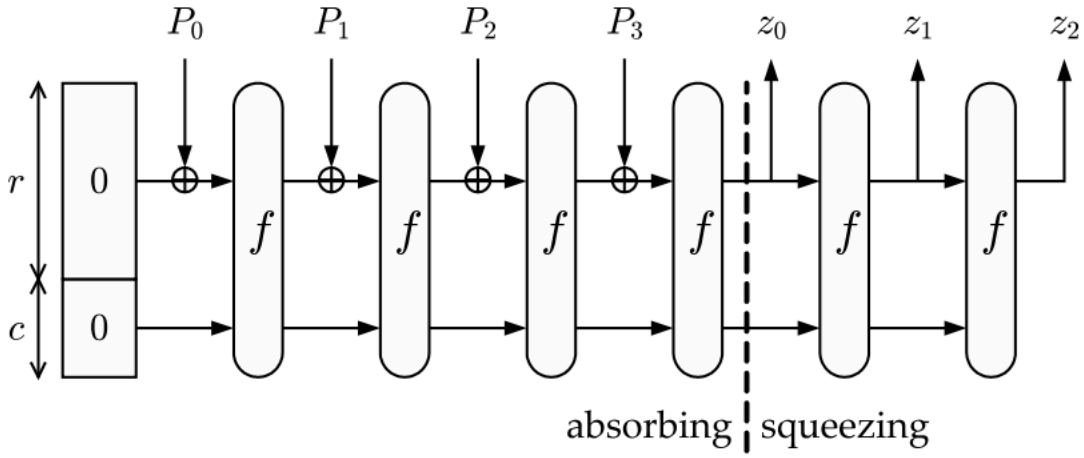
CictroHash

This document will describe the Hash Function CictroHash. It follows modern hashing techniques in a truly elegant and hardware-capable design. The sponge construction is a popular method used by hash functions to transfer through entropy to variable input. It is used in the ever-popular Keccak Hash function and is demonstrated below



Note that for CictroHash the digest will simply be the value z_0. Also, for CictroHash r is 4 bytes in length and c is four bytes in length. The starting state of r and c (denoted S) is shown below.

$$S = \{31, 56, 156, 167, 38, 240, 174, 248\}$$

The pre-image for CictroHash is also padded to the nearest 4-byte increment by all zeros. The state transformation function f is a masterpiece in modern cryptography. It utilizes all of people's favorite operations like left shift, right shift, left rotate, right rotate, and swapping! The first step of f is to take the state array S and put it into the matrix form like below (each element is a byte)

$$w = \begin{bmatrix} S_0 & S_1 & S_2 & S_3 \\ S_4 & S_5 & S_6 & S_7 \end{bmatrix}$$

Then the round function (∇) will be applied 50 times on w, or equivalently

$$f = \nabla^{50}(w)$$

Finally, we can define ∇ as below

$$\nabla(w) = \delta \circ \gamma \circ \beta \circ \alpha(w)$$

Clearly as ∇ is merely a composition of our trivial transformation we just need to define them, and we are done.

$$\alpha(w) = swap(w(0), w(1))$$

$$\beta(w) = \begin{cases} w(0,0) \oplus = w(1,3) \\ w(0,1) \oplus = w(1,2) \\ w(0,2) \oplus = w(1,1) \\ w(0,3) \oplus = w(1,0) \end{cases}$$

$$\gamma(w) = \begin{cases} w(0,0) \to w(0,3) \\ w(0,1) \to w(1,2) \\ w(0,2) \to w(1,3) \\ w(0,3) \to w(1,1) \\ w(1,0) \to w(0,1) \\ w(1,1) \to w(1,0) \\ w(1,2) \to w(0,2) \\ w(1,3) \to w(0,0) \end{cases}$$

$$\delta(w) = \begin{cases} rotate - left: \\ \quad w(0,0) \\ \quad w(1,0) \\ \quad w(0,2) \\ \quad w(1,2) \\ rotate - right: \\ \quad w(0,1) \\ \quad w(1,1) \\ \quad w(0,3) \\ \quad w(1,3) \end{cases}$$

For the δ component the rotations will be 1-bit. After z_0 is calculated the last 4 bytes are dropped off to produce a 4-byte digest.

Example Inputs/Outputs

CictroHash(HELLOWORLD) = 0xb5a79bee
CictroHash(GOODBYEWORLD) = 0xbc08f3d9
CictroHash(kUgKZMdQkn) = 0x38da3d00