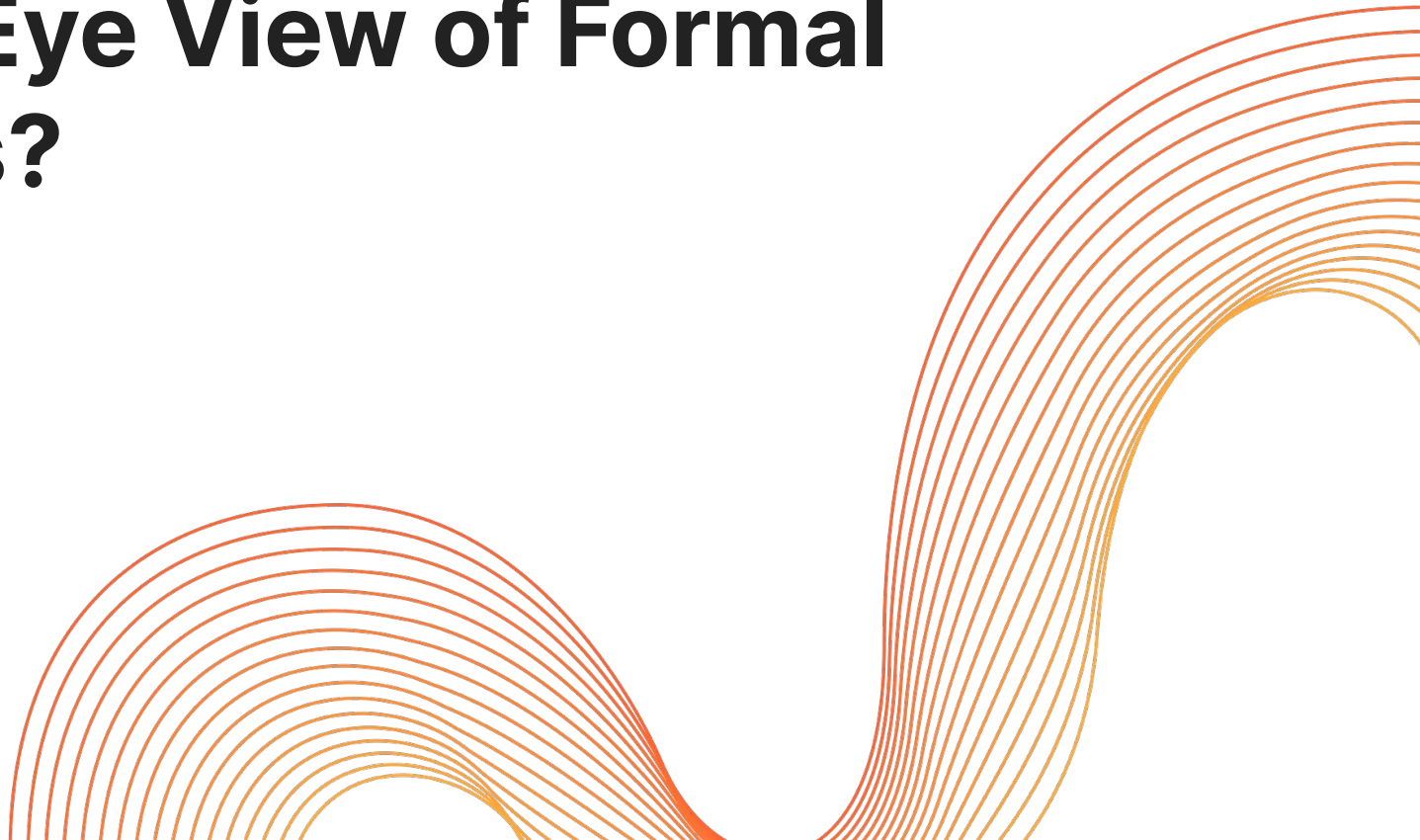
CLOUDFLARE

A Bird's Eye View of Formal Methods?

And Why We Should Care



CLOUDFLARE

Who Am I

- Jonathan Hoyland
- Cryptographer at Cloudflare
- TLS Enthusiast
- Formal Analysis background
  - TLS 1.3, KEMTLS, Exported Authenticators, Delegated Credentials, ODoH, OHTTP, ...

CLOUDFLARE

A Bird’s Eye View of Formal Methods

CLOUDFLARE

Formal Methods

Formal methods are a group of techniques that use maths to *prove* that a system is “correct”.

Often broken up into two broad areas:

Formal Analysis:

- “If my draft / RFC was implemented perfectly, would it have the properties I want?”
- Alternatively: Does the design do what I think it does?

Formal Verification:

- “Does this piece of code implement my draft / RFC correctly?”
- Alternatively: Does the code implement the design

CLOUDFLARE

Formal Analysis

A successful formal analysis proves that a protocol does what we think it does*†‡¶§#. It might proceed as:

1. Write an algebraic description of the protocol
2. Write an algebraic description of the properties we want
3. Use a model checker to prove that the algebraic description of the protocol has the properties described in our algebraic description of the properties

CLOUDFLARE

Formal Analysis

Unfortunately, this is hard.

- The problem is undecidable in the general case
- Describing the protocol correctly is hard
- Deciding the correct properties is hard
- Describing the properties correctly is hard
- Producing the proof that the protocol model has the modelled properties is hard
- Ensuring that implementors understand the protocol the same way the modeller does is hard

CLOUDFLARE

Formal Verification

A successful formal verification proves that an implementation correctly implements a protocol*†‡¶§#. It might proceed as:

1. Implement the protocol in a dedicated language
2. Write an algebraic description of the protocol
3. Use an model checker to prove that the implementation of the protocol is functionally correct
4. Use a (hopefully verified) compiler to produce code

Formal Verification

Unfortunately, this is hard.

- The problem is undecidable in the general case
- Implementing the protocol correctly is hard
- Deciding the correct properties is hard
- Describing the desired protocol properties correctly is hard
- Compiling the implementation such that the proofs hold for the compiled code is hard

CLOUDFLARE

Formal Methods at the IETF

CLOUDFLARE

If it's all so hard, why should I care?

Results

- Formal methods have been applied to security critical IETF drafts for a number of years, and have found bugs that were fixed *before* they became RFCs
- Cremers et al. found a bug that chained multiple handshakes and resumptions in such a way that an attacker could impersonate an authenticated client
- Bhargavan et al. found a bug in PSK bindings that allowed a malicious server to redirect a client's authentication attempt to an honest server.

CLOUDFLARE

If it’s all so hard, why should I care?

Results

- Multiple formally verified implementations of TLS 1.3 exist
- Formally verified implementations of the Signal protocol exist
- Hacspec
- Formally verified code has been included in Chrome and Firefox
- Formally verified implementations can be used to provide test vectors for other implementations
- Formally verified implementations can be used to provide an “oracle” for fuzzers

CLOUDFLARE

Conclusions

CLOUDFLARE

What I think a future Research Group should care about

Encouraging the use of Formal Methods in Standards Development

- Encouraging and enabling formal analysis *during* the standardisation process
- Acting as a repository of knowledge on Formal Methods and a source of experts and educators
- Providing feedback to tooling developers to make tools easier to use
- Peer review on analyses