Per-protocol tag for CBOR

On storing CBOR encoded items on stable storage

draft-ietf-cbor-file-magic-01

Michael Richardson
<mcr@sandelman.ca>

Motivation

- Observed CBOR encoding of “certificates”, private keys, etc.
  - PKIX already has poor extension of “.pem”, which could be mostly anything at all. Wish not to repeat.
- CBOR objects get stored on disk, want a way to identify them which is very very easily removed prior to transmission.
  - Content which is part of constrained device firmware, will still get stored on build system/source code.

Motivation (2)

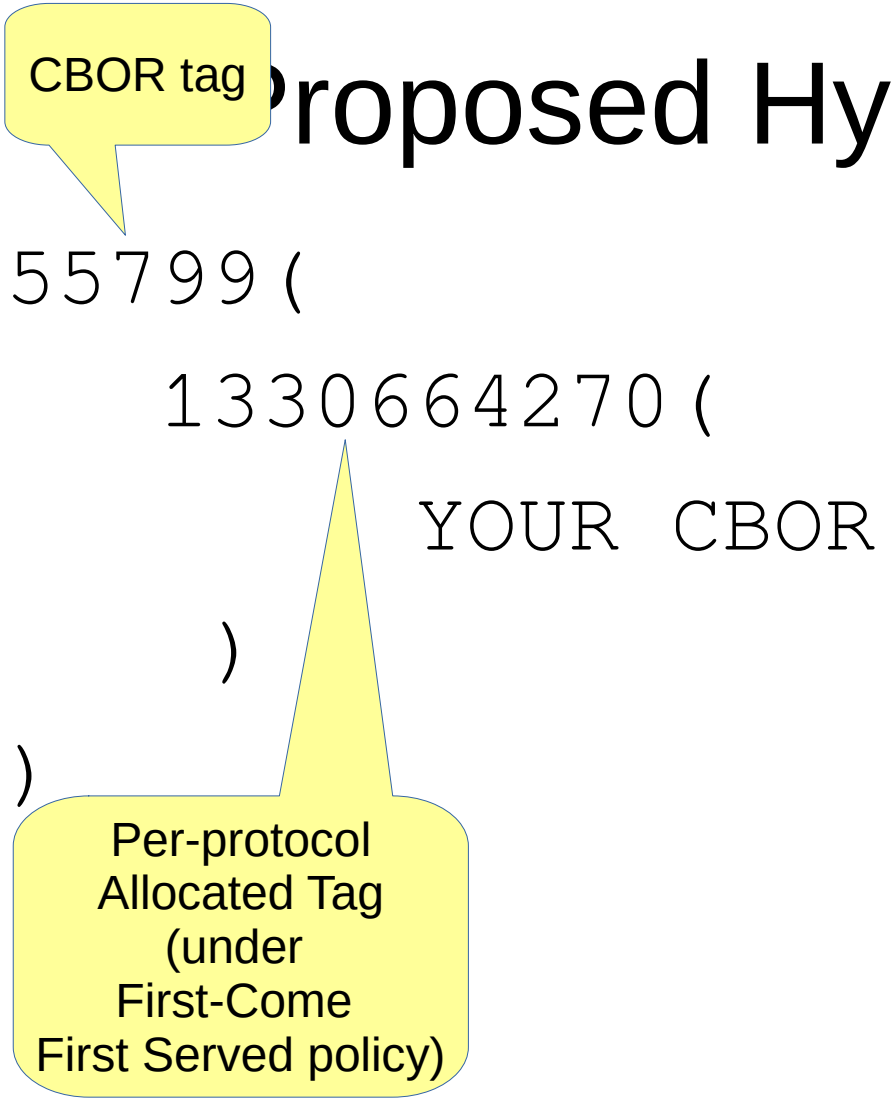
- Author converted bespoke private encoding to CBOR
  - To eliminate need for cross-language binding for IPC
- Use CBOR Sequence, can be easily removed before transmitting
- Dissastisfied with:

```
%file OUTPUT/ikev2client.record.x86_64
OUTPUT/ikev2client.record.x86_64: Concise Binary Object Representation
                    (CBOR) container (array) (tagged)
```

Changes since Adoption call

- Adopted March 3, 2021
- use both CBOR Sequence and straight CBOR Tag
- needs new tag allocation for CBOR Sequence
  - proposed 55800
- documented the Unicode/UTF-8/UTF-16 considerations for 55799 and 55800 in more detail.

Proposed Hybrid solution (1)

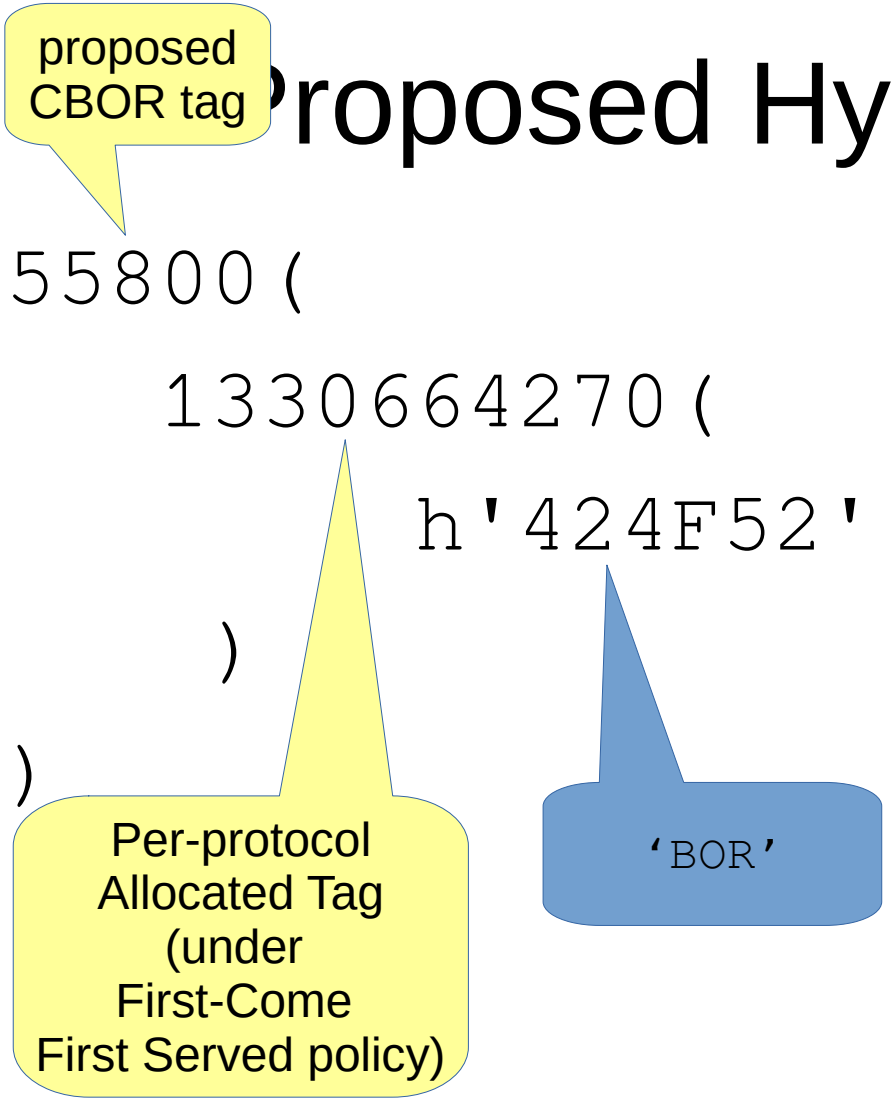


Initial 8 bytes are unique

Tag naively stays with the data when transmitted.

Tag can be removed by aware software.

Proposed Hybrid solution (2)



Initial 12 bytes are constant.

File contains ASCII “CBOR” when examined.

YOUR CBOR follows as second sequence.

Easy to carve out CBOR without speaking CBOR.

Conclusion

```
%file OUTPUT/ikev2client.record.x86_64

OUTPUT/ikev2client.record.x86_64:
Openswan WHACK file (CBOR encoded)
```

Simple proposed BCP

Adopt and publish