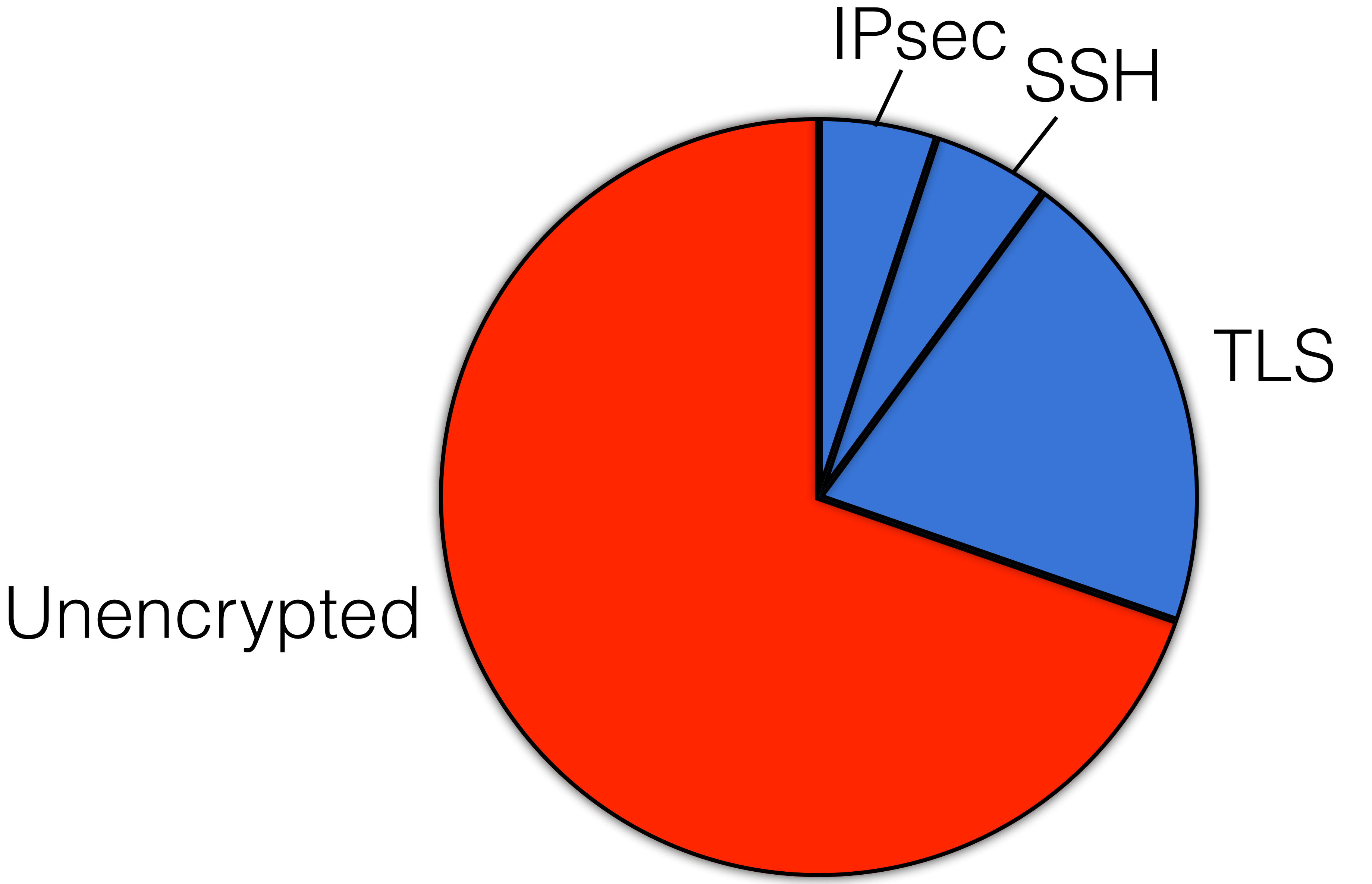
tcpcrypt

Andrea Bittau, Dan Boneh, Mike Hamburg,
Mark Handley, David Mazières, Quinn Slack

Stanford, UCL

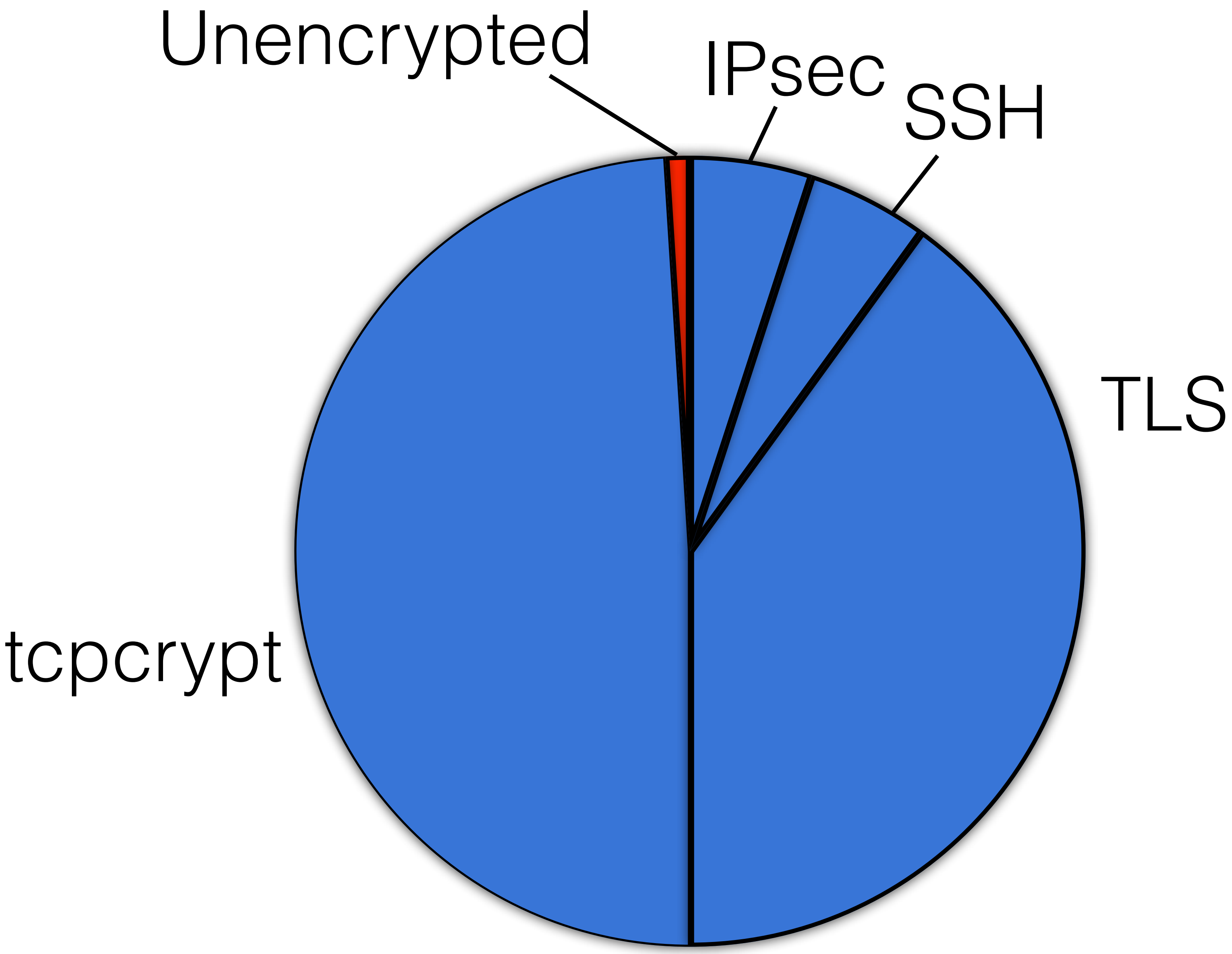
Reminder: project goal



TCP traffic today

Not drawn to scale

Reminder: project goal



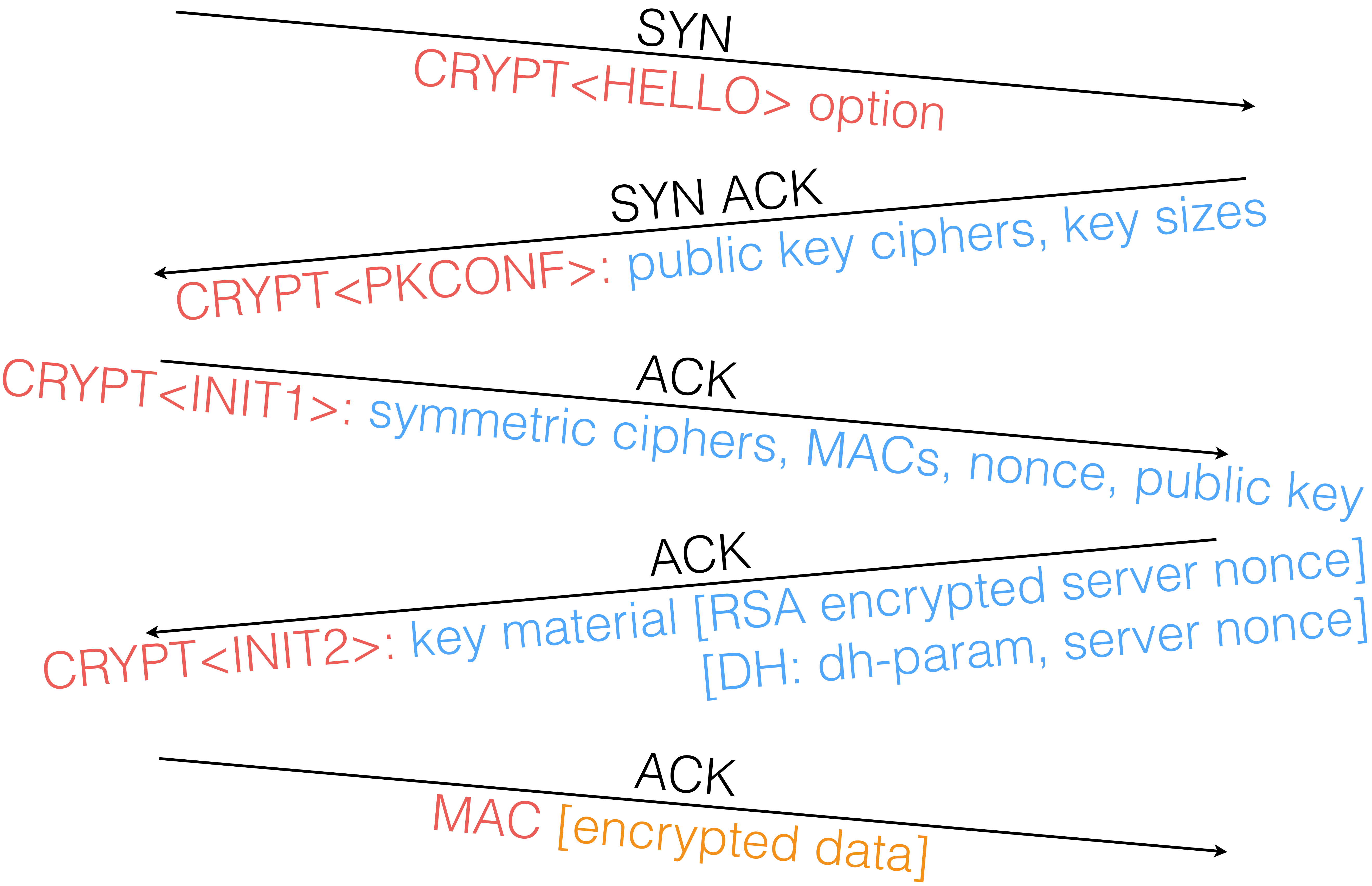
Goal for TCP traffic

Not drawn to scale

tcpcrypt summary

- Two new TCP options: CRYPT & MAC.
- Public-key based session key negotiation in extended 4-way TCP handshake.
    - Cached session completes in normal 3-way handshake.
- Encryption of payload, integrity of most TCP header fields and full payload.
- Session ID, app-support bit for future app-level security.

tcpcrypt in action



SYN

CRYPT<HELLO> option

SYN ACK

CRYPT<PKCONF>: public key ciphers, key sizes

ACK

CRYPT<INIT1>: symmetric ciphers, MACs, nonce, public key

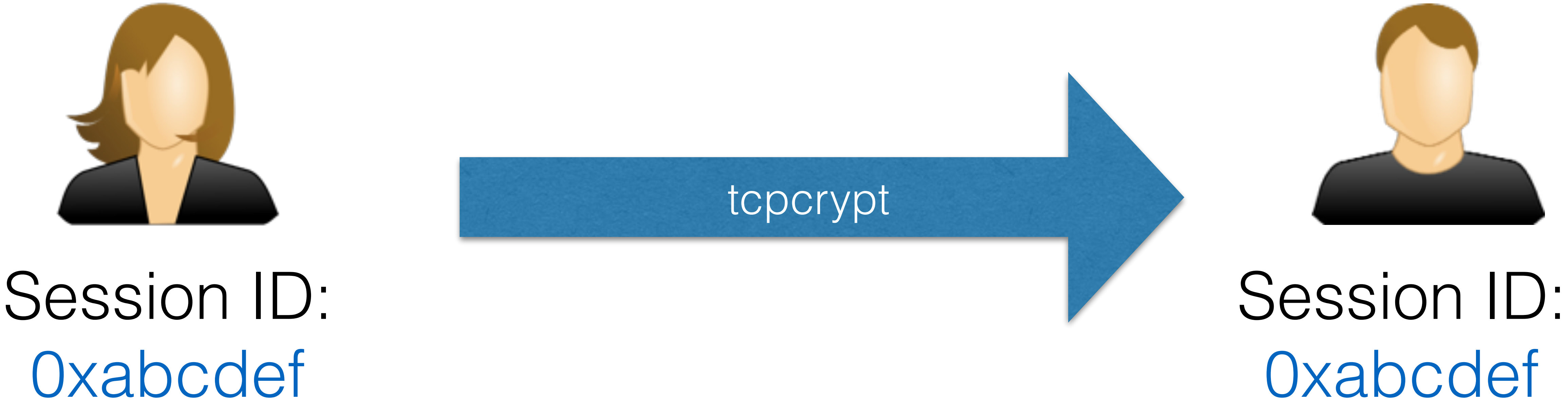
ACK

CRYPT<INIT2>: key material [RSA encrypted server nonce]
[DH: dh-param, server nonce]

ACK

MAC [encrypted data]

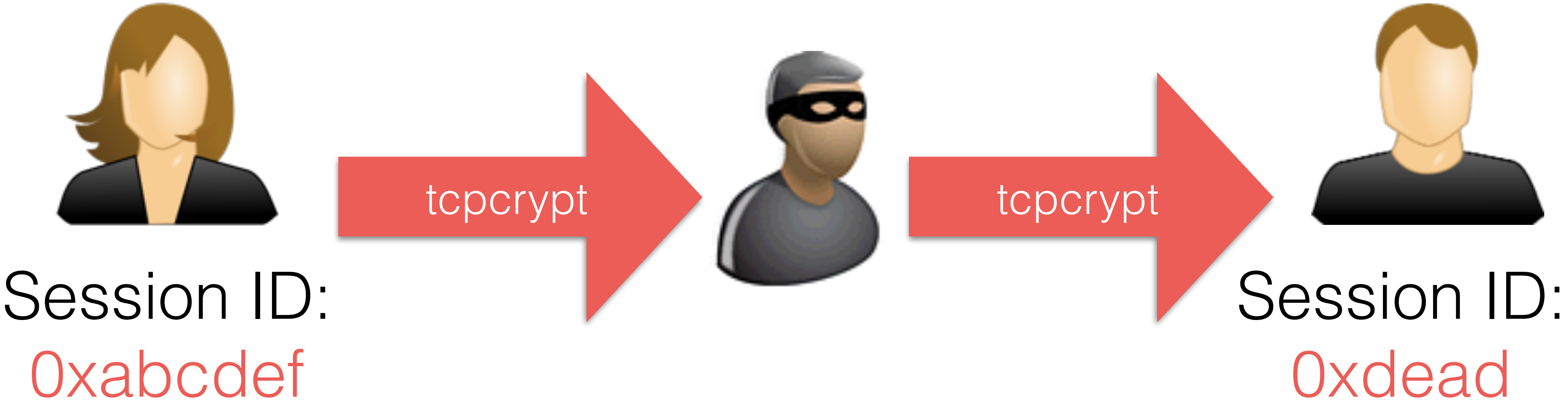
Session ID



tcpcrypt

Session ID:
0xabcdef

Session ID:
0xabcdef



tcpcrypt

tcpcrypt

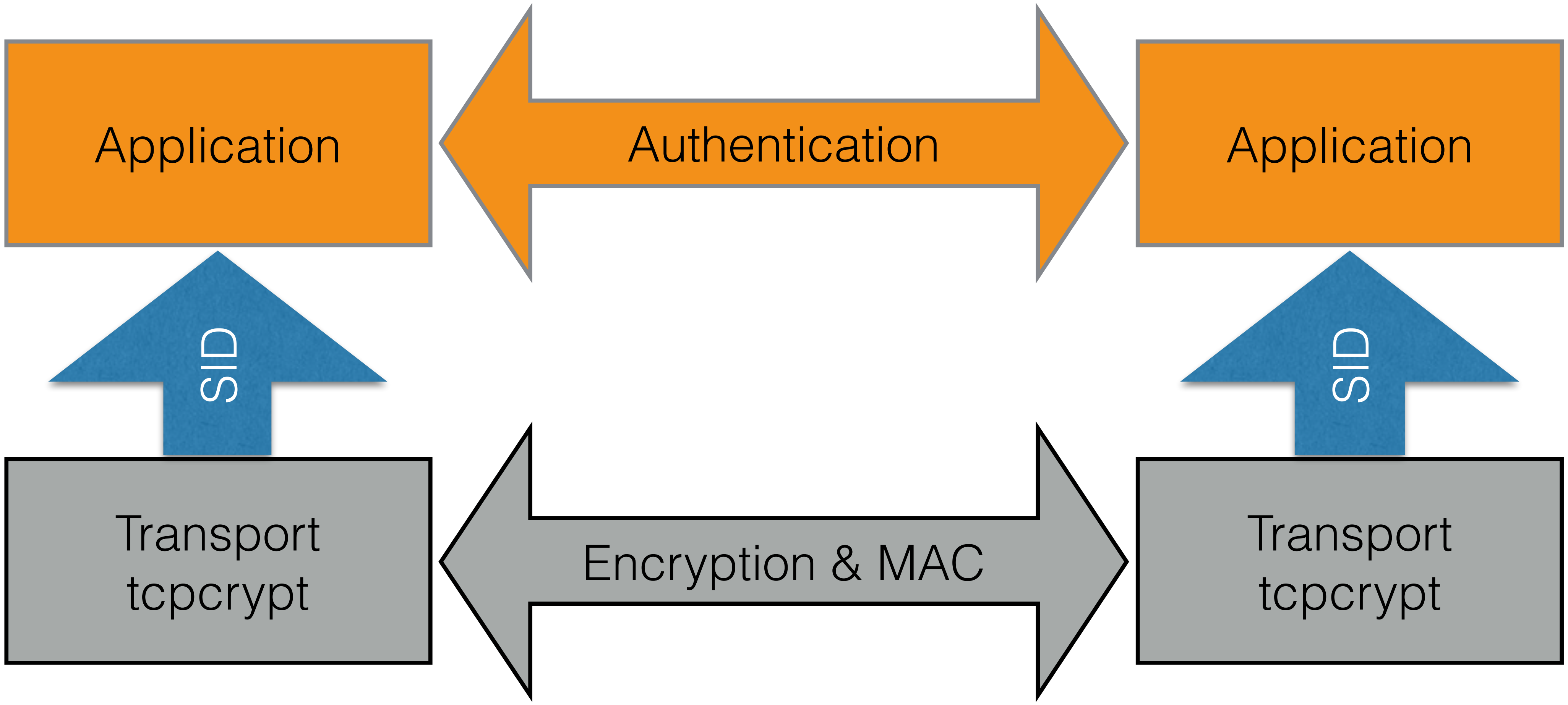
Session ID:
0xabcdef

Session ID:
0xdead

Application-layer authentication

E.g.: HMAC(password, SID)

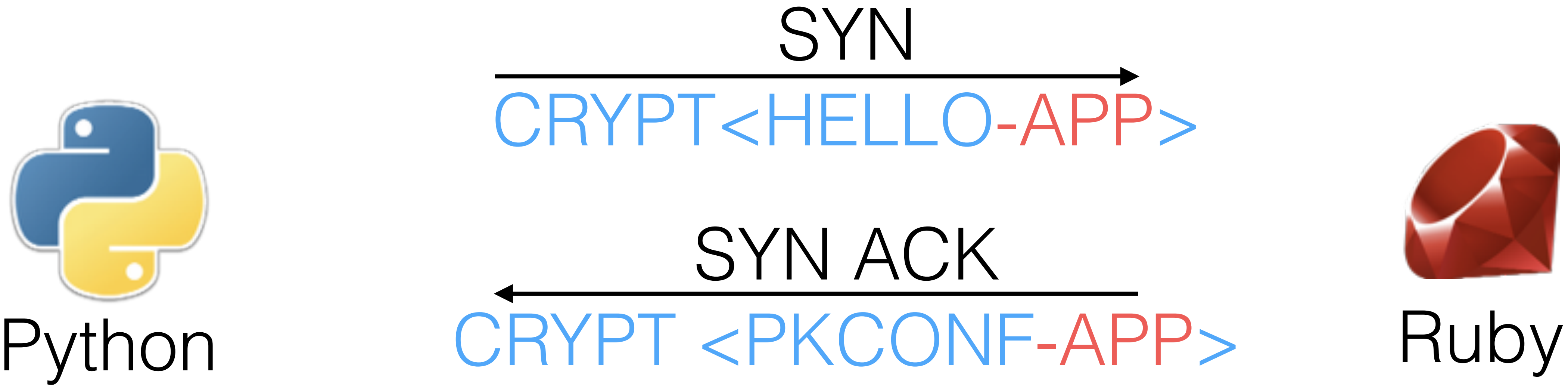
getsockopt
SESSION_ID



Application-aware bit
E.g., DANE integration

```
…
s = connect("hello.com")
send(s, "GET / HTTP/1.0\n\n")
…
```

- Get certificate from DNS
- Sign tcpcrypt SID


SYN

CRYPT<HELLO-APP>

SYN ACK

CRYPT <PKCONF-APP>

Feedback from last IETF

~~0. Why encrypt at the TCP layer?!?~~

1. Must never delay or break TCP connections.

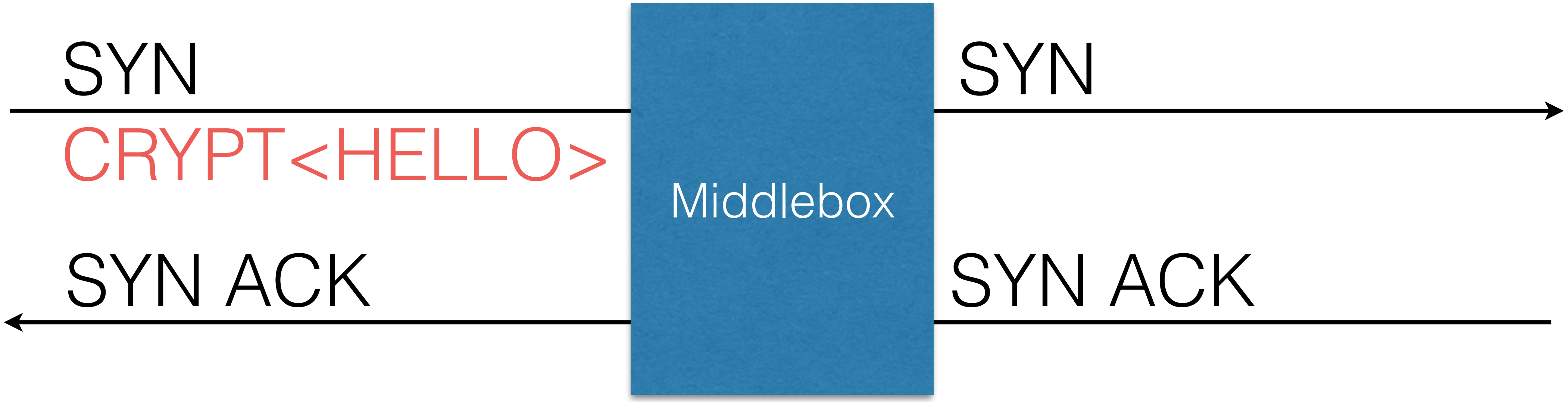
2. Want more data on middlebox behavior.

Previous middlebox compatibility measures

- Fallback to TCP if CRYPT option stripped.
- Do not MAC source port, destination port, timestamp option, etc.
- MAC sequence number and ack number offsets from ISN.
- Consistent retransmission of ciphertext. See Authenticated Sequence Mode encryption in draft.
- Integrity check of RST disabled by default.

Possible failure cases

Scenario 1: middlebox strips unknown options.

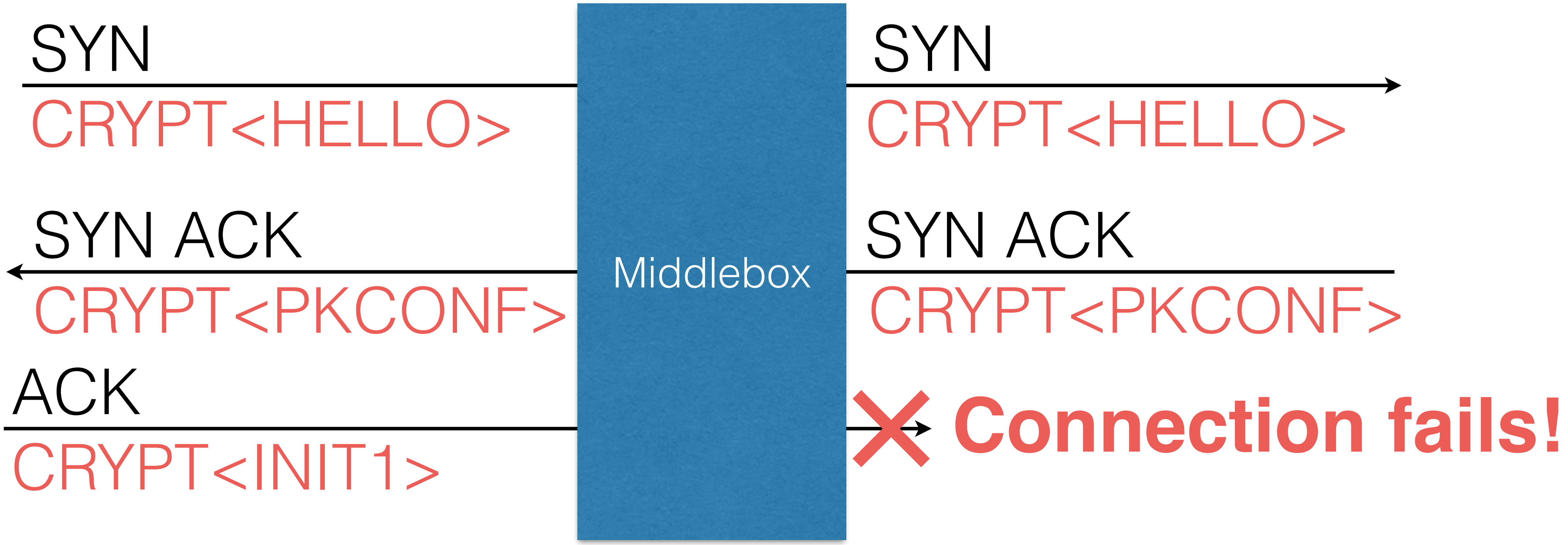


Result: no tcpcrypt, but TCP works fine ✓

Possible failure cases

Scenario 2:

- Middlebox leaves unknown options.
- Middlebox drops packets that don't conform to application-layer protocol.
- E.g., HTTP over tcpcrypt won't look like HTTP.



tcpcrypt check server

- tcpcrypt implementation performs following tests on ports 80 and 7777 of check.tcpcrypt.org

  1. Sends a GET request on plain TCP connection.

  2. Sends non-HTTP data on plain TCP connection.

  3. Performs a tcpcrypt connection.

- If tests fail, tcpcrypt is disabled. No disruption to connections.

- Initially allows us to gather statistics on middlebox behavior.

Implementation status

- Easy to install user-space implementation. No kernel mods.
  - tcpcrypt protocol is amenable to implementation as a simple packet rewriter.
- Released a new stable Windows version.
- Official Debian package is being created.
- Runs on Windows, Mac, Linux, FreeBSD.
- 8,000 lines of code.

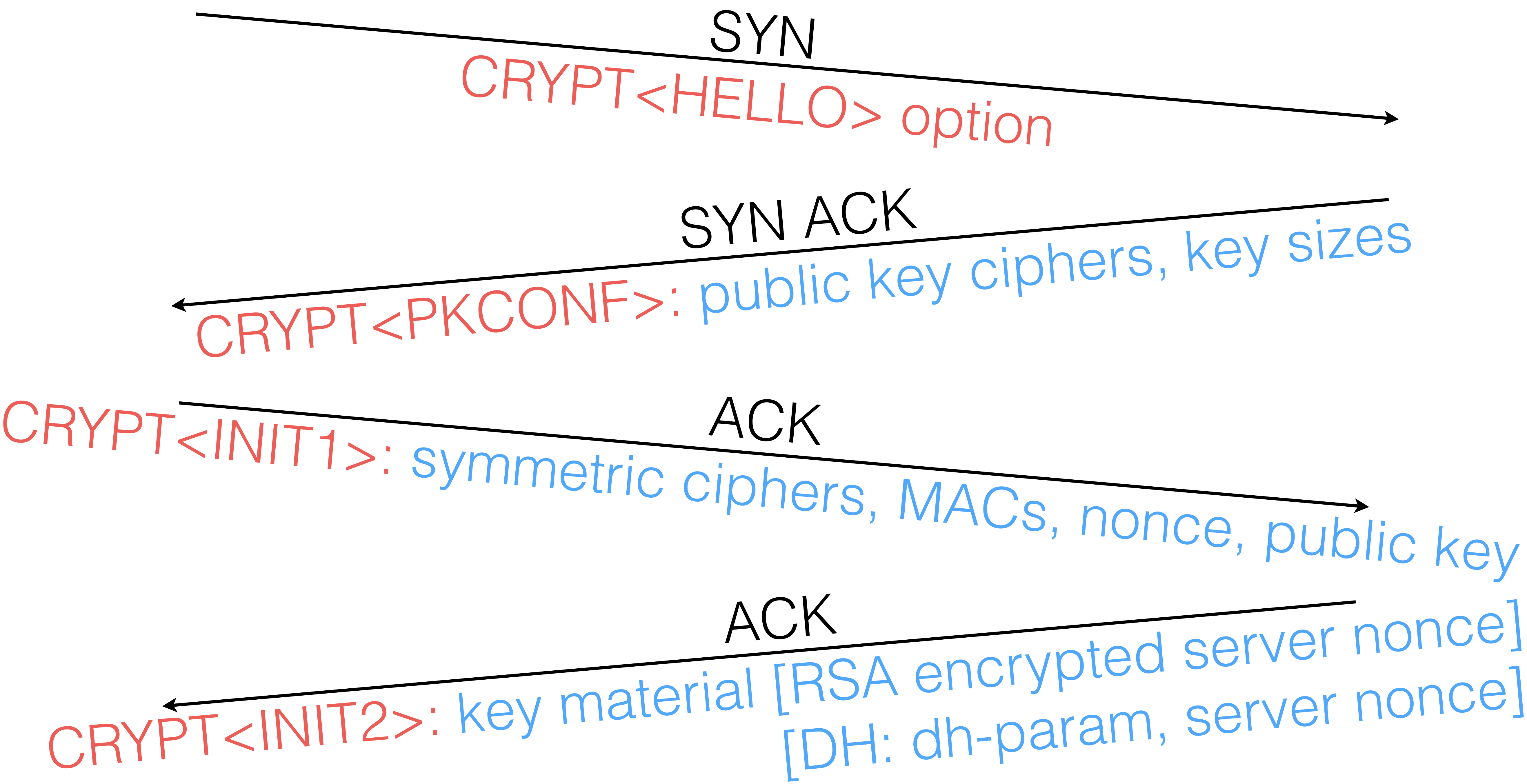
Key tcpcrypt properties

- Leverage TCP handshake to negotiate increased security.
- Opportunistic forward secrecy with no configuration.
- Provide applications a Session ID with which to authenticate connections.
- Provide applications with out-of-band signaling to negotiate authentication of Session ID.

http://tcpcrypt.org

Backup slides

Handshake



SYN

CRYPT<HELLO> option

SYN ACK

CRYPT<PKCONF>: public key ciphers, key sizes

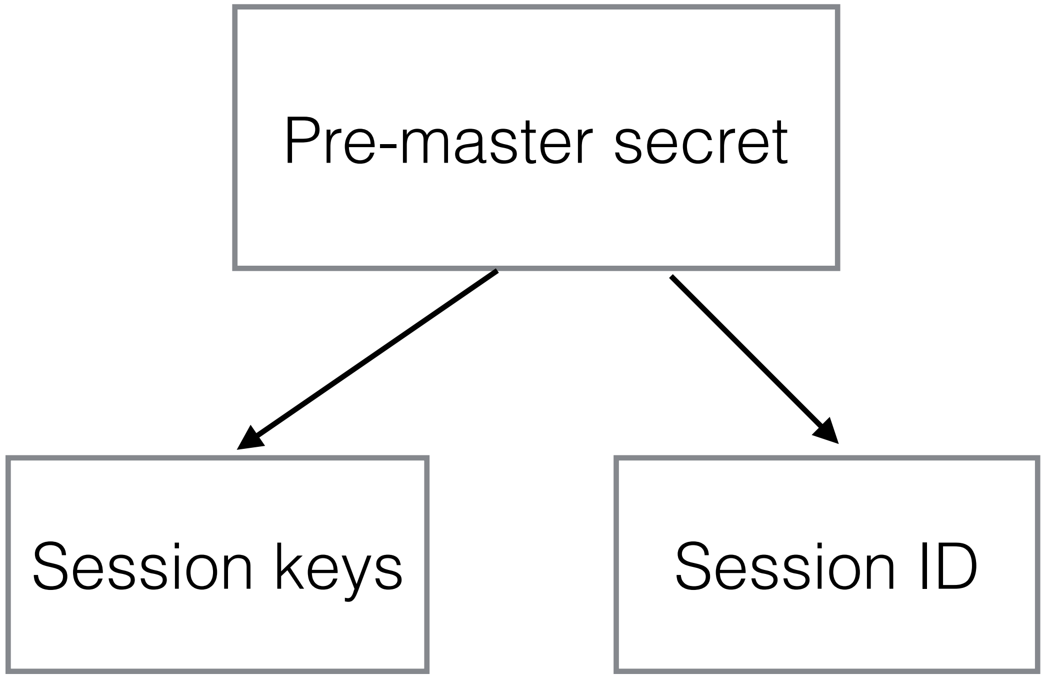
ACK

CRYPT<INIT1>: symmetric ciphers, MACs, nonce, public key

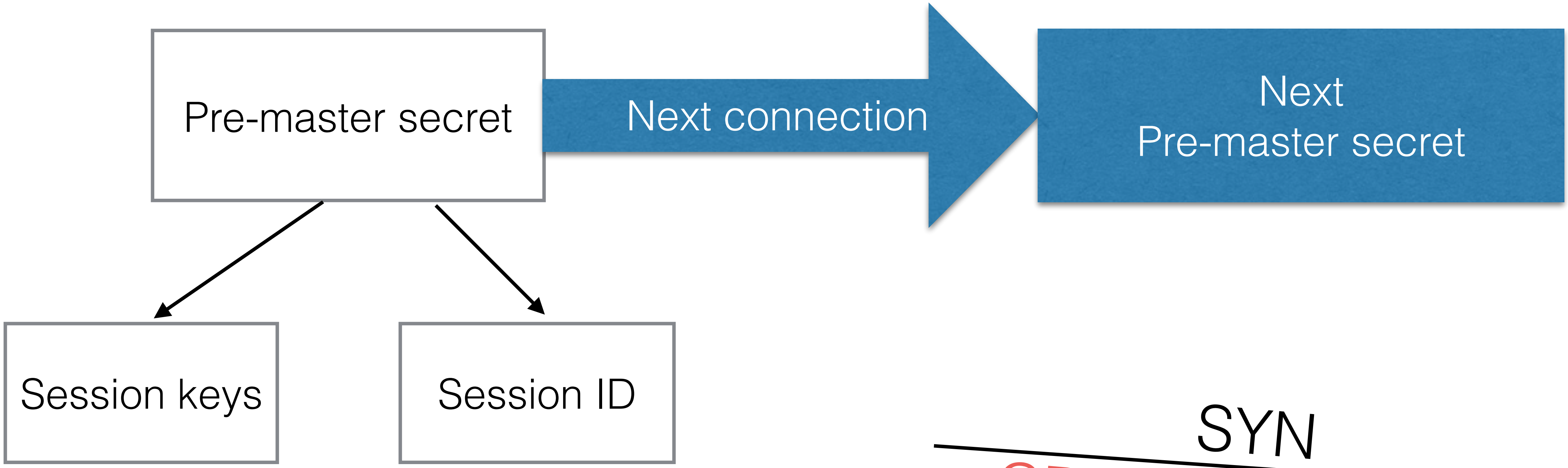
ACK

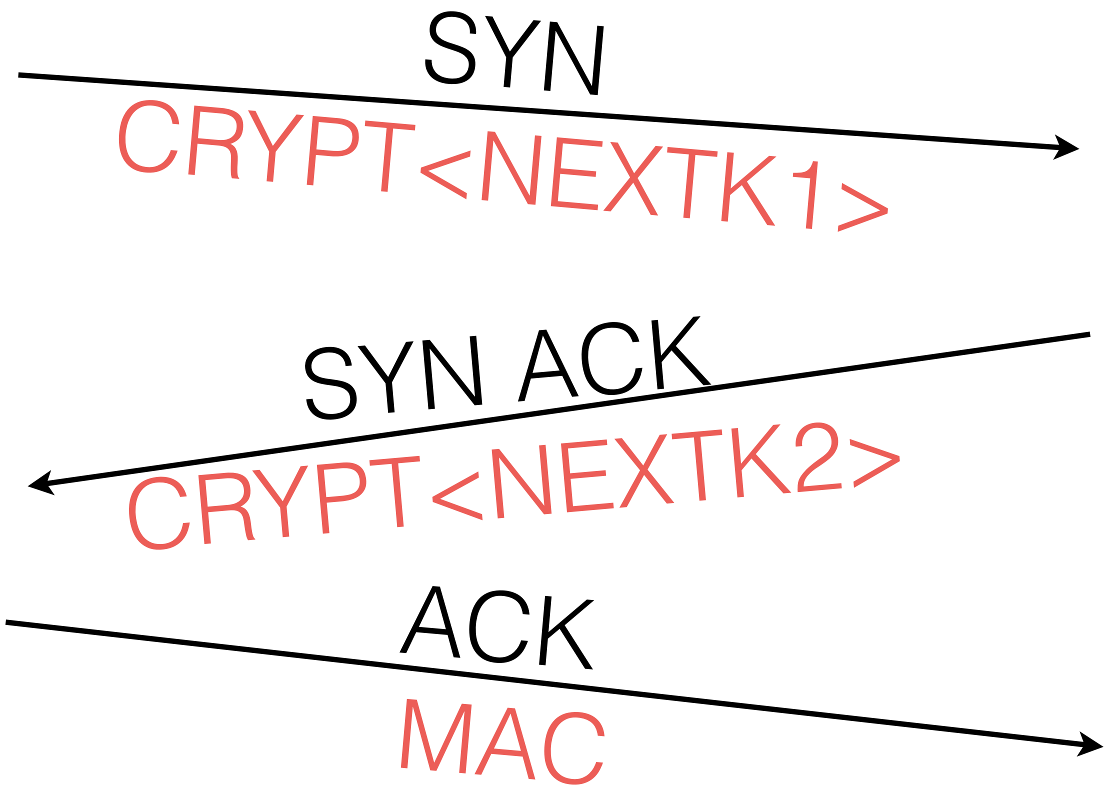
CRYPT<INIT2>: key material [RSA encrypted server nonce]
[DH: dh-param, server nonce]

Session cached handshake



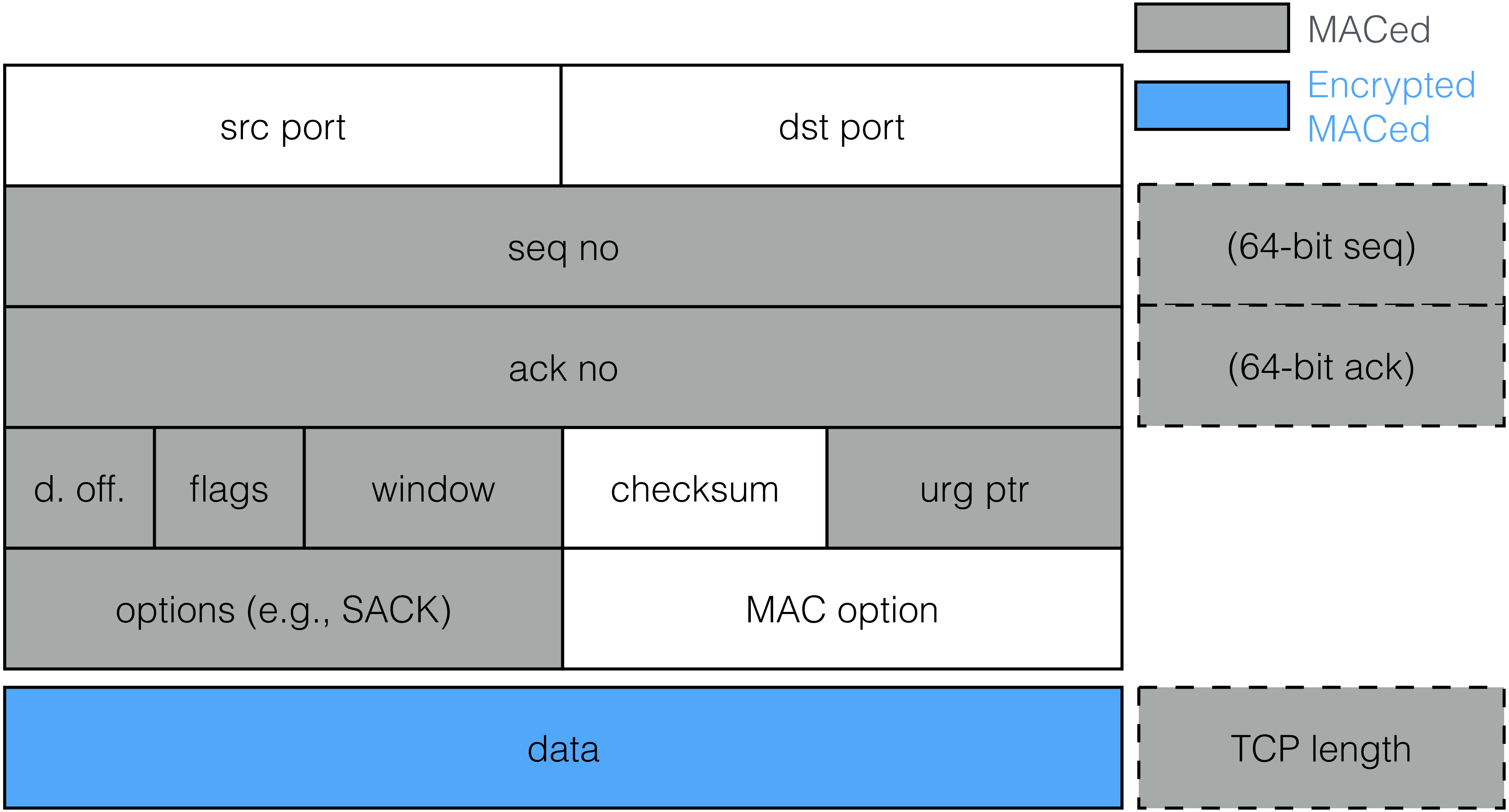
Session cached handshake





Low latency!

MAC and encryption



MACed

Encrypted MACed

src port

dst port

seq no

(64-bit seq)

ack no

(64-bit ack)

d. off.

flags

window

checksum

urg ptr

options (e.g., SACK)

MAC option

data

TCP length

tcpinc requirements

No modifications to upper layers.	Yes
Forward secrecy with per-connection granularity and integrity protection	Yes
NAT and firewall traversal.	Yes
Key rollover without significant impact.	Yes
Lower overhead than stacked solutions.	Yes
No manual configuration.	Yes
Crypto agility.	Yes
Fallback to TCP.	Yes
Minimize option space especially in SYN segments.	Yes
Must not require authentication but must provide hooks for authentication.	Yes
No extra linkability by third party eavesdroppers.	When session cache disabled
Client has option to defeat fingerprinting.	Yes - modulo configuration