ChaCha20 and Poly1305 Cipher Suites for TLS

Adam Langley
Wan-Teh Chang
Outline

- ChaCha20 stream cipher
- Poly1305 authenticator
- ChaCha20+Poly1305 AEAD construction
- TLS cipher suites
- Performance
ChaCha20 stream cipher

- Designed by Dan J. Bernstein
- A variant of Salsa20 to improve diffusion
- Used in BLAKE, a SHA-3 finalist

- 256-bit key
- 64-bit nonce
- 64-bit block counter
- Outputs a 64-byte block of key stream and increments block counter in each invocation
- Plaintext is XOR’ed with the key stream
ChaCha20 function

Input
- constants
- key
- counter
- nonce

State (16 words)
- S0
- S1
- S2
- S3
- S4
- S5
- S6
- S7
- S8
- S9
- S10
- S11
- S12
- S13
- S14
- S15

Output
- key stream (64 bytes)
Poly1305 authenticator

- A Wegman-Carter, one-time authenticator
- Designed by Dan J. Bernstein
- 256-bit key
- 128-bit output
Poly1305 calculation

Key \((r, s)\): \(r \Rightarrow \text{integer } R, s \Rightarrow \text{integer } S\)

Input is divided into 16-byte chunks

\[
C_0 \cdot R^n + C_1 \cdot R^{n-1} + C_2 \cdot R^{n-2} + \ldots + C_{n-2} \cdot R^2 + C_{n-1} \cdot R \mod (2^{130-5}) + S \mod 2^{128}
\]

Evaluate this polynomial with Horner’s Rule
AEAD construction

- The AEAD key is a ChaCha20 key
- For each nonce, derive:
  Poly1305 key = ChaCha20(000...0, counter=0)
  Discard the last 32 bytes of output

- A = additional data
- S = ChaCha20(plaintext, counter=1)
- T = Poly1305(A || len(A) || S || len(S))
- Ciphertext = S || T
TLS cipher suites

● Three forward secret, AEAD ciphers:
  ○ TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256
  ○ TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256
  ○ TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256

● No implicit nonce: fixed_iv_length = 0
● No explicit nonce: record_iv_length = 0
  ○ 8-byte ChaCha20 nonce is the TLS sequence number
Performance

Intel Xeon E5-2690@2.9GHz with Hyper-Threading and Turbo Boost disabled

<table>
<thead>
<tr>
<th>Method</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>AES-128-GCM, AES-NI disabled</td>
<td>131 MB/s</td>
</tr>
<tr>
<td>AES-128-GCM, AES-NI enabled</td>
<td>892 MB/s</td>
</tr>
<tr>
<td>ChaCha20+Poly1305</td>
<td>427 MB/s</td>
</tr>
<tr>
<td>ChaCha20+Poly1305, -march=native</td>
<td>560 MB/s</td>
</tr>
</tbody>
</table>
# Performance

**ARM Cortex-A9@1.2GHz**

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>AES-128-GCM</td>
<td>25 MB/s</td>
</tr>
<tr>
<td>ChaCha20+Poly1305</td>
<td>92 MB/s</td>
</tr>
</tbody>
</table>
MAC performance

Intel Xeon E5-2690@2.90GHz with Hyper-Threading and Turbo Boost disabled

● To authenticate 1KB of data

<table>
<thead>
<tr>
<th>Method</th>
<th>Time</th>
<th>Memory usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMAC (128-bit, with AES calls removed)</td>
<td>270 ns</td>
<td>248 bytes</td>
</tr>
<tr>
<td>Poly1305</td>
<td>561 ns</td>
<td></td>
</tr>
</tbody>
</table>
MAC performance

ARM Cortex-A8@1.2GHz, with NEON enabled

- To authenticate 1KB of data

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Time (ns) with Memory (bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMAC (128-bit, with AES calls removed)</td>
<td>5015.1 with 248 bytes of memory</td>
</tr>
<tr>
<td>Poly1305 (code from SUPERCOP)</td>
<td>3567 ns</td>
</tr>
</tbody>
</table>