Grease

QUIC Down Under

“Anything that isn’t an invariant might change”

But really, what have we done to ensure that this is possible?

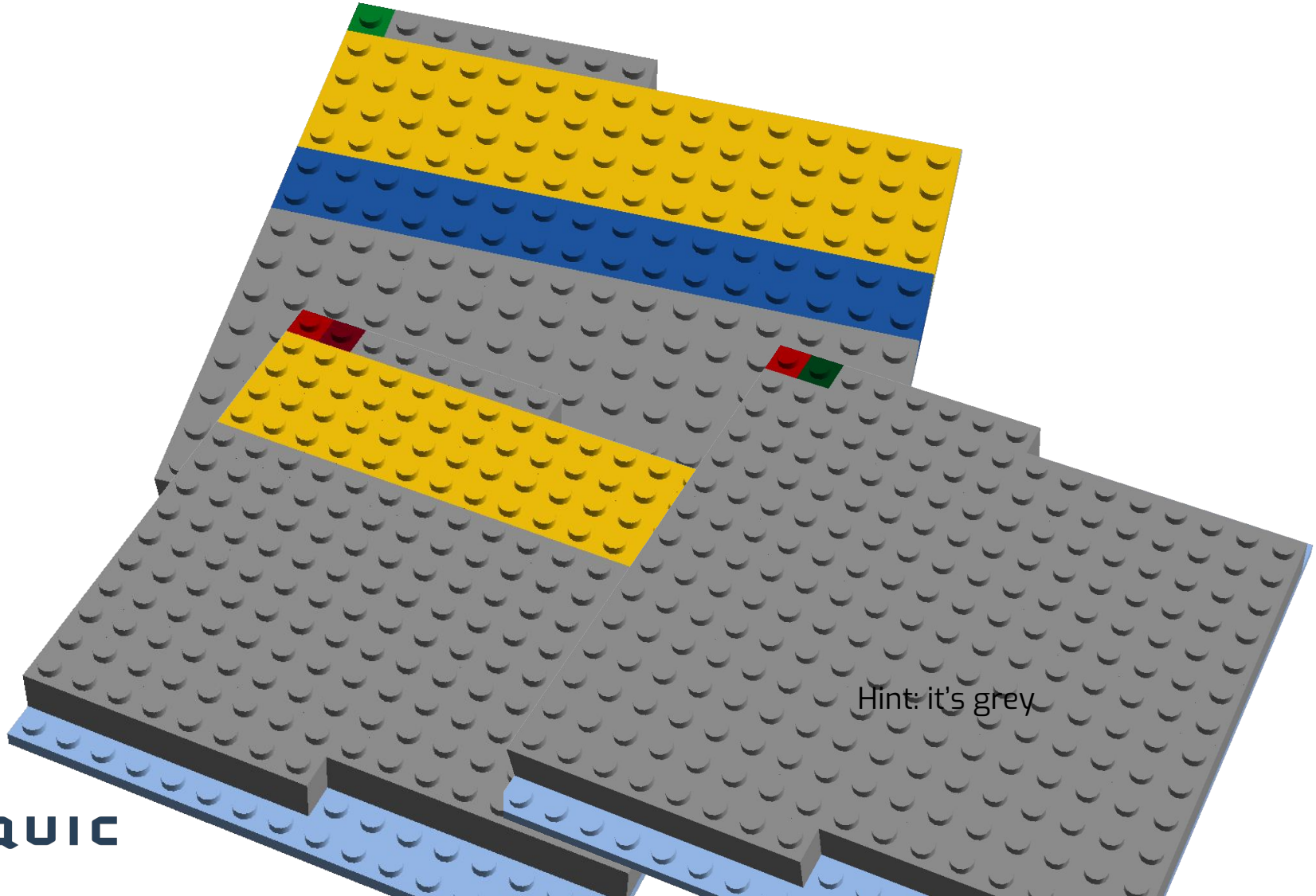
***ossification**, n.*

The hardening or calcification of soft tissue into a bonelike material.

This is what you get when you deploy a protocol and later discover that the network won’t let you change something

If you don’t believe that this is a problem:
https://youtu.be/_mE_JmwFi1Y

What can we defend?



QUIC -08 status

Version-specific encryption for handshake packet payload

- need to know the key to get plaintext
- need to know the version to get the key and cipher

Some bogus versions reserved

TLS can be greased (though it might not be necessary)

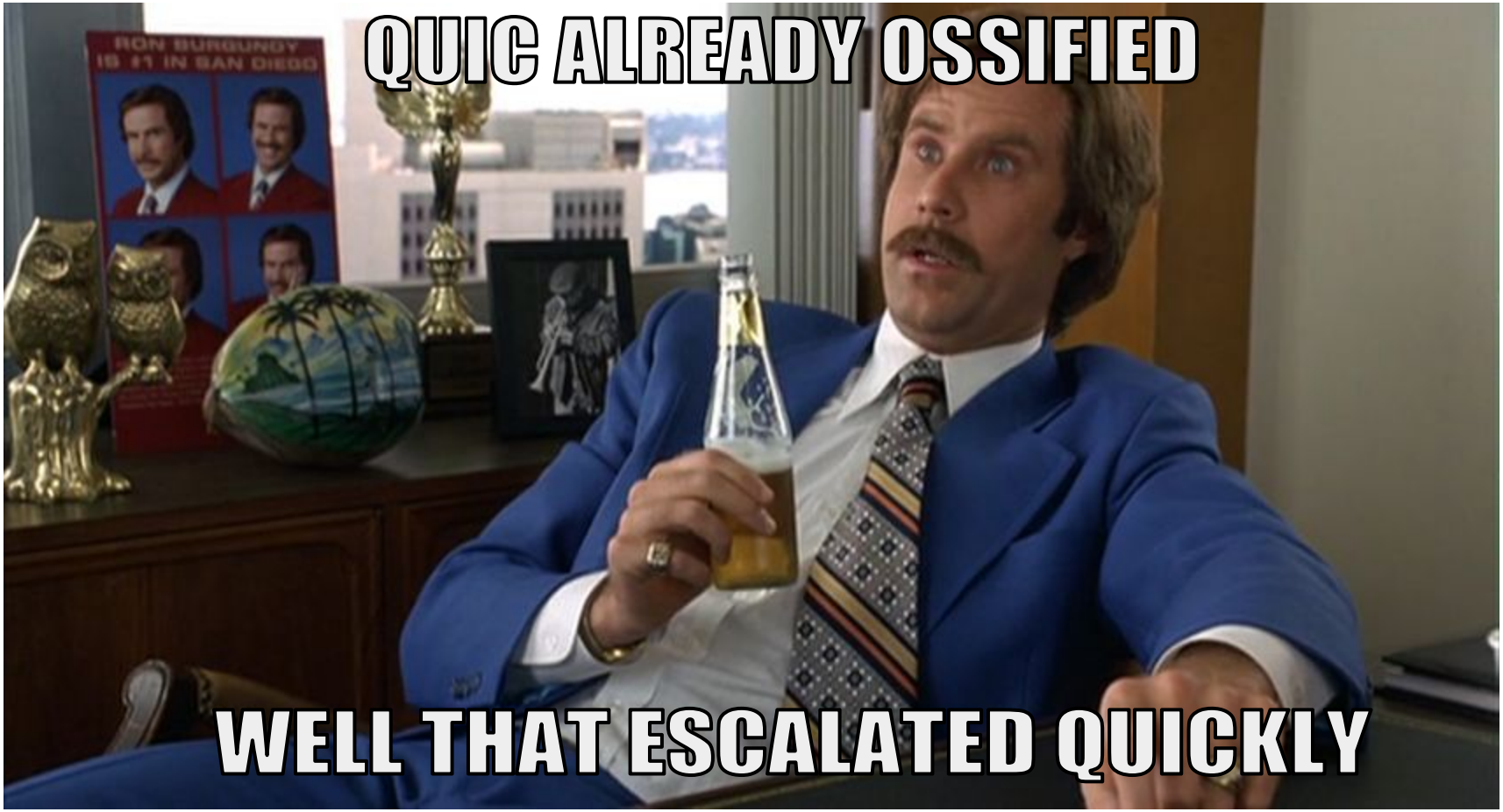
Notable exceptions: **packet numbers** increase monotonically, and packet **type** is unencrypted

Why?

It is important to preserve our ability to make changes at some time in the future

But if something never changes in practice, then it might not be changeable in practice

Why?



Principles

Change things all the time

- Ideal: encrypt, but this turns out to be hard
- Maybe: vary for every packet (hard for some things)
- Good enough: vary for every connection

No more simple mappings of codepoint to semantic

Create incentive to understand the protocol

- **Defend against Murphy, not Machiavelli**

#1043 - obfuscation of packet number and type

Uses the modern variant of the Caesar cipher - Vigenère

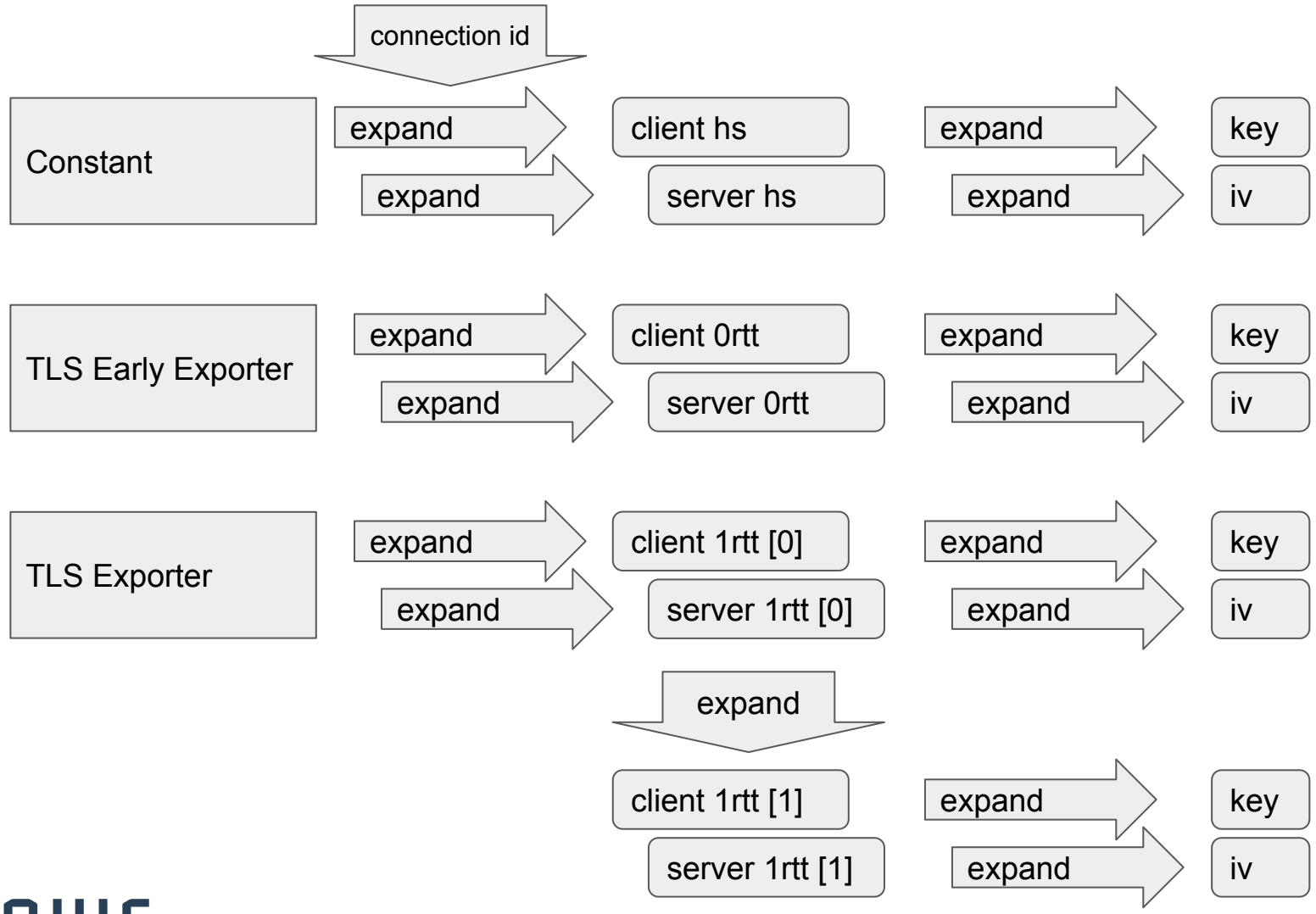
The secret is constant during the handshake

Integrated into packet protection after handshake

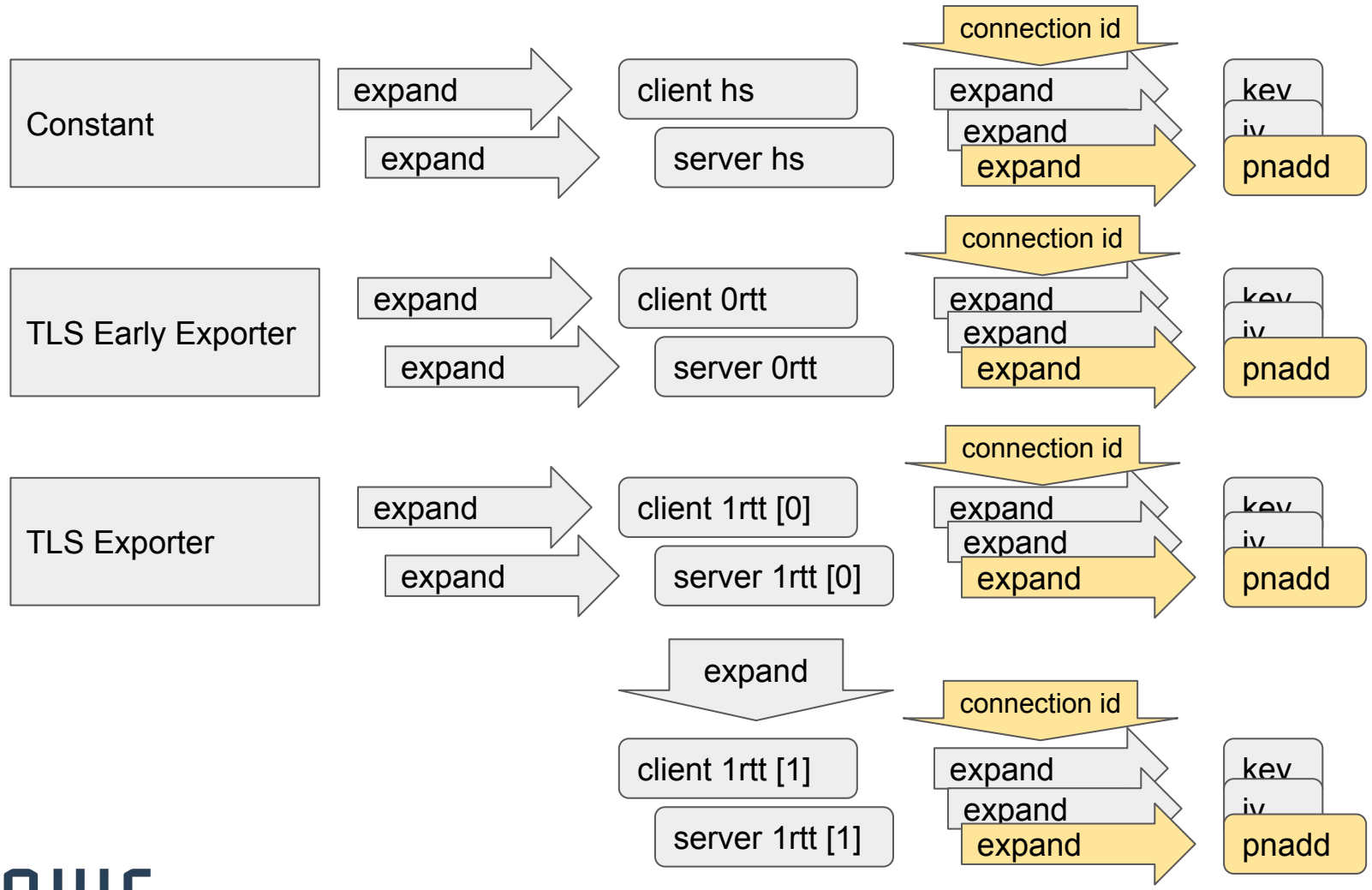
Key takes connection ID and endpoint role as input

Packet numbers start at zero (no odd randomization)

Key Schedule - Old



Key Schedule - Updated



Collateral

No more funny randomization (good)

- ACK frames are smaller initially (good)
- largest_acknowledged was 8 octets ~75% of the time

This uses the entire space of values for Type

- That affects multiplexing

0-3	STUN
16-19	ZRTP
20-63	DTLS
64-79	TURN Channel
128-191	SRTP
0-255	**QUIC**
64-255	**QUIC w/o CID**

This leaves the really hard stuff

Connection ID - stability is all that is important, so this is OK

The long/short bit - invariant, very hard

Version negotiation - invariant, also very hard

Packet timing and size - traffic analysis resistance is hard

Monotonically increasing packet numbers - not for discussion until we resolve the spin bit issue; if we don't do that we might use a simple PRP to strengthen this