Roughtime at IETF 112

The problem we’re trying to solve

- Need time source accurate to within a few seconds
- Capable of being used for certificate validation
- 30% of certificate errors are caused by inaccurate client clocks

Issues

1. Short tags at end moves PAD field, complicating packet construction
2. MISP: what is moment of processing?
3. ROUGHTIM is waste of bytes
4. Mandate one of TCP or UDP
5. 32 bit offsets too big
6. 32 bit pair size too big
7. SHA-512 mandate
8. Add test vectors
9. Milli versus microseconds
10. RFC 3161
11. No problem statement

More issues

12. Precision too high
13. Leap seconds, DTAI-UTC unnecessary
14. Signatures are slow
15. Microseconds exceed 32 bits, problem on microcontrollers
16. Not clear how to use as only timekeeper

Responses

We think many of the issues can be solved through adding additional text/are editorial.

We’ll move to milliseconds everywhere, shrink fields accordingly

Timestamping is possible, will add section

Some deserve more discussion

SHA-512 only

- Rationale: cross-protocol signing is bad
- Ed25519 uses SHA-512 internally
- Need complete agreement on support
- Doesn’t change rollout of change
- See also PKIX where signature algorithms limited by same reason
- Counterational: extensibility good, cross-protocol more theoretical than real argument

GREASE

- Rationale: use it or lose it. Long and bitter TLS experience

Slow signatures

- Ed25519 is among fastest signature algorithms for signing and verification
- RSA would be faster verification but much slower signing
- In context of TLS chain verification not big change, batchable

Only timekeeper

- Same as NTP, but will be less accurate (few milliseconds off)
- Need to do experiments

Leap seconds, DTAI-UTC unnecessary

- Leap seconds, TAI-UTC difference, and UT1-UTC difference tags are OPTIONAL for both server and client.
- Current lack of standardized distribution schemes for this data.
- ITU-R TF.460-6 recommends that TAI-UTC and UT1-UTC are included in “time-signal emissions”.