NOT-QUITE-SO-BROKEN TLS 1.3

MECHANISED CONFORMANCE CHECKING

David Kaloper Meršinjak
Hannes Mehnert

University of Cambridge, Computer Labs

TRON, 21 February 2016

MOTIVATION

- Internet standards are written informal as RFC
- People interpret RFCs differently
- Primitives (HMAC, AES, ..) can be tested using static vectors
- Protocols include choice points
- State space explodes

TLS TESTING NOWADAYS

- Test against widely deployed implementations
- Using different command-line options to cover positive space
- Anybody automated tests of renegotiation/resumption?
- Interoperability with Widely deployed implementation
- Even if violating RFC

`be conservative in what you do, be liberal in what you accept from others` -- Postel

SOME PROBLEMS

- SignatureAlgorithms are not used for certificate selection
- Padding (client hello) included length (`servers MAY check`)
- Blocking semantics during renegotiation
- Early CCS: RFC does not state all preconditions for a message

CHOICE POINTS

- Fragmentation and padding
- Client: ciphersuites, extensions, signature algorithms
- Client: which keyshares to transfer?
- Server: version, key exchange (PSK? 0RTT?), ciphersuite
- Server: certificate chain (SigAlgs, ciphersuite, KeyShare, SNI)
- Server: encrypted extensions (pretty clear guideline)

OUR CONTRIBUTION

- Keep in mind: all records besides hellos are now encrypted!
- Provide tools for automated testing and analysis
- Support TLS implementors with tools for debugging

BACKGROUND: NQSB-TLS

- A clean-slate TLS 1.x implementation/model
- Started beginning of 2014
- Around 6000 lines of OCaml code
- Interoperates with major stacks
- Performance same ballpark as OpenSSL
- Protocol handler without side effects:
  - Transforms TLS state and input bytes to
  - Error OR
  - Ok (new TLS state, out bytes, decrypted payload)

STATUS OF NQSB-1.3

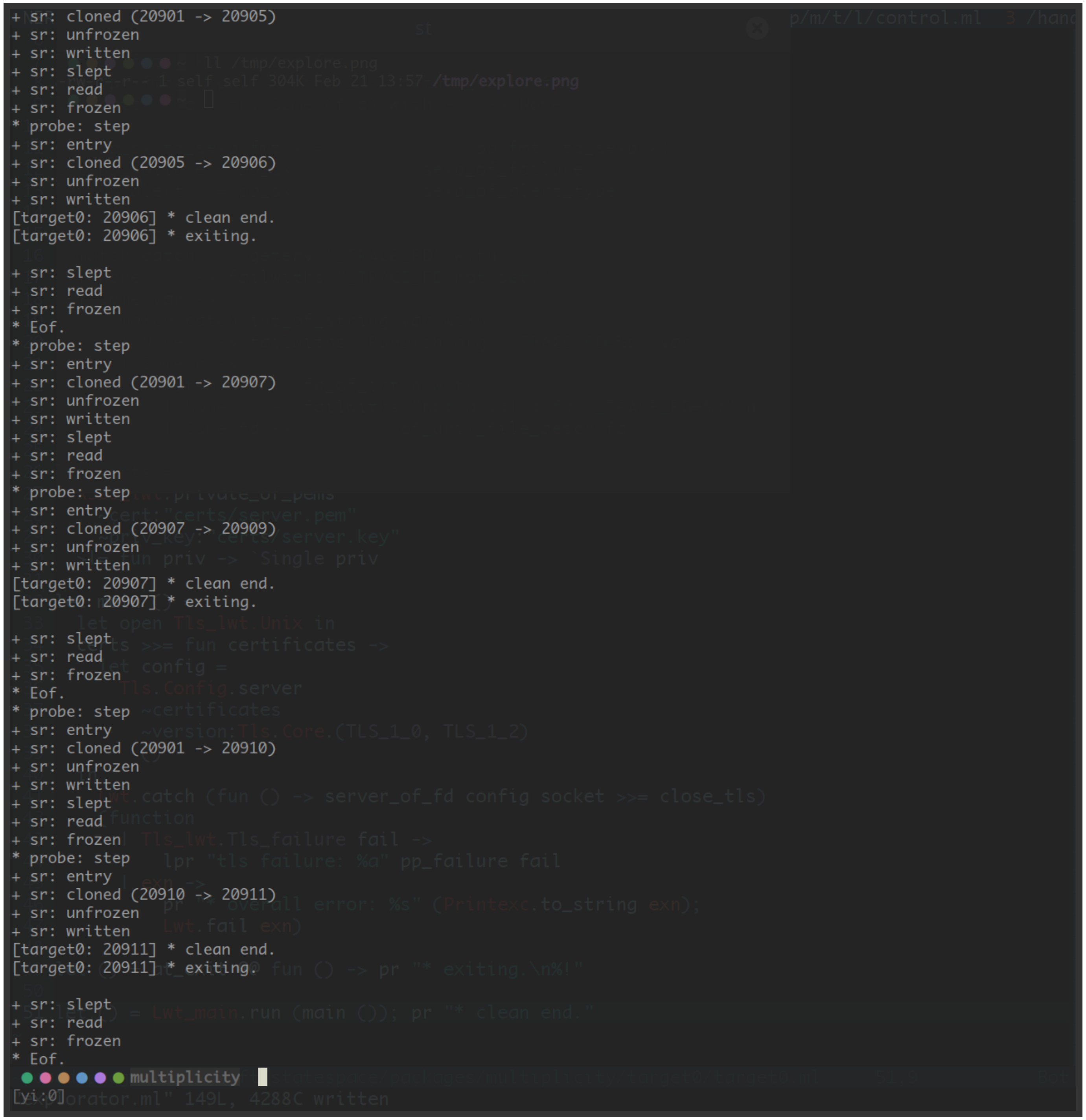
- loc: 1862 insertions, 358 deletions (now 6000)
- 1.3 state machine separate (apart from hello handling)
- draft11
- Can talk to itself :) (DHE, PSK, DHE-PSK)
- Missing 1.3 features: ECC, 0-RTT, client authentication

NQSB-1.3 TESTING TOOLS

- Check conformance of YourTLS by exploring its state space
- Render sequence diagrams from trace
- Replay recorded trace
- Validate session between any two stacks

CONFORMANCE CHECKING

- Explores state space by enumerating choice points in nqsb
- Executes unmodified YourTLS with all sequences of choices
- Covers space of valid interactions
- Reports sequences of choices which lead to failure



```
+ sr: cloned (20901 -> 20905)
+ sr: unfrozen
+ sr: written
+ sr: slept
+ sr: read
+ sr: frozen
* probe: step
+ sr: entry
+ sr: cloned (20905 -> 20906)
+ sr: unfrozen
+ sr: written
[target0: 20906] * clean end.
[target0: 20906] * exiting.

+ sr: slept
+ sr: read
+ sr: frozen
* Eof.
* probe: step
+ sr: entry
+ sr: cloned (20901 -> 20907)
+ sr: unfrozen
+ sr: written
+ sr: slept
+ sr: read
+ sr: frozen
* probe: step
+ sr: entry
+ sr: cloned (20907 -> 20909)
+ sr: unfrozen
+ sr: written
[target0: 20907] * clean end.
[target0: 20907] * exiting.

+ sr: slept
+ sr: read
+ sr: frozen
* Eof.
* probe: step
+ sr: entry
+ sr: cloned (20901 -> 20910)
+ sr: unfrozen
+ sr: written
+ sr: slept
+ sr: read
+ sr: frozen
* probe: step
+ sr: entry
+ sr: cloned (20910 -> 20911)
+ sr: unfrozen
+ sr: written
[target0: 20911] * clean end.
[target0: 20911] * exiting.

+ sr: slept
+ sr: read
+ sr: frozen
* Eof.
 ● ● ● ● ● ● multiplicity  █
[yi:0]
```

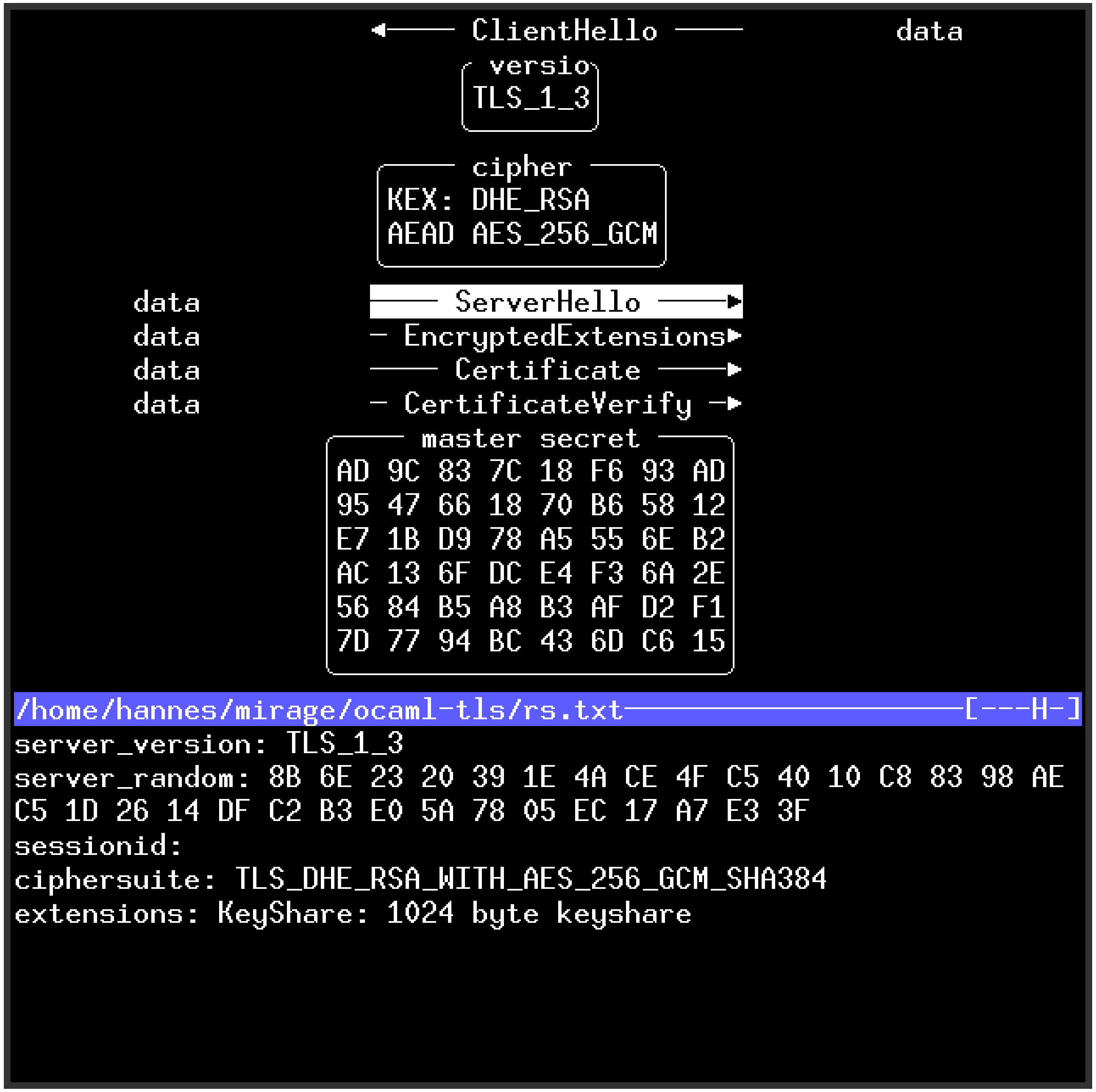
WIP: CONFORMANCE CHECKING

- Read pre-shared keys
- Read ServerConfiguration and secret for 0-RTT
- Evaluate code coverage in nqsb and YourTLS
- Trigger post handshake authentication if YourTLS server
- Negative tests

VISUALISATION

- Input: recorded trace from nqsb
- Renders trace as sequence diagram (terminal/html)
- Purpose: easier to analyse than a trace as text
- WIP: PDF output
- WIP: online server with database

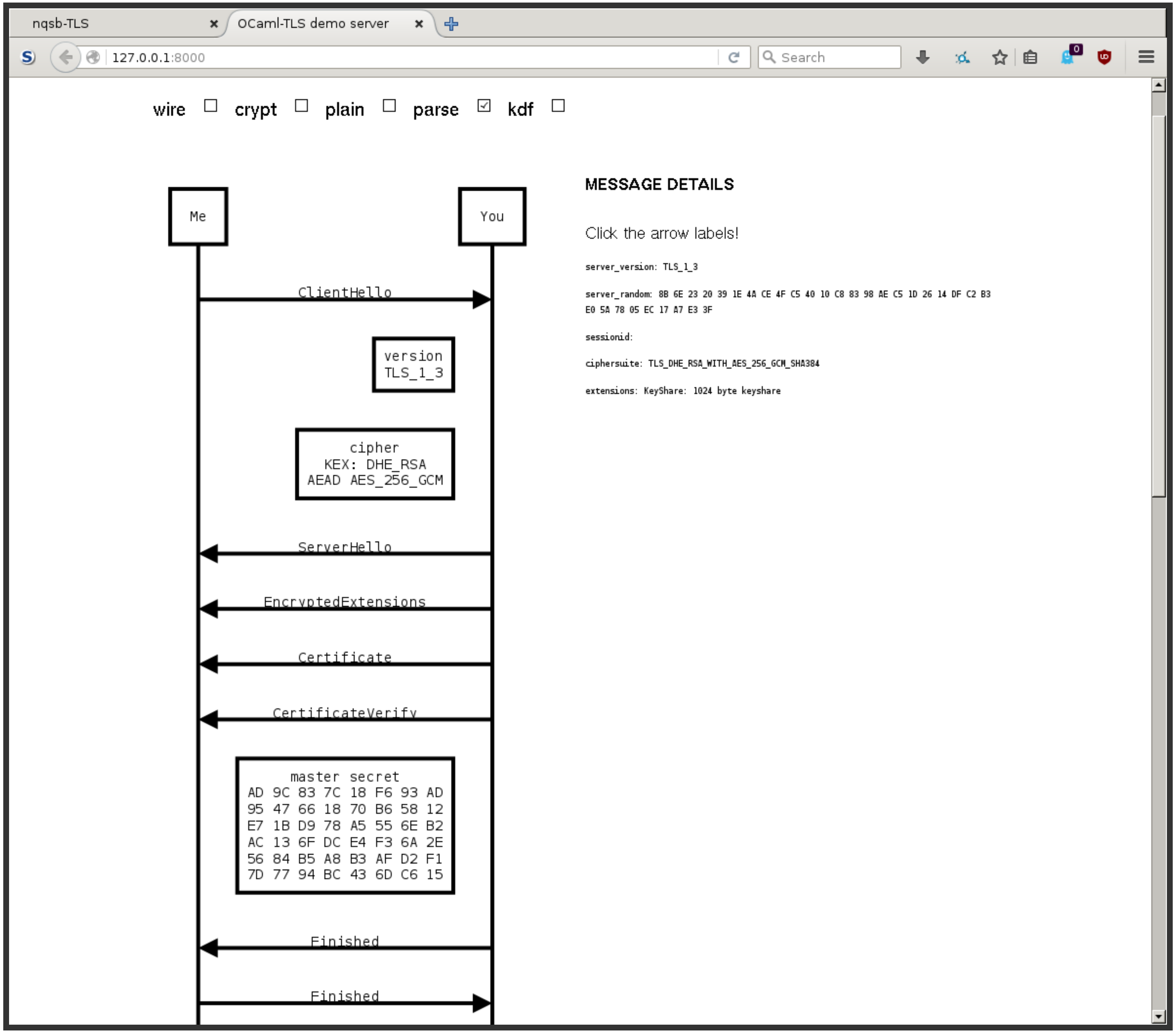
A live demo of vis



```
                    ◄─── ClientHello ───               data
                          ╭ versio╮
                          │TLS_1_3│
                          ╰───────╯

                       ╭─── cipher ───╮
                       │KEX: DHE_RSA  │
                       │AEAD AES_256_GCM│
                       ╰──────────────╯

     data              ─── ServerHello ───►
     data              ─ EncryptedExtensions►
     data              ─── Certificate ───►
     data              ─ CertificateVerify ─►
                     ╭─── master secret ───╮
                     │AD 9C 83 7C 18 F6 93 AD│
                     │95 47 66 18 70 B6 58 12│
                     │E7 1B D9 78 A5 55 6E B2│
                     │AC 13 6F DC E4 F3 6A 2E│
                     │56 84 B5 A8 B3 AF D2 F1│
                     │7D 77 94 BC 43 6D C6 15│
                     ╰───────────────────────╯
/home/hannes/mirage/ocaml-tls/rs.txt────────────────────[---H-]
server_version: TLS_1_3
server_random: 8B 6E 23 20 39 1E 4A CE 4F C5 40 10 C8 83 98 AE
C5 1D 26 14 DF C2 B3 E0 5A 78 05 EC 17 A7 E3 3F
sessionid:
ciphersuite: TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
extensions: KeyShare: 1024 byte keyshare
```



wire ☐ crypt ☐ plain ☐ parse ☑ kdf ☐

```
version
TLS_1_3
```

```
cipher
KEX: DHE_RSA
AEAD AES_256_GCM
```

```
master secret
AD 9C 83 7C 18 F6 93 AD
95 47 66 18 70 B6 58 12
E7 1B D9 78 A5 55 6E B2
AC 13 6F DC E4 F3 6A 2E
56 84 B5 A8 B3 AF D2 F1
7D 77 94 BC 43 6D C6 15
```

MESSAGE DETAILS

Click the arrow labels!

```
server_version: TLS_1_3

server_random: 8B 6E 23 20 39 1E 4A CE 4F C5 40 10 C8 83 98 AE C5 1D 26 14 DF C2 B3
E0 5A 78 05 EC 17 A7 E3 3F

sessionid:

ciphersuite: TLS_DHE_RSA_WITH_AES_256_GCM_SHA384

extensions: KeyShare: 1024 byte keyshare
```

WIP: REPLICATION

- Input: trace, ephemeral and static secret, YourTLS binary
- Replays one side of trace against YourTLS
- Reports discrepancy in behaviour
- Records new trace

WIP: SESSION VALIDATION

- Input: session as TCP stream, ephemeral and static secrets
- Validates session against nqsb-TLS protocol handler
- Looks ahead for decisions (ciphersuite, random, ..)
- Result: would nqsb have also accepted/denied the session?
- (outdated 1.2/1.1/1.0 version available at https://github.com/hannesm/trace-checker)

EARLY DRAFT11 COMMENTS

- Should signing (PSS) hash handshake_log again (#407)?
- NewSessionTicket: one or any number after a single Finished?
- NewSessionTicket: one more after client authentication?
- NewSessionTicket: useful after PSK/(EC)DHE_PSK?
- Fragment buffers must be empty before switching crypto
- Rely on 32bit UNIX epoch time (#348)
- KeyShareEntry encoding (#410)

INSTALLATION

- Requires OCaml >= 4.02.2 and opam >= 1.2.2 (http://ocaml.org)
- `opam remote add tls13 https://github.com/mirleft/tls13-opam.git`
- `opam update`
- `opam install tls` (provides echo_client, echo_server, etc.)
- `opam install tlstoools` (provides tlsweb and tlsvis)

CONCLUSION

- A partial TLS 1.3 implementation/model
- Conformance checking, used as mechanised specification
- Eager to get interoperability working with YourTLS!
- 2-clause BSD licensed
- Contact: tls13@nqsb.io
- Information: https://nqsb.io