

TLS 1.2 Update

Eric Rescorla

Network Resonance

`ekr@networkresonance.com`

Status

- All open issues now closed
- Summary of major changes on following slides
- Document is in WGLC
- Please read it

Hash Agility

- Digest and signature algorithms now specified in pairs

```
enum {  
    none(0), md5(1), sha1(2), sha256(3), sha384(4),  
    sha512(5), (255)  
} HashAlgorithm;  
  
enum { anonymous(0), rsa(1), dsa(2), ecdsa(3), (255) }  
    SignatureAlgorithm;  
  
struct {  
    HashAlgorithm hash;  
    SignatureAlgorithm signature;  
} SignatureAndHashAlgorithm;  
  
SignatureAndHashAlgorithm  
    supported_signature_algorithms<2..216-1>;
```

- This provides clearer semantics
- Some previous selection rules relaxed

Signature Algorithms: Server Side

- All certs MUST be signed with algorithms in `signature_algorithms`
- EE Cert MUST contain a key that matches the cipher suite
- ServerKeyExchange MUST be signed with an algorithm in `signature_algorithms`.
- Fixed DH certificates may be signed with any permissible algorithm (relaxation of rule from 4346)
- Sensible defaults if `signature_algorithms` not provided

Signature Algorithms: Client Side

- All certs MUST be signed with algorithms in `CertificateRequest.supported_signature_algorithms`
- EE Cert MUST contain a key that matches `CertificateRequest.certificate_types`
CertificateVerify MUST be signed with an algorithm in `CertificateRequest.supported_signature_algorithms`
- Fixed DH certificates may be signed with any permissible algorithm (relaxation of rule from 4346)

Other changes

- Added implementation pitfalls (thanks Pasi)
- `verify_data` is now variable length (cipher suite defined)
- `TLS_RSA_WITH_AES_128_CBC_SHA` is now mandatory to implement
- Removed RC2, DES, and IDEA
- SSLv2 backward compatibility client hello is a MAY