CictroKDF

This document will describe the Key Derivation Function CictroKDF. Very loosely based on SHA2 this KDF was engineered with efficiency in mind. For backwards compatibility purposes the KDF only excepts phrases who are a multiple of 4 in character length. CictroKDF has a state, a round function, and an expand function. The state of CictroKDF is initialized as:

$$\mathrm{w} = [\text{'h'}, \text{'a'}, \text{'s'}, \text{'h'}]$$

For each byte in the input the round function is called. The round function the calls the Φ function 100 times. Or more elegantly, the round function, r, applied to the i^{th} byte of the initial message is defined as:

$$r(b_i) = \Phi^{100}(b_i)$$

The Φ function is defined as calling two sub-functions, Ω and Π, or mathematically:

$$\Phi(b_i) = \Omega(\Pi(b_i, k), b_i)$$

where k is the iteration of Φ (zero-indexed). The Π function is then defined as setting the (k mod 8)th bit in byte to 1 and then permuting the byte in the following way:

$$\begin{gathered} b_i(0) \to b_i(3) \\ b_i(1) \oplus b_i(0) \to b_i(4) \\ b_i(2) \oplus b_i(5) \to b_i(6) \\ b_i(3) \to b_i(7) \\ b_i(4) \to b_i(1) \\ b_i(5) \to b_i(0) \\ b_i(6) \to b_i(5) \\ b_i(7) \oplus b_i(5) \to b_i(2) \end{gathered}$$

Then Ω is simply an exclusive-or defined as:

$$\Omega(x, y) = x \oplus y \oplus 56$$

Finally, the value of $r(b_i)$ is stored in the $(i \bmod 4)^{th}$ slot in the state array. After every byte of the message is sent through r then the state array is expanded to 19 bytes through the expansion function, Γ. Each new element to be added to the state array is computed by:

$$\Gamma(w, l) = \Pi(w(0) \oplus w(1) \oplus w(l-1) \oplus w(l), 101)$$

where l is the index of the last element in the state array. As a final step the first 3 bytes of the KDF output is dropped.