Verified Time Balancing of Security Protocols

A Case Study

Student Name: Donovan Crichton
Student Number: u6881864

The Message Preamble

All ZRTP Messages are prefixed by a preamble, message length and message type.

- Let p represent the message preamble $0x505a$ followed by a 16-bit length in 32-bit words, followed by the message type 2-word block string (e.g. "hello").

The Hello Messages

- Let p represent the message preamble.
- Let v represent the 4-character long string containing the version of the ZRTP protocol.
- Let c represent the client identification string which identifies the vendor and release of the ZRTP software.
- Let $h3$ represent the following:
    - Let $h0$ represent a 256-bit random nonce.
    - Let $h1$ represent a hash of $h0$ with SHA-256.
    - Let $h2$ represent a hash of $h1$ with SHA-256.
    - Let $h3$ represent a hash of $h2$ with SHA-256.
- Let z represent the 96-bit long unique identifier for the ZRTP endpoint. This is a random number generated at installation time to act as a key when looking up shared secrets in a local cache.
- Let fs represent a 4-bit long sequence of flags. The leading bit is 0, the second represents a signature-capable flag, the third a MiTM flag to identify that this message was sent from a PBX. The fourth and final bit represents a passive flag whichis only used on devices that send hello, but will never send commit messages.

- Let hc represent the number (count) of hashing algorithms.
- Let cc represent the number of cipher algorithms.
- Let ac represent the number of auth tag types.
- Let kc reresent the number of key agreement types.
- Let sc represent the number of SAS types.
- Let hs represent the ordered set of hashing algorithms.
- Let cs represent the set of cipher algorithms.
- Let as represent the set of SRTP auth tag types.
- Let ks represent the set of key agreement types.
- Let ss represents the set of short authentication string types.
- Let m represents the machine authentication code that finishes the ZRTP message.

Alice says Hello to Bob.

$alice_0(p, v, c, h3, z, fs, hc, cc, ac, kc, sc, hs, cs, as, ks, ss, m)$

Bob receives Alice's Hello.

$bob_0(p_{a0}, v_{a0}, c_{a0}, h3_{a0}, z_{a0}, fs_{a0}, hc_{a0}, cc_{a0}, ac_{a0}, kc_{a0}, sc_{a0}, hs_{a0}, cs_{a0}, as_{a0}, ks_{a0}, ss_{a0}, m_{a0})$

Bob acknowledges Alice's hello via HelloAck.

$bob_1(p)$

Alice receives Bob's HelloAck

$alice_1(p_{b1})$

Bob says Hello to Alice.

$bob_2(p, v, c, h3, z, fs, hc, cc, ac, kc, sc, hs, cs, as, ks, ss, m)$

Alice receives Bob's Hello.

$alice_2(p_{b2}, v_{b2}, c_{b2}, h3_{b2}, z_{b2}, fs_{b2}, hc_{a0}, cc_{b2}, ac_{b2}, kc_{b2}, sc_{b2}, hs_{b2}, cs_{b2}, as_{b2}, ks_{b2}, ss_{b2}, m_{b2})$

Alice acknowledges Bob's Hello via HelloAck.

$alice_3(p)$

Bob receives Alice's HelloAck.

$bob_3(p_{a3})$

The Commit and DHPart Messages.

test