# Agenda

<table>
<thead>
<tr>
<th>Time</th>
<th>Speaker</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>EKR</td>
<td>Agenda Bash</td>
</tr>
<tr>
<td>15</td>
<td>Brian Minard</td>
<td>draft-dugal-tls-ecmqv-00</td>
</tr>
<tr>
<td>10</td>
<td>Nagendra Modadugu</td>
<td>draft-ietf-tls-ctr-00</td>
</tr>
<tr>
<td>10</td>
<td>Russ Housley</td>
<td>draft-housley-tls-authz-extns-00.txt</td>
</tr>
<tr>
<td>10</td>
<td>Yngve Petterson</td>
<td>Interop issues</td>
</tr>
<tr>
<td>10</td>
<td>Magnus Westerlund</td>
<td>draft-ietf-mmusic-rfc2326bis-12</td>
</tr>
<tr>
<td>60</td>
<td>EKR</td>
<td>draft-ietf-tls-rfc4346bis-00</td>
</tr>
</tbody>
</table>
draft-ietf-tls-rfc4346bis-00

Eric Rescorla
Network Resonance
ekr@networkresonance.com
Background

- RFC 4346 (TLS 1.1) is just waiting for RFC-Ed to push it out
- Recent attacks on MD5 and SHA-1
  - Don't *immediately* threaten TLS, but...
- Rechartered to do a TLS 1.2
  - To do hash function fixes
- Output is draft-ietf-tls-rfc4346
Changes in this draft

- Merged in TLS Extensions and AES Cipher Suites
- Extension for client to indicate which hash functions are supported in certificates
- Replacement of MD5/SHA-1 in the PRF
- Replacement of MD5/SHA-1 in the digitally-signed element.
Digitally-signed

- RSA
  - Sign a concatenated MD5/SHA of handshake messages
- DSA/ECC
  - Sign a SHA-1 hash
- Replaced with hash used to sign the certificate
- ... or SHA-1 for DSA/ECDSA
KDF

- HMAC-based PRF construction
  - XOR SHA-1 and MD5 values
- Retain basic PRF structure
  - based on negotiated hash function in cipher suite
  - What to do about MACs which aren’t hash-based?
- And what about other PRFs? GOST, NIST 800-56, etc.
Finished Message

- Uses the same PRF as for the KDF
  - Current structure: $PRF(H(Handshake\_messages))$
  - This avoids the need to buffer (key is first input to PRF)
    * But it’s less secure
    * Should we move to PRF of the whole handshake

- But... the Finished messages provide downgrade protection
  - Only as strong as weakest common hash function
  - We’re now in the business of approving/disapproving algorithms
    * Hard to get around this
    * Reminder: it’s mostly preimages we care about
Framing the Discussion

• Certificate selection can be done by extension

• The main reason for a TLS 1.2 is to replace the PRF and digitally-signed elements
  – There is no currently known threat to these
  – But it seems ugly to be tied to hashes that don’t meet there design goals

• So should we be making a proactive change like this?