

Using SigComp in TLS?

Carsten Bormann, 2002-11-20

SigComp — Signaling compression

- Problem: SIP signaling takes too long at cellular bandwidths (e.g., 8K at 1K/s = 8 s)
 - SIP is highly compressible though
- Problem: Compression heavily encumbered
 - inhibits standardizing one compression scheme
- **Universal Decompressor Virtual Machine**
- draft-ietf-rohc-sigcomp-07.txt
 - Approved by IESG for PS (in RFC ed queue)

UDVM code example (LZSS)

- 0f8604a0 c48d00a0 c41e2031 020900a0 ff8e048c bfff0117 508d0f23
06222101 13210123 16e51d04 22e80611 030e2463 14505123
22525116 9fd22300 00bfc086 a08906

(from draft-price-rohc-sigcomp-user-guide-01.txt)

- **67 bytes**
- **13 instructions**

```
set (udvm_memory_size, 8192)
set (state_length, (udvm_memory_size - 64))
at (32) :index pad (2) set (index_lsb, (index + 1))
:length_value pad (2) :old_pointer pad (2)
at (64) :byte_copy_left pad (2) :byte_copy_right pad (2)
:input_bit_order pad (2) :decompressed_pointer pad (2)
align (64)
MULTILOAD (64, 4, circular_buffer, udvm_memory_size, 0, circular_buffer)
:decompress_sigcomp_message :next_character
INPUT-HUFFMAN (index, end_of_message, 2, 9, 0, 255, 16384, 4, 4096, 8191, 1)
COMPARE ($index, 8192, length, end_of_message, literal)
:literal OUTPUT (index_lsb, 1)
COPY-LITERAL (index_lsb, 1, $decompressed_pointer) JUMP (next_character)
:length INPUT-BITS (4, length_value, !) ADD ($length_value, 3)
LOAD (old_pointer, $decompressed_pointer)
COPY-OFFSET ($index, $length_value, $decompressed_pointer)
OUTPUT ($old_pointer, $length_value) JUMP (next_character)
:end_of_message
END-MESSAGE (0, 0, state_length, 64, decompress_sigcomp_message, 6, 0)
:circular_buffer
```

The UDVM approach

- UDVM Code for a **decompressor** is small
 - 50–500 B — smaller than a cert
 - Download it from compressor at start of session
 - No need to standardize compressors any more
- UDVM optimized for decompression
 - May slow down **decompression** by factor of 3
 - Compression is much slower than that anyway

SigComp and TLS

- SigComp runs fine **above** TLS (or TCP or UDP)
 - This is one way it is used in SIP
- Idea: Allow use of SigComp **in** TLS
 - Allows use of compression by existing apps
 - No need to register/standardize zillions of compressors
 - Instant interoperability of new compressors
- SigComp can use TLS for parameter negotiation

SigComp and TLS: to do

- Define details of using SigComp in TLS:
 - Negotiation of SigComp parameters
 - UDVM size, cycles per bit, ...
 - Interaction of SigComp with Record Protocol
 - UDVM startup, block vs. stream mode, ...
- Bar BOF **tbd** (come to the front at the end)