TLS 1.3 SNI Protection Big Picture

Eric Rescorla

ekr@rtfm.com

SNI Protection in Context

- Basic assumption is that multiple sites share a given IP
- Motivations
  - Anti-censorship
  - Prevent monitoring
- Other vectors
  - DNS queries
  - Traffic analysis of TLS
- Protecting SNI is a down payment on preventing attack
  - Not a complete solution
  - Need TLS padding and probably assume DNS-E is coming

SNI Protection Options (High Level)

- Do nothing
- Abandon SNI
  - Possibly with delegation
- Pre-arranged key
  - Disseminated via DNS, prior handshake, etc.
- Anonymous DHE before SNI
  - What was in draft-rescorla-tls13-flows-01

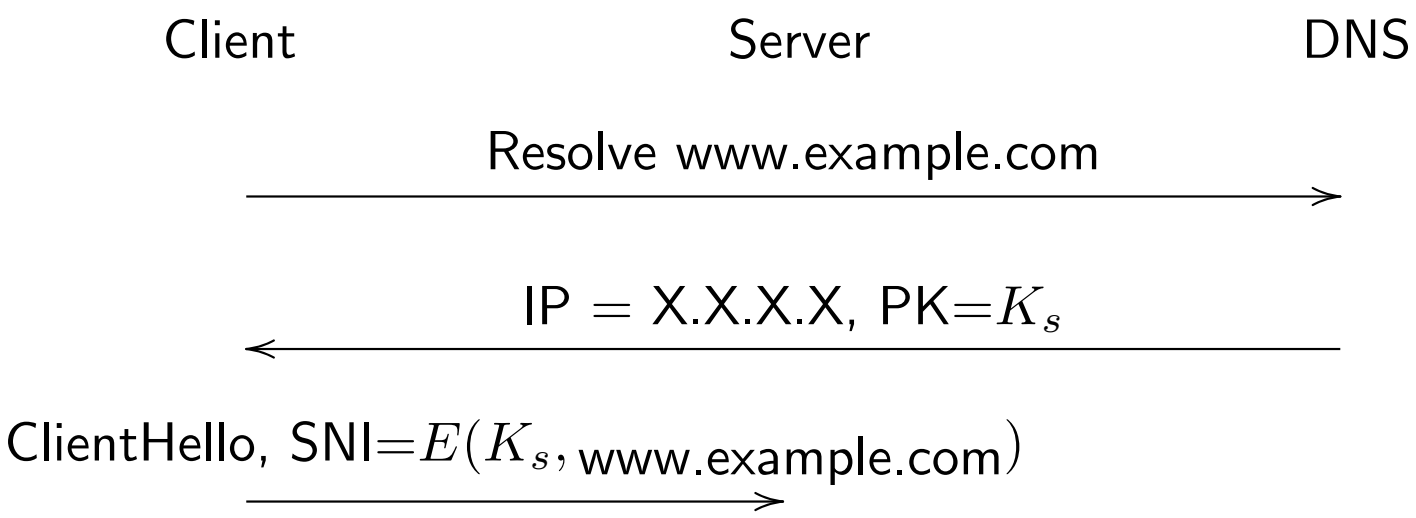
Delegation?

- Server has certificates for domains A and B
  - SNI lets the client indicate which one it wants
- Alternative: let A endorse B
  - So client can always ask for B
  - Use next-level up indicator (e.g., `Host:`) for switching/routing
- Lots of implementation options
  - DNS SRV
  - TLS header
  - HTTP Alt-Svc (this is mostly an HTTP issue)
- This has obvious other advantages

Optimism versus Certainty

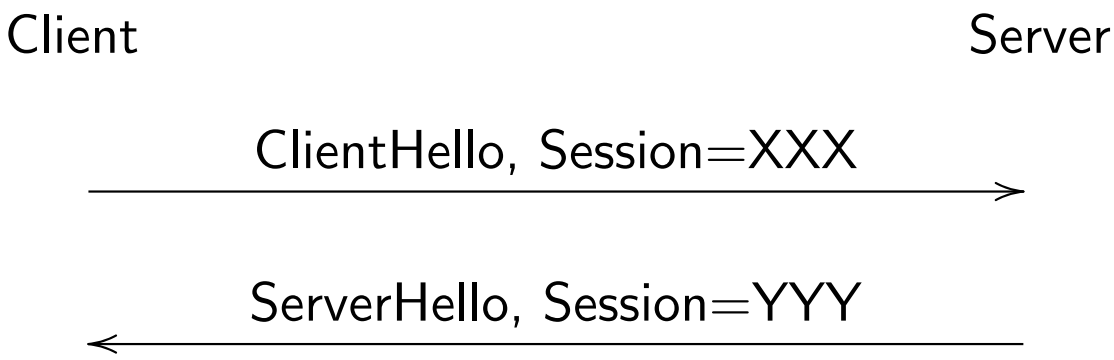
- Traditionally TLS assumes clients know nothing about server
  - We are moving towards modes that assume client state
- Optimistic
  - Client thinks it knows server capabilities
  - But falls back if it's wrong
- Certain
  - Client knows server capabilities
  - Fails if it's wrong

Example: SNI Encryption With External Key



- What happens if the server forgets K_s?
- Connection failure

Example: Session Resumption



- If server loses state, it just corrects the client and it falls back
- Connection succeeds
- Cost is protocol complexity

Can we split the problem?

- External keys and delegation assume client knowledge
- But that knowledge might be implanted in multiple ways
  - DNS (probably DNSSEC)
  - Previous TLS connections
  - HTTP headers
  - Other unspecified mechanisms
- Would be nice to have flexibility here

What do we need from secure DNS?

- DNSSEC provides integrity (ostensibly we have this now)
  - Absolutely needed for delegation
  - Needed for encryption with active attack resistance
  - ... but not against passive attack
- DNS-E (whatever that is) provides confidentiality
  - But caching gives a lot of this anyway

Who bears the cost?

- It's clear this is not a universally wanted feature
- Can we put the cost on those who want it
  - Opt-in from servers
  - Allow clients to ignore

What decision do we need to make?

- Does the TLS state machine need to accomodate this?
- In-band DHE requires significant state machine support
- The rest do not
- Need to decide between those groups
- Can do detailed design of delegation, external keys, etc. later.