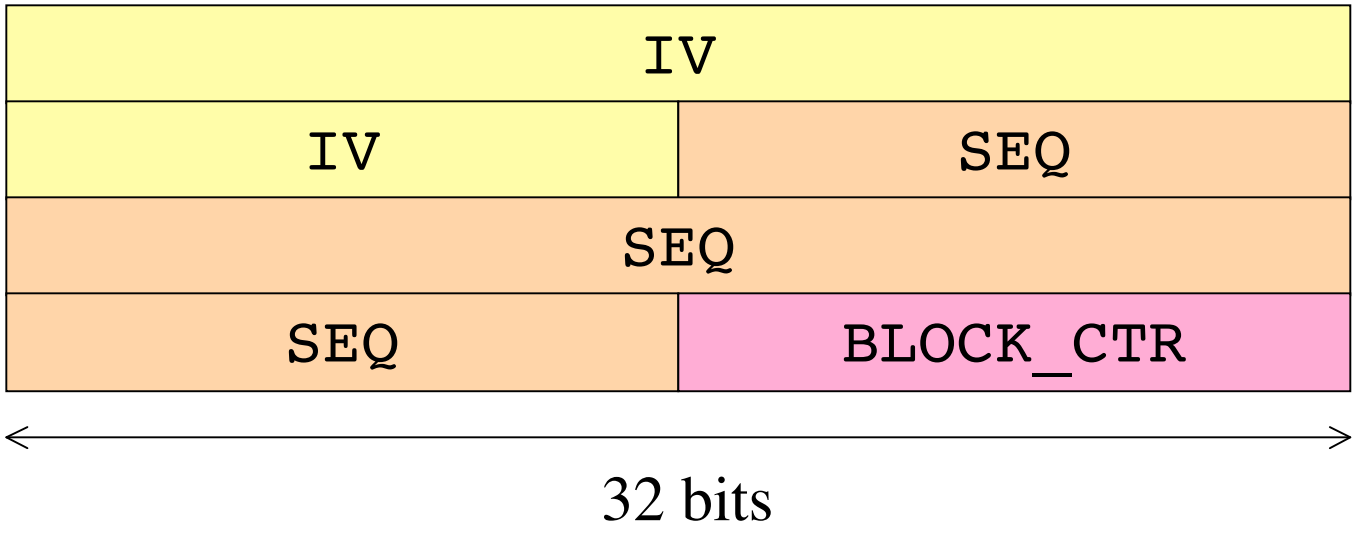
# Motivation

- Low bandwidth
  - Save between 17-32 bytes compared to CBC
- Random access (for DTLS)
- Parallelizable/pipelining
- Implement both block/stream ciphers with AES

# Strategy

1. Secure
2. Minimize differences between TLS/DTLS
3. Base off AES-CTR in IPsec if possible

# Counter Design [1]




- `IV := {client_write_IV, server_write_IV}`
  (Least significant 48-bits)
- `SEQ := {seq_num}` (64-bits)
- `BLOCK_CTR := 1` (16-bits)

# Counter Design [2]

- IV's generated by TLS/DTLS KDF
  - Refreshed upon session re-negotiation
- Sequence number
  - Implicit for TLS
  - For DTLS, use (`epoch || seq_num`)
- Block counter
  - 16-bits plenty for TLS/DTLS records

# Alternative Design

- Implicit record "tag" (in place of `seq_num`)
  - e.g. LFSR generated
- No point, since:
  - Uses up more bandwidth
  - Sufficient for counter to be unique

$\Rightarrow$ Back to original design

Questions?