



ERICSSON

# IETF 90 UTA OVERVIEW AND ANALYSIS OF TLS OVERHEAD

JOHN MATTSSON

---

ERICSSON RESEARCH

# MOTIVATION



- Background
  - Ongoing debate how much overhead TLS adds.
  - Quite hard to find information.
- Goals
  - Show exactly how much overhead TLS adds (traffic, latency, processing, memory).
  - Show that TLS for long connections adds very little overhead.
  - Show that for TLS record layer, there is a correlation between high security and low overhead.
  - Give recommendations on how to lower overhead.
  - Give recommendations on how to not lower overhead.

# TLS OVERHEAD ANALYSIS



- TLS overhead can be divided into several aspects
  - Traffic overhead from TLS handshake
  - Latency overhead from TLS handshake
  - Traffic overhead from TLS record layer
  - Processing overhead from TLS handshake
  - Processing overhead from TLS record layer
  
- OCSP Certificate revocation



# TLS HANDSHAKE



- Traffic Overhead

- The TLS handshake typically adds 4-7 kB of traffic overhead. (Details: TLS versions, ciphersuites, extensions, and implementations) **UPDATE**
- TLS compression reduces traffic overhead, but has negative security implications and should be turned off.
- Move from 1024 to 2048 bit RSA keys increases overhead but is needed for security reasons.

- Latency Overhead

- In TLS 1.2, the initial handshake takes 2 round-trips and session resumption takes 1 round-trip
- In TLS 1.3 the target is 1 round-trip for the initial round-trip and 0 round-trips for session resumption.
- Because of the emphasis on reducing latency (instead of only security), TLS 1.3 is expected to have much faster deployment than earlier versions.

- OCSP (Coming in next draft update) **UPDATE**

# TLS CIPHERS IN USE



## Data from (ICSI, July 2014)

- Summarized over record layer cipher.
- AES-CBC, RC4, and HMAC-SHA1 dominates.
- AES-GCM and ChaCha20-Poly1305 are starting to showing significant usage

Cipher	Usage
AES_128_CBC_SHA	29.1 %
RC4_128_SHA	17.4 %
AES_128_GCM	14.7 %
AES_256_CBC_SHA	14.0 %
NULL_SHA	9.8 %
RC4_128_MD5	8.3 %
CHACHA20_POLY1305	1.4 %
3DES_EDE_CBC_SHA	< 1.2 %

# TLS CIPHER TRAFFIC OVERHEAD



IP	TCP	TLS Header	[IV/Nonce]	Enc. Content	MAC	[Padding]
----	-----	------------	------------	--------------	-----	-----------

## AES\_128\_CBC\_SHA, AES\_256\_CBC\_SHA

Per-packet overhead (TLS 1.0)	26-41 bytes (avg. 33.5)
TLS header	5 bytes
HMAC-SHA-1	20 bytes
CBC padding	1-16 bytes
Per-packet overhead (TLS 1.1, 1.2)	42-57 bytes (avg. 49.5)
TLS header	5 bytes
Explicit IV	16 bytes
HMAC-SHA-1	20 bytes
CBC padding	1-16 bytes

## 3DES\_EDE\_CBC\_SHA

Per-packet overhead (TLS 1.0)	26-33 bytes (avg. 29.5)
TLS header	5 bytes
HMAC-SHA-1	20 bytes
CBC padding	1-8 bytes
Per-packet overhead (TLS 1.1, 1.2)	34-41 bytes (avg. 37.5)
TLS header	5 bytes
Explicit IV	8 bytes
HMAC-SHA-1	20 bytes
CBC padding	1-8 bytes

## RC4\_128\_SHA, NULL\_SHA

Per-packet overhead (TLS 1.0, 1.1, 1.2)	25 bytes
TLS header	5 bytes
HMAC-SHA-1	20 bytes

## RC4\_128\_MD5

Per-packet overhead (TLS 1.0, 1.1, 1.2)	21 bytes
TLS header	5 bytes
HMAC-MD5	16 bytes

## AES\_128\_GCM, AES\_256\_GCM

Per-packet overhead (TLS 1.0, 1.1, 1.2)	29 bytes
TLS header	5 bytes
Explicit Nonce	8 bytes
GMAC	16 bytes

## CHACHA20\_POLY1305

Per-packet overhead (TLS 1.0, 1.1, 1.2)	29 bytes
TLS header	5 bytes
Explicit Nonce	8 bytes
Poly1305	16 bytes

# PROCESSING OVERHEAD



- On processors with hardware support for AES and CLMUL (all modern x86 CPUs). AES\_GCM is much faster than RC4\_SHA, AES\_CBC\_SHA, or CHACHA20\_POLY1305.
- Going from AES\_128\_CBC\_SHA to AES\_128\_GCM reduces processing overhead with 57 % on a Core-i7-3770.
- Without hardware support for AES and CLMUL, CHACHA20 with POLY1305 is much faster than AES\_GCM.
- Going from AES\_128\_CBC\_SHA to CHACHA20\_POLY1305 reduces processing overhead with 68 % on Snapdragon S4 Pro.
- No overhead reason to use NULL\_SHA

Cipher	Speed (cycles/byte)
AES_128_GCM	2.42
AES_128_CBC_SHA	5.59
RC4_128_SHA	8.97

(Core-i7-3770, Gueron, 2013)

Chip	AES_128_GCM	CHACHA20_POLY1305
OMAP 4460	24.1 MB/s	75.3 MB/s
Snapdragon S4 Pro	41.5 MB/s	130.9 MB/s
Sandy Bridge Xeon	900.0 MB/s	500.0 MB/s

(Langley, 2014)



Needed: Update with more ciphers, same unit, message length, and memory...

# CONCLUSIONS




- AES-GCM combines security, low traffic overhead and great performance on modern hardware.
  - Use ChaCha20-Poly1305 on platforms without hardware support for AES-GCM.
- Going from TLS 1.1 with AES\_128\_CBC\_SHA to AES\_128\_GCM or CHACHA20\_POLY1305
  - reduces record layer traffic overhead with 41 %.
  - reduces processing overhead with 57–68 %
- There is actually a correlation between high security and low overhead.



# CONCLUSIONS



- For everything but very short connections, TLS is not inducing any major traffic overhead (nor CPU or memory overhead).
- Needed: Statistics on number of connections and traffic per connection for real world TLS usage in different deployments. What about e.g.
  - Webpage with many parallel short connections.
  - Push mail with many serial short connections. 
- Main impact of TLS is increased latency. This can be reduced by using session resumption, cache information closer to end users, or waiting for TLS 1.3.



**ERICSSON**