

cTLS Overview



draft-rescorla-tls-ctls-02
Richard Barnes
Eric Rescorla

Rationale

- We spent a lot of time on TLS 1.3
- Widely studied, implemented, and deployed
 - 10+ papers
 - 20+ implementations
 - > 20% of Firefox, Chrome, and Safari traffic
- Fully general
 - Already seeing extensions like ESNI, subcerts, etc.
- ... but not compact

Two General Approaches

1. Keep the protocol general but cut as much encoding overhead as possible
 - Remove redundant length fields
 - Variable-length integers instead of fixed-size length fields
 - Implicit values where possible
 - Shorten excessively long cryptovars
 - This is effectively TLS 1.3 with a better encoding
2. Nail down protocol modes and remove negotiation for parameters which are now redundant
 - Signature algorithms, key exchange modes, etc.
 - Explicit or implicit "Shape" parameter to tell you what mode you are in
 - This is effectively a form of compression
 - Probably expand the transcript to stock TLS 1.3 (with cross-protocol defense)

Either approach has a reasonable chance of keeping TLS 1.3 proofs valid

Base: ClientHello

```
uint16 ProtocolVersion;  
opaque Random[32];  
  
uint8 CipherSuite[2]; /* Cryptographic suite selector */  
  
struct {  
    ProtocolVersion legacy_version = 0x0303; /* TLS v1.2 */  
    Random random;  
    opaque legacy_session_id<0..32>;  
    CipherSuite cipher_suites<2..2^16-2>;  
    opaque legacy_compression_methods<1..2^8-1>;  
    Extension extensions<8..2^16-1>;  
} ClientHello;
```

Re-Encoding Example: ClientHello

```
uint8 ProtocolVersion;    // 1 byte
opaque Random[16];       // shortened
uint8 CipherSuite;       // 1 byte

struct {
    ProtocolVersion versions<0..255>;
    Random random;
    CipherSuite cipher_suites<1..V>;           // Varint length
    Extension extensions[remainder_of_message]; // Implicit length
} ClientHello;
```

Compression Example: ClientHello

```
struct {  
    // Versions and ciphers negotiated elsewhere  
    // ... but still included in transcript  
    // ... via the decompressed ClientHello  
    opaque random[16];  
    opaque dh_key<0..255>;  
} ClientHello;
```

Preliminary Results (mutual auth)

Strategy	Re-encoding	Compression
Flight 1	59	48
Flight 2	175 + Cert/ID	152 + Cert/ID
Flight 3	113 + Cert/ID	104 + Cert/ID

What's next?

- These are preliminary results
 - A number of obvious optimization opportunities
- Next steps
 - Demonstrate isomorphism to TLS 1.3
 - So we know the proofs carry over
 - Is it worth doing compression strategy?
 - Most compact
 - But also less general
 - Do we want to expand the transcript?