Exported Authenticators

- Exported Authenticators provide proof that a peer independently controls multiple certificates
- We propose to an extension to extend that proof to prove that a peer jointly controls multiple certificates
- Joint authentication means that all of the EAs were generated by the same actor
• Potentially more than one party can create EAs for a TLS channel

• Potential Scenarios:
  – A server using a static-RSA key-exchange with TLS 1.2 might be being monitored and modified
  – An attacker who has compromised a server’s certificate, can masquerade as that server
  – A CDN coalescing multiple keyless SSL certificates
  – A server shared a resumption master secret with a different server, with potentially different certificates
Layered Exported Authenticators

• We propose an extension that links an Exported Authenticator to a previous one, forming a chain
  
  struct {
    opaque prev_certificate_request_context<0..2^8-1>;
    opaque prev_Finished[Hash.length];
  } LayeredEA;

• If an actor receives a layered EA with an uncompromised certificate then the sender believes all certificates in the chain up to that point are valid.
Use Cases

1. Updating pinned certificates (even after compromise)
   - A server signs the pinned certificate with the new certificate
   - An attacker needs to compromise the old certificate and obtain a mis-issued certificate

2. Proof of acceptance
   - If a peer binds to a certificate the actor, then the actor knows the bound certificate was accepted

3. Authentication on static-RSA TLS 1.2 connections
   - A middlebox cannot make either the client or server accept an inserted certificate into a chain