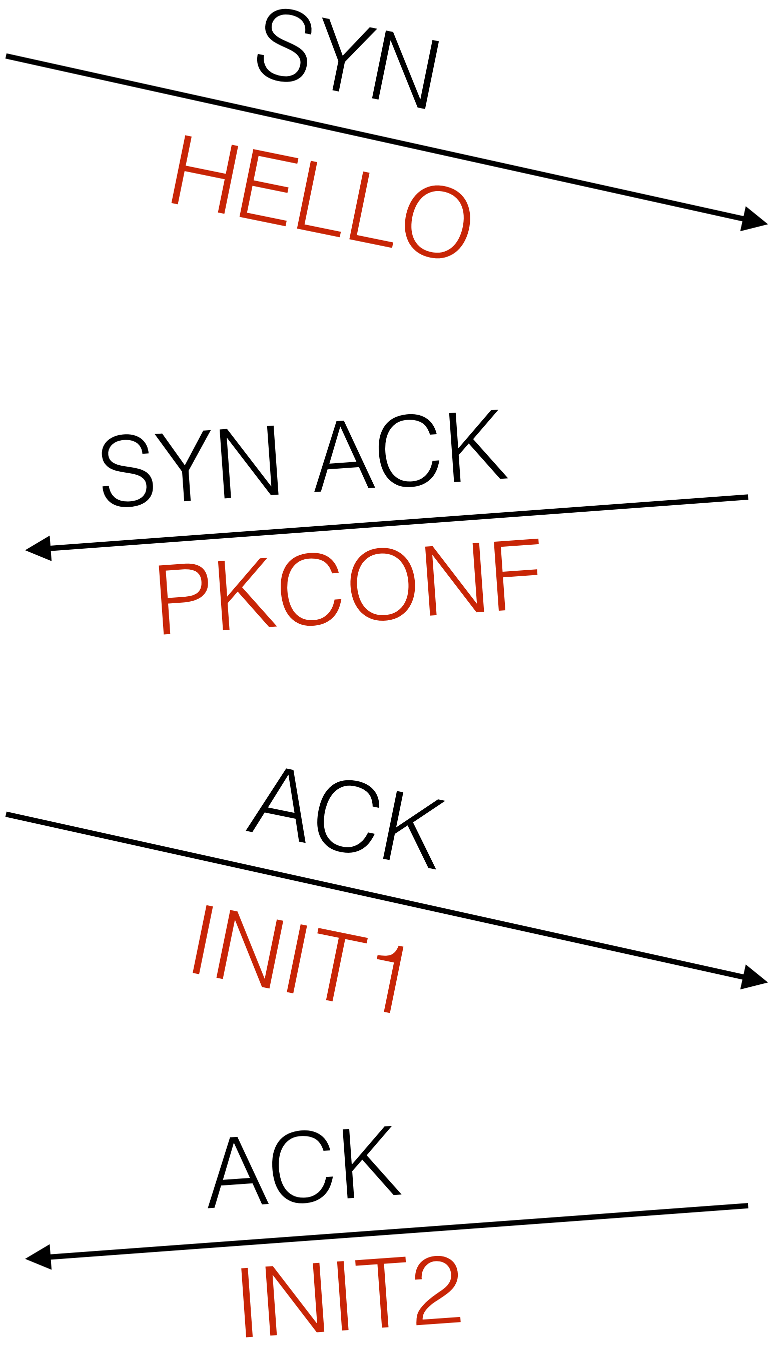
tcpcrypt

Andrea Bittau, Dan Boneh, Mike Hamburg,
Mark Handley, David Mazières, Quinn Slack, Daniel Giffin, Eric Smith

Outline

- tcpcrypt review
- Current issues and how we plan to address them
- Design goals moving forward.

tcpcrypt: previous draft



SYN

HELLO

SYN ACK

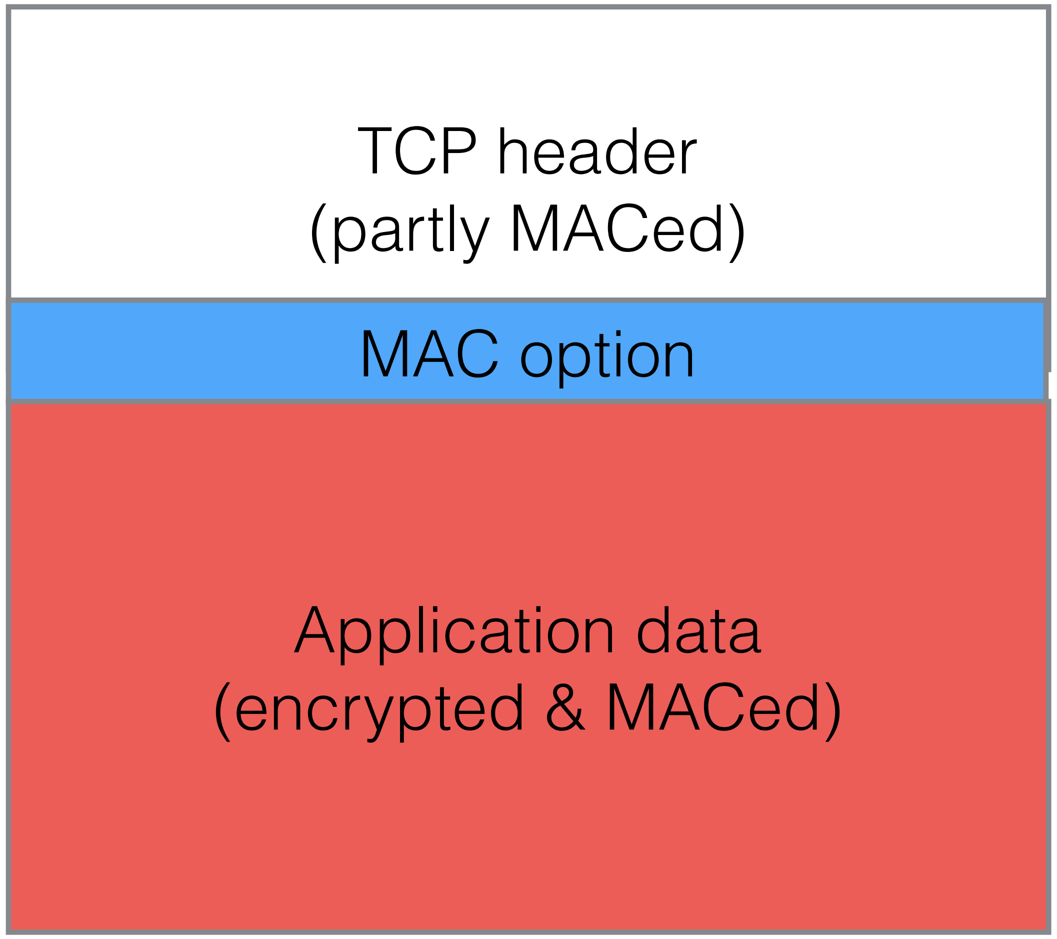
PKCONF

ACK

INIT1

ACK

INIT2



TCP header
(partly MACed)

MAC option

Application data
(encrypted & MACed)

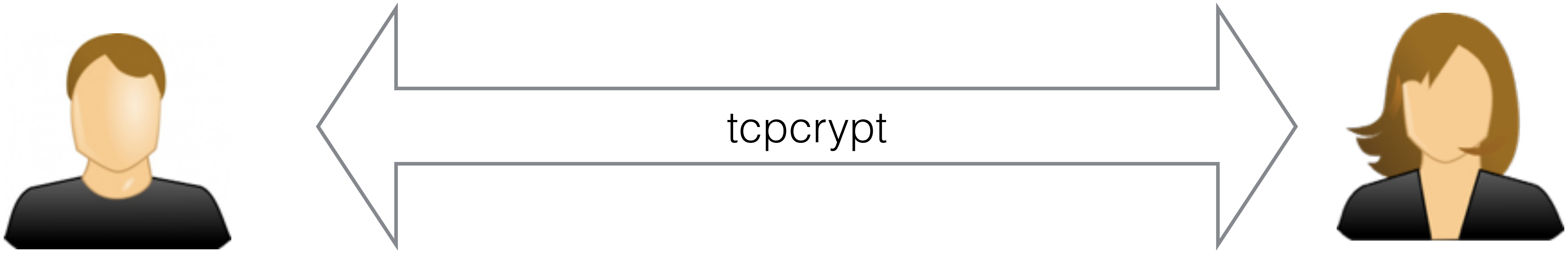
Session ID

HMAC(cookie, ABCD…);

ABCD…
Signed by Alice

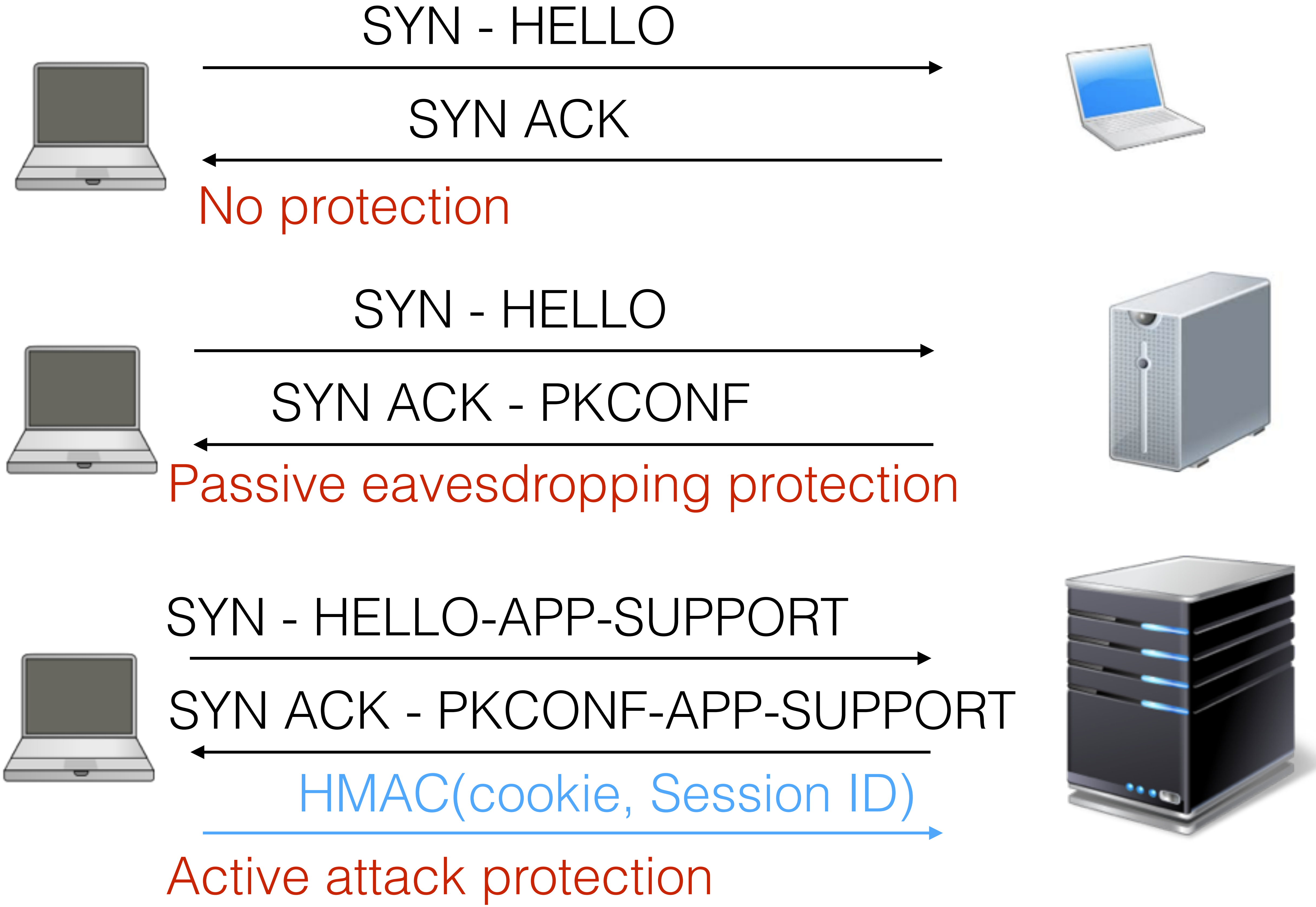
Session ID: ABCD…

Session ID: ABCD…



getsockopt(s, IPPROTO_TCP, TCP_CRYPT_SESSIONID, …);

Application support bit



SYN - HELLO

SYN ACK

No protection

SYN - HELLO

SYN ACK - PKCONF

Passive eavesdropping protection

SYN - HELLO-APP-SUPPORT

SYN ACK - PKCONF-APP-SUPPORT

HMAC(cookie, Session ID)

Active attack protection

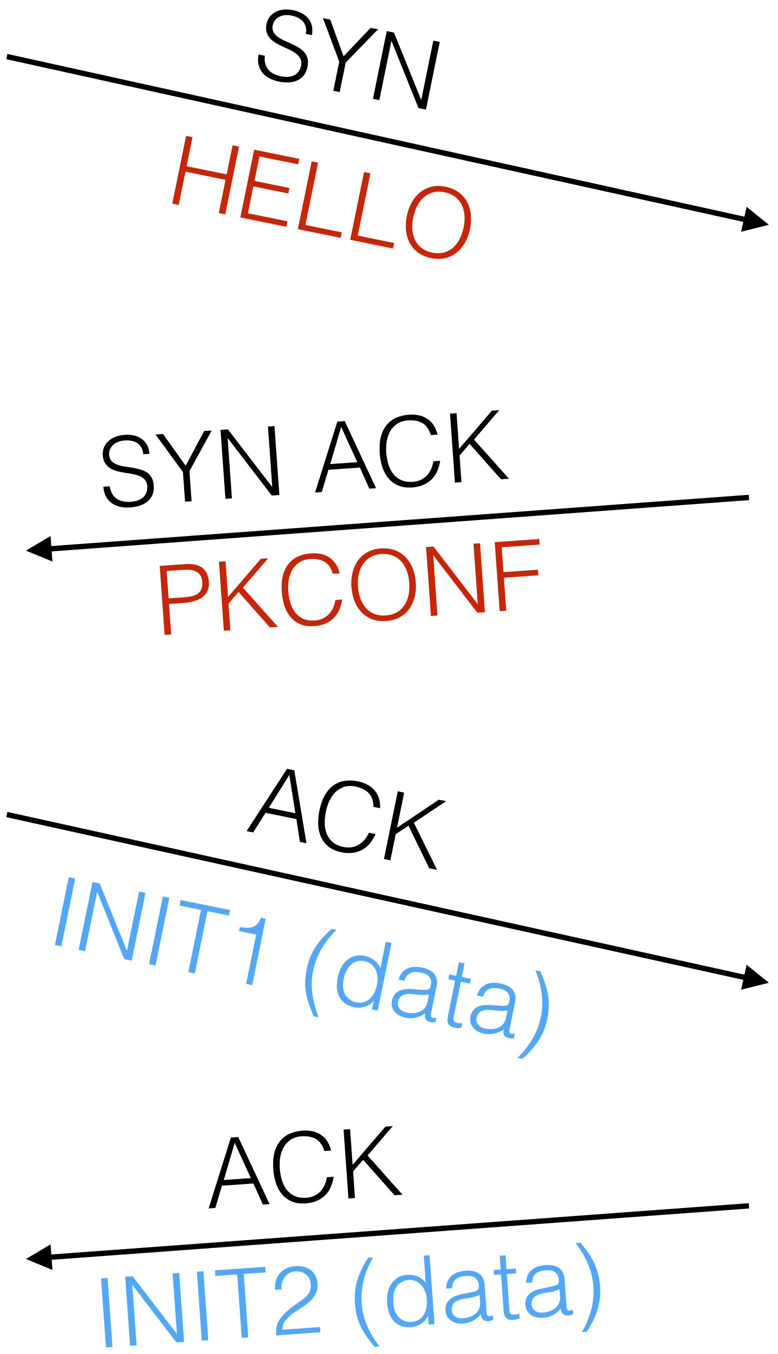
tcpcrypt status

- Official Ubuntu and Debian packages. Thanks: Daniel Gillmor.
- Official Fedora package. Thanks: Paul Wouters
- Windows, Mac OS, FreeBSD versions available.

Current issues

- Criticism: TCP header operation may be incompatible with some middleboxes.
  - Response: next draft won’t MAC TCP header.
- Criticism: use EDO instead of payload for INIT1 and INIT2.
  - Response: key exchange tied to stream not segments. (TCP-use-TLS does not place key exchange in options either.)
  - Note: EDO works well with current MAC option.
- Open question: store MAC in TCP header or use Type Length Value?
  - Response will depend on today’s meeting.

Handshake already uses TLV



SYN

HELLO

SYN ACK

PKCONF

ACK

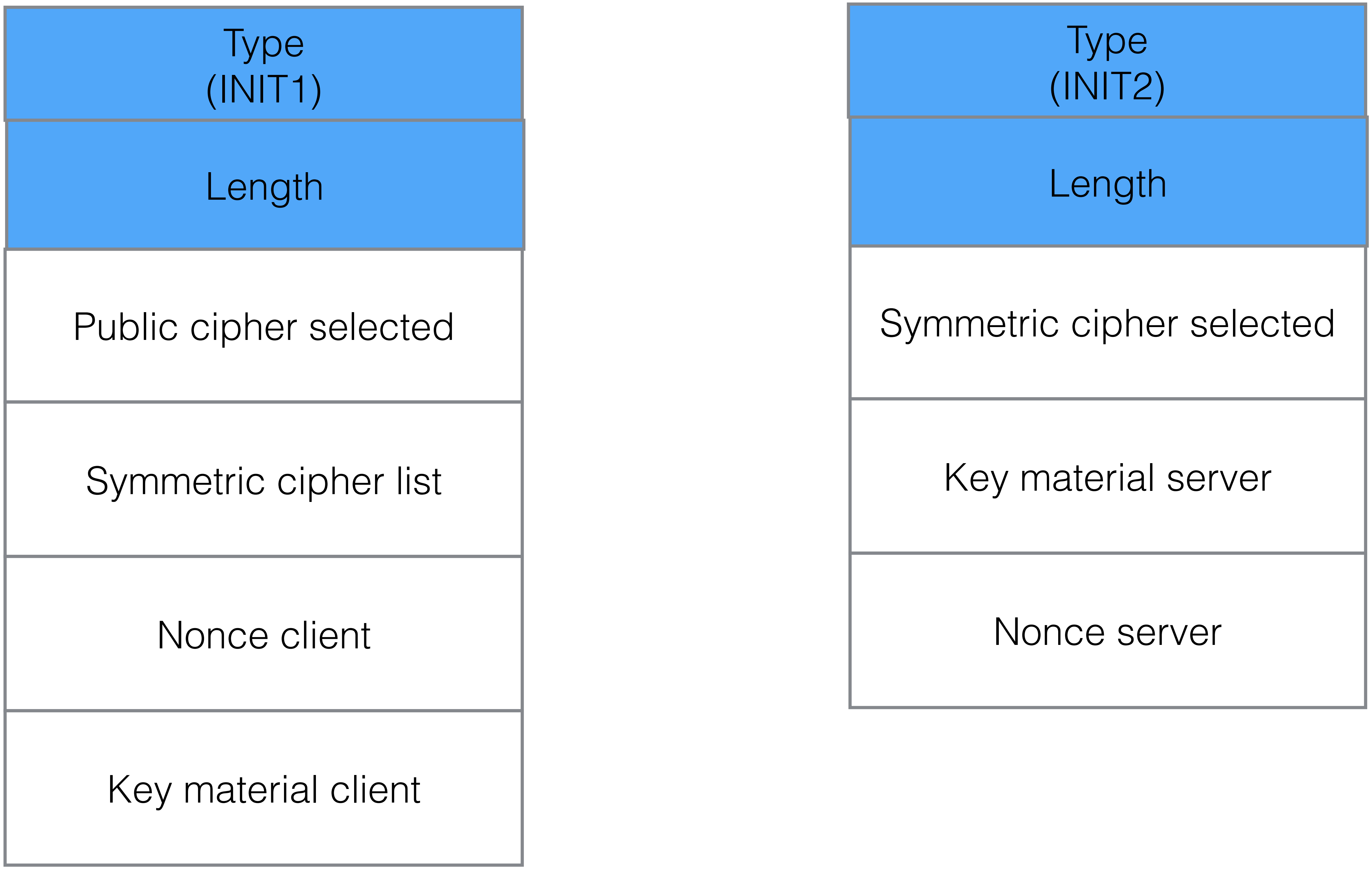
INIT1 (data)

ACK

INIT2 (data)

Type (4 bytes)	Length (4 bytes)	Value (Length bytes)

Key exchange



Type
(INIT1)

Length

Public cipher selected

Symmetric cipher list

Nonce client

Key material client

Type
(INIT2)

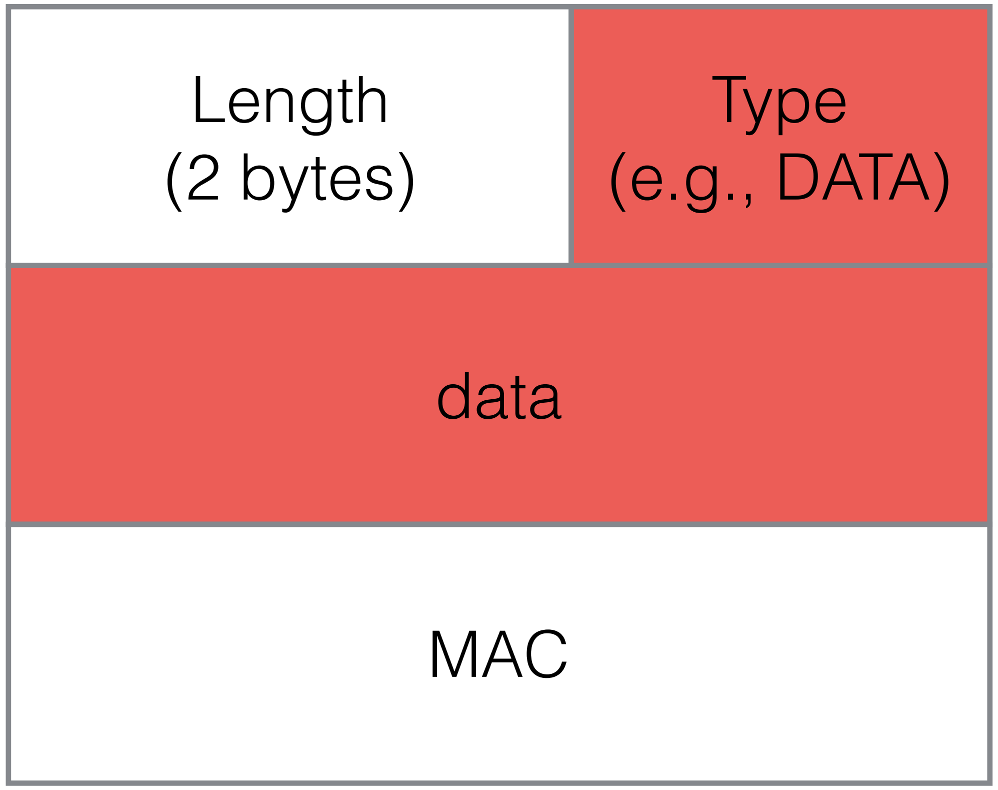
Length

Symmetric cipher selected

Key material server

Nonce server

TLV after key exchange?



Length
(2 bytes)

Type
(e.g., DATA)

data

MAC

🟥 Encrypted then MACed

MAC option vs TLV

- Advantages of MAC option:
    - Simplifies implementation. No need to buffer unauthenticated ciphertext; likely to have different bugs from TLS which uses TLV.
    - Easy to discard corrupt packets without aborting connection.
    - No changes to TCP semantics (flow control, socket buffer options).
- Advantages of TLV:
    - Greater robustness to middleboxes.
    - Easier to make work with TSO.

tcpcrypt goals adopted so far

- Layer 4 encryption is useful (tcpinc). Previous debate: IPSec & TLS suffice - no need for Layer 4.
- Out-of-band signaling for support (e.g., in SYN) critical for incremental deployment and backward compatibility.
- Separating authentication and encryption.
- Session ID (channel binding) useful abstraction for authentication support.

Other tcpcrypt design goals

- Application support bits.
- Low latency connection establishment. Piggyback on entire 3-way handshake.
- Simple security proofs.
- Amenable to simple implementations (including a verified implementation).