TRANSPORT LAYER SECURITY PURELY IN OCAML

Hannes Mehnert and David Kaloper
University of Cambridge

OCaml 2014, Göteborg, 5th Sep 2014

https://github.com/mirleft/ocaml-tls/

CURRENT STATE

- Mirage operating system uses OCaml
- Memory safety, abstraction, modularity

CURRENT STATE

- Mirage operating system uses OCaml
- Memory safety, abstraction, modularity
- But for security call unsafe insecure C code??
- Each line of C code is one line too much!!

MOTIVATION

- Protocol logic encapsulated in declarative functional core
- Side effects isolated in frontends
- Concise, useful, well-designed API

WHAT IS TLS?

- Cryptographically secure channel (TCP) between two nodes
- Most widely used security protocol (since > 15 years)
- Protocol family (SSLv3.0, TLS 1.0, 1.1, 1.2)
- Algorithmic agility: negotiation of key exchange, cipher and hash
- Uses X.509 (ASN.1 encoding) PKI for certificates

PROTOCOL DETAILS

- Security properties:
  - Authentication (optional mutual)
  - Secrecy
  - Integrity
  - Confidentiality
  - Forward secrecy (using ephemeral Diffie Hellman)
- Handshake, Change Cipher Spec, Alert, Application Data, Heartbeat subprotocols

AUTHENTICATION (X.509)

- Client has set of trust anchors (CA certificates)
- Server has certificate signed by a CA
- During handshake client receives server certificate chain
- Client verifies that server certificate is signed by a trust anchor

HANDSHAKE

Showing live!

ATTACKS

- Apple's "goto fail"
- Heartbleed
- "Change cipher suite" message
- Timing attacks (Lucky13, Bleichenbacher, ..)

OCAML-TLS STATS

- Code size: OpenSSL 350kloc, LibreSSL 300kloc, PolarSSL 50kloc, **OCaml-TLS 10kloc**
- Interoperability (server served > 50000 sessions)
- Missing features: client authentication, session resumption, ECC ciphersuites
- Performance: roughly 5 times slower than OpenSSL, but most time spent in C (3DES)

FUTURE

- Prepare another release
- Performance improvements
- Generation of comprehensive test suites
- Implement missing features
- Finish porting to Mirage directly on Xen
- Establish trust into OCaml-TLS: read our code!

CONCLUSION

- Took roughly 3 months to implement (still polishing)
- Modular functional language encapsulates protocol logic (separation of side effects)
- Nocrypto library (`opam install nocrypto`)
- ASN.1, X.509 libraries (`opam install asn1-combinators x509`)
- TLS (`opam install tls`) with mirage and lwt frontends
- Blog series http://openmirage.org/blog/introducing-ocaml-tls