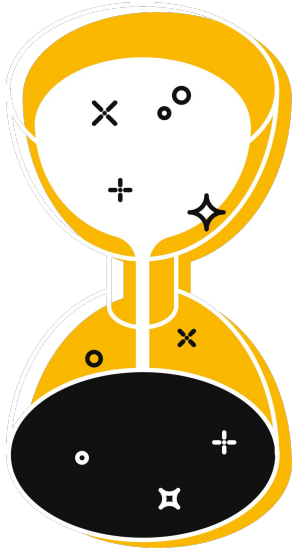
Roughtime: Securing time for IoT devices

Christer Weinigel, Netnod

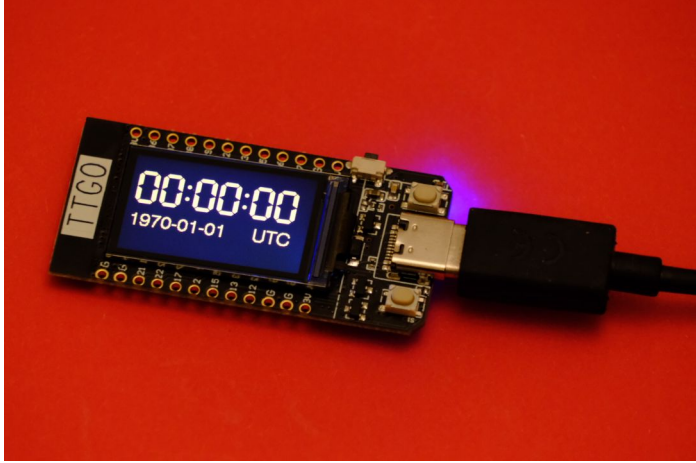
Why accurate time is important

- Many security critical protocols need accurate time
  - DNSSEC, secure domain name lookups
  - TLS, the basis of many other protocols
    - HTTPS, everything on the web
    - SMTPS, IMAPS, POP3S, secure mail
  - Accuracy requirement: within a few minutes or hours
  - Risks of not having accurate time
    - Fall back to insecure algorithms
    - Use old (maybe leaked) information
- The application itself might need time
  - Example: electronic door lock
  - May need more accurate time than minutes or hours

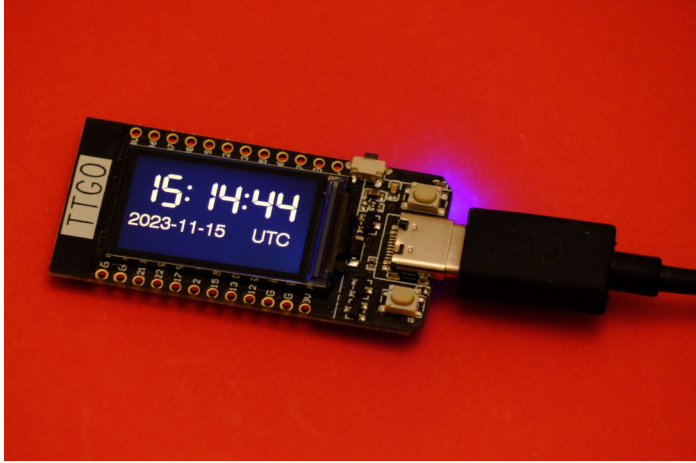


Keeping time

- All devices can keep time
  - When powered on
- But not when powered off
  - IoT devices may not have a Real Time Clock (RTC)
  - Raspberry Pi - has RTC hardware, but no battery backup by default
  - "Shipping mode"
    - Even with a battery the clock will not run before first power on because the battery is not connected
- "Ten year on the shelf problem"
  - A device can sit on a shelf for a long time before being turned on



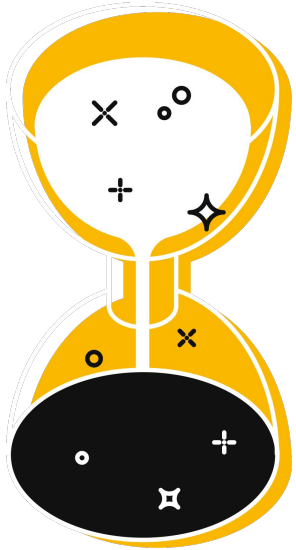
Getting time over the network



- NTP (Network Time Protocol)
  - Lacks security
- NTS (Network Time Security)
  - Adds security
  - Bootstrapping problem
    - NTS depends on TLS
    - Which depends on having accurate time
  - Heavyweight, not suited for resource constrained devices
- Others (e.g. HTTP/HTTPS date header)
  - No security, or depends on TLS and thus has the bootstrapping problem
- What if a time server fails or is compromised?
  - A common configuration for NTP is to use only one server
  - Single point of failure

Possible solution: Roughtime

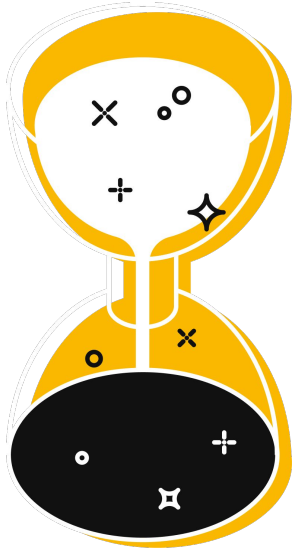
- Protocol is an IETF Draft
  - A. Malhotra, A. Langley, W. Ladd, M. Dansarie
- Started out as a way to solve the bootstrapping problem
  - Secure
  - Was not intended to replace NTP
    - Only 10 second accuracy
  - Fairly low CPU usage and small memory footprint



Roughtime: concepts

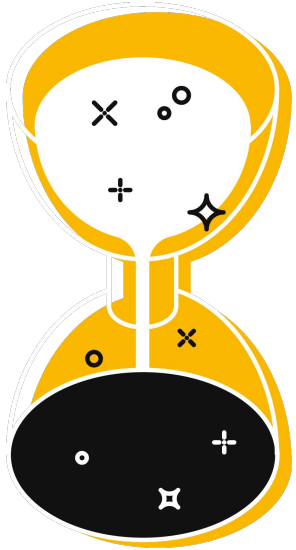
- Uses Ed25519 signatures, Merkle tree
- Hardcoded public keys
  - No bootstrapping problem
  - This is a trade off which turns it into a key distribution problem
- Client asks many servers for time
  - Requires consensus
  - Removes single point of failure
- Intended for devices where the server list can be updated
  - Or part of a firmware update

- These concepts could be used with other time protocols



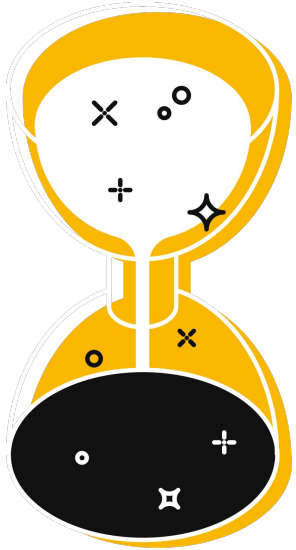
Roughtime: additional neat ideas

- A 32 byte nonce from the client is signed together with the timestamp
  - This is necessary for security anyway and is basically free
  - Allows signing of any document with a timestamp
  - A document can be the signed timestamp from another roughtime server
  - This can provide proof of misbehaving roughtime servers
  - Allows for accountability / auditing of roughtime servers
- Merkle tree reduces CPU load on the server
  - Ed25519 signing is a costly operation
  - Merkle tree spreads cost over multiple requests



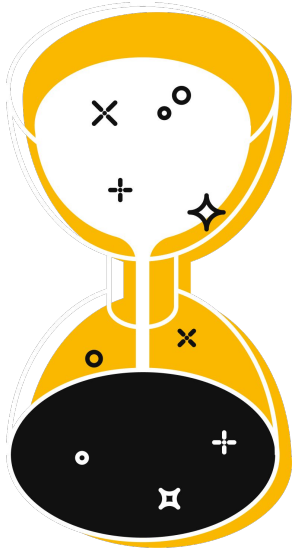
Roughtime: evolution

- It is now a decent generic time protocol
  - With significantly better accuracy than 10 seconds
    - Microsecond resolution
  - Which is secure (NTP is not)
  - Which can run on resource constrained clients (NTS is rather heavyweight)
  - Which still solves the bootstrapping problem



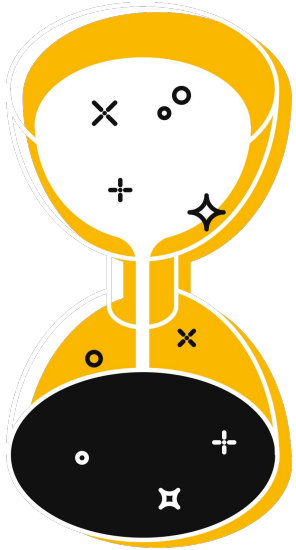
Roughtime: complexity

- Some of these new features conflict with the original goals
  - Microsecond accuracy of timestamps
  - Handles leap seconds and leap smearing
    - Required for sub second accuracy
  - Has support for multiple timescales
    - TAI - Temps Atomique International - time without leap seconds
    - UT1 - basically solar time - astronomical time
  - Not necessary if only 10 second accuracy is needed



Next steps

- Roughtime development has stalled
  - RIPE community funded project to revive it!
- Going forward
  - Kickstart work on protocol
  - Collect requirements
    - What do we need to secure time on IoT devices?
    - Getting community involvement and feedback
  - Update draft based on requirements
    - Add missing features, maybe drop unnecessary features
  - Update implementations
    - Hackathon
  - Submit Roughtime to IETF RFC Editors



Resources

- Roughtime Draft
  - https://datatracker.ietf.org/doc/html/draft-ietf-ntp-roughtime
- Working client implementation of draft version 4, 5 and 7
  - https://vadarklockan.readthedocs.io
- Roughtime servers
  - Netnod: sth1.roughtime.netnod.se, sth2.roughtime.netnod.se (v7)
  - Marcus Dansarie: roughtime.se (v7)
- Mailing list: "proto-roughtime"
- Blog posts with background about Roughtime
  - https://blog.cloudflare.com/roughtime/
- Contact me: wingel@netnod.se