tcpcrypt

Andrea Bittau, Michael Hamburg, Mark Handley,
David Mazières, Dan Boneh

Stanford University, University College London

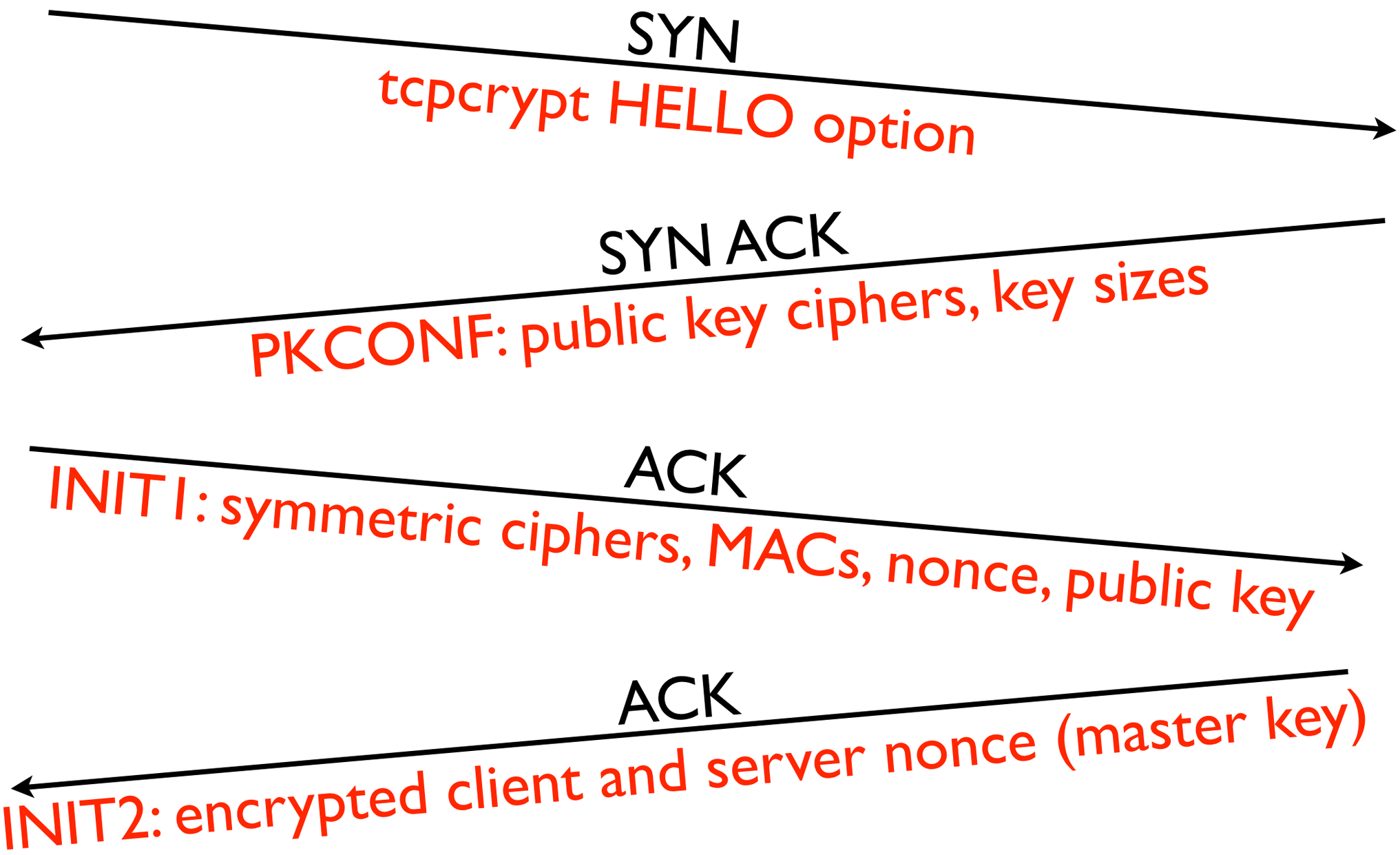
Goal: encrypt most TCP traffic

- Zero configuration, works with NATs.
- Integrate with app-level authentication
- High performance, especially on servers.
- Avoid double encryption.

Maximize security for each scenario

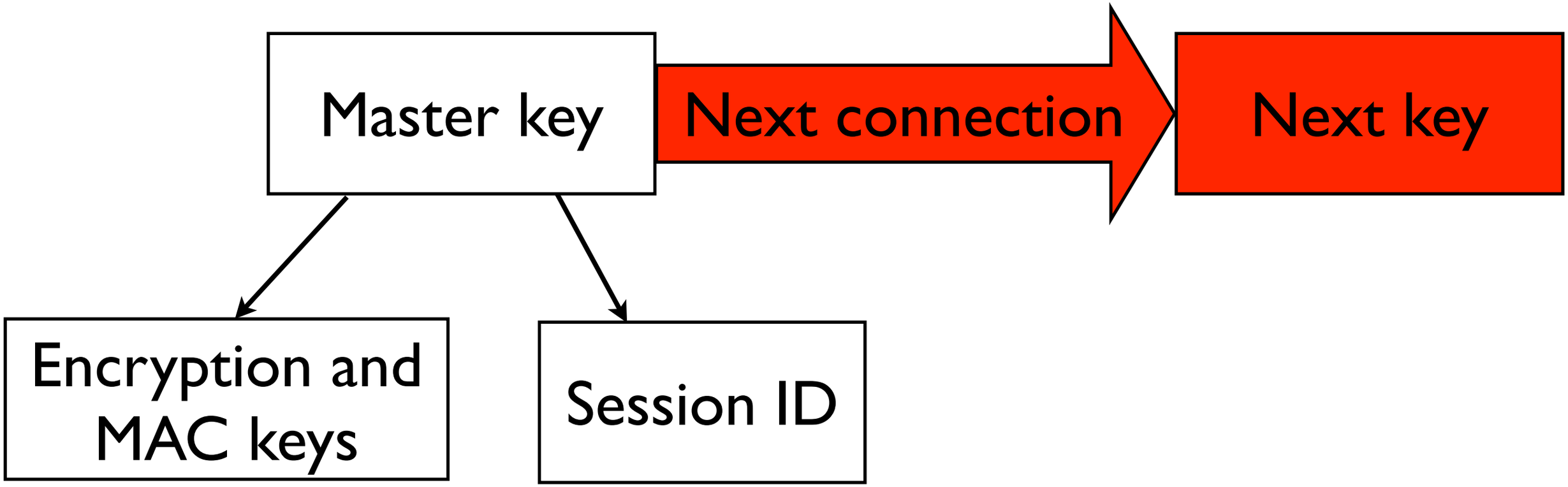
Use case	Preconfiguration	Today's security	Possible security
News site	None	None	No passive eavesdropping
Online shop	Server certificate	Server auth	Server auth
Forum	Shared secret (cookie) no server certificate	None	Mutual auth
Banking	Shared secret and server certificate	Mutual auth if cert and pass OK	Mutual auth if pass OK

tcpcrypt handshake

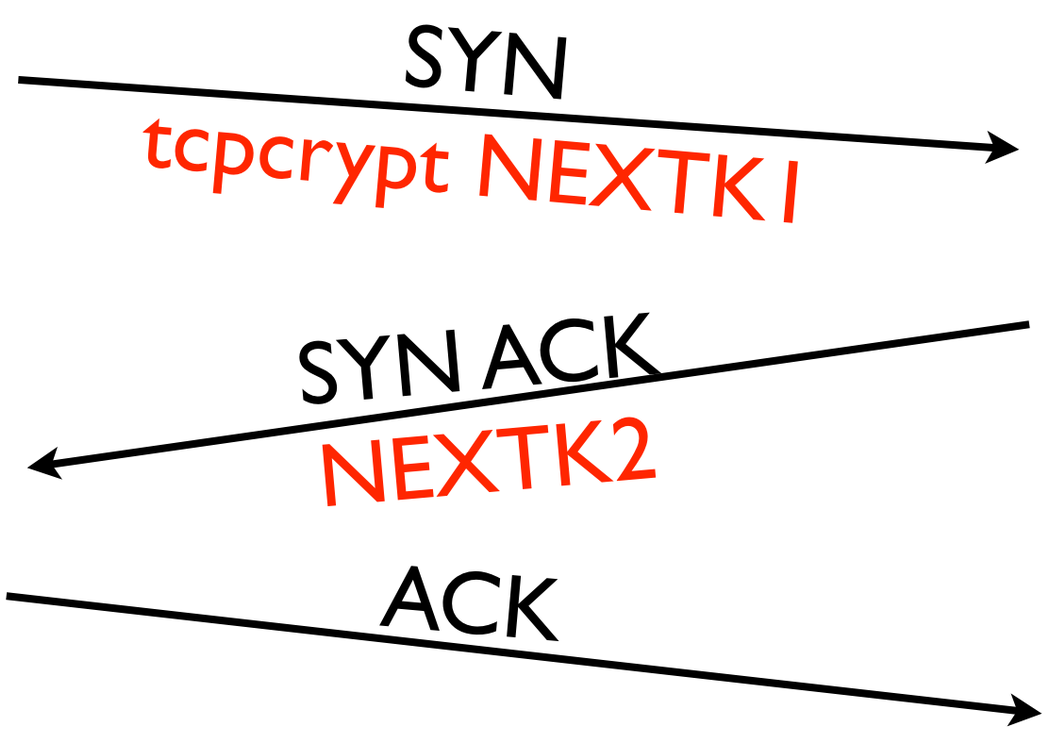


INIT1/2 don't fit in SYN / ACK: sent as data invisible to apps.

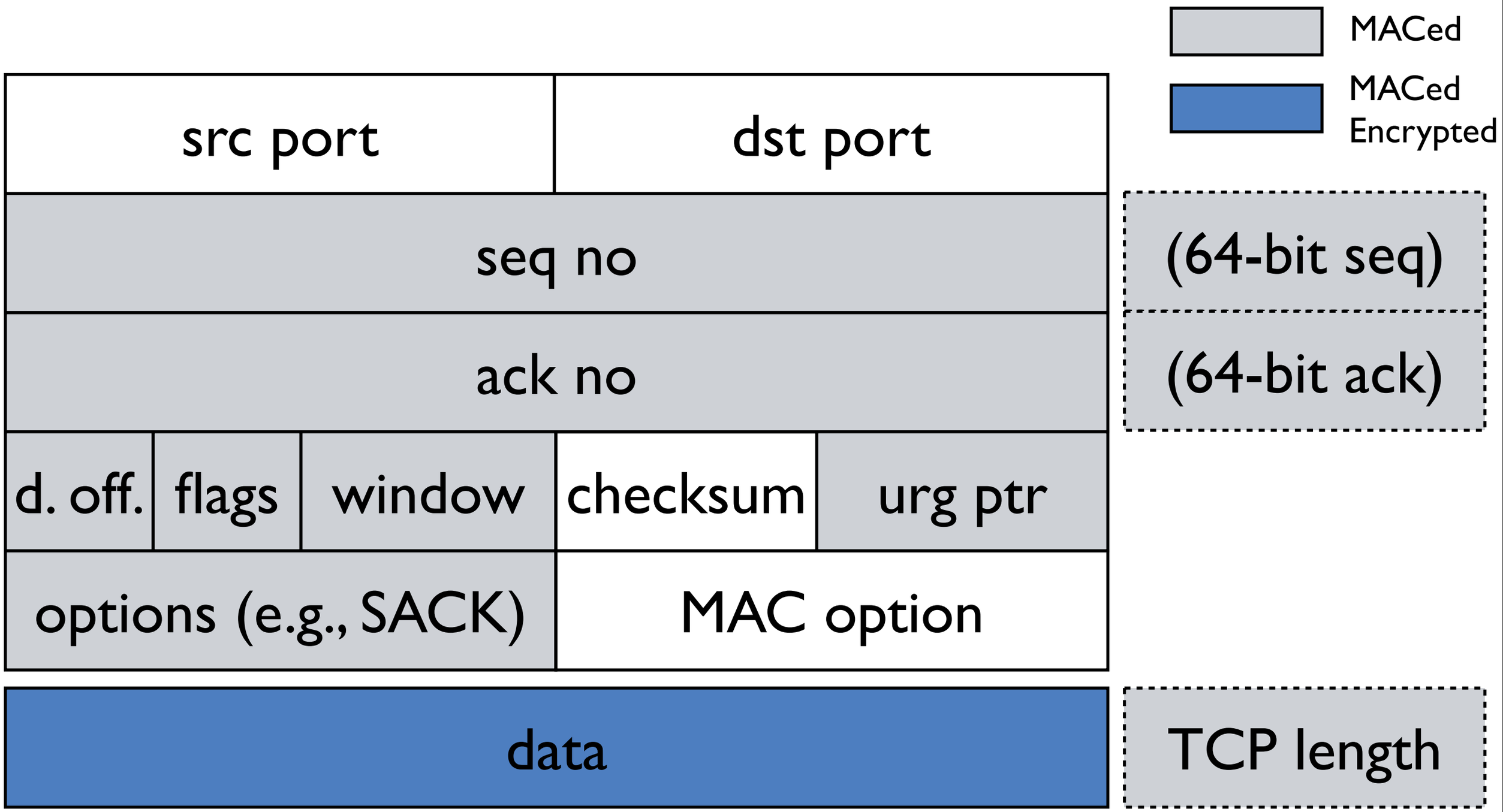
Session cached handshake



Low latency!



MAC and Encryption



tcpcrypt semantics

```
getsockopt(s, SOL_TCP, TCP_SESSIONID,
           &sid, sizeof(sid));
```

- If session ID is equal on both endpoints, no man in the middle.
- Authenticating session ID authenticate connection:
  - E.g., sign SID with cert, HMAC with cookie, password (PAKE), ...
  - Can also log and check after the fact.

High performance

- Up to 25x higher connection accept rate than SSL on servers.
- Near TCP connection latency for session cached tcpcrypt connections.
- 9Gbit/s using AES+UMAC with AES-NI

Conclusion

- Encryption is general-purpose and practical to enable by default.
- Benefits of encrypting at transport layer:
  - Backwards compatible (e.g., NATs).
  - Benefits legacy apps.
  - Natural granularity for authentication.
  - Leverage existing handshake for negotiation

http://tcpcrypt.org