POSEIDON Hashes for SNARKs over MTN4 and MTN6 based pairings

We instantiate the x^{-1}-POSEIDON hash function over the scalar fields of both curves $\mathrm{MNT4\text{-}753}$ and $\mathrm{MNT6\text{-}753}$ which we denote by F_6 and F_4, respectively.

By now, both instantiations work with the same number of field elements $t = 3$, capacity $c = 1$ and absorption rate $r = 2$ field elements.

According to the recommendations by Grassi, et al., we choose

$$\begin{aligned} R_F &= 8 \qquad (\text{including 2 rounds security margin}), \\ R_p &= 57 \qquad (\text{including } 7.5\% \text{ security margin}), \end{aligned}$$

to achieve a security level of 128 Bit, see the discussion below.

POSEIDON primitives PH_{F_4/F_6}

We describe the primitive PH_F, where F is either F_4 or F_6, as mapping from

$$PH_F : F \times F \longrightarrow F,$$

hashing a vector of two field elements (x_0, x_1) to a single field element. Domain extension is discussed separately.

PH_F is as follows:

- the two input elements x_0, x_1 are loaded additively to the (initialized) internal state $\vec{s} = (s_0, s_1, s_2)$, i.e.

$$(s_0, s_1, s_2) \leftarrow (s_0, s_1, s_2) + (x_0, x_1, 0),$$

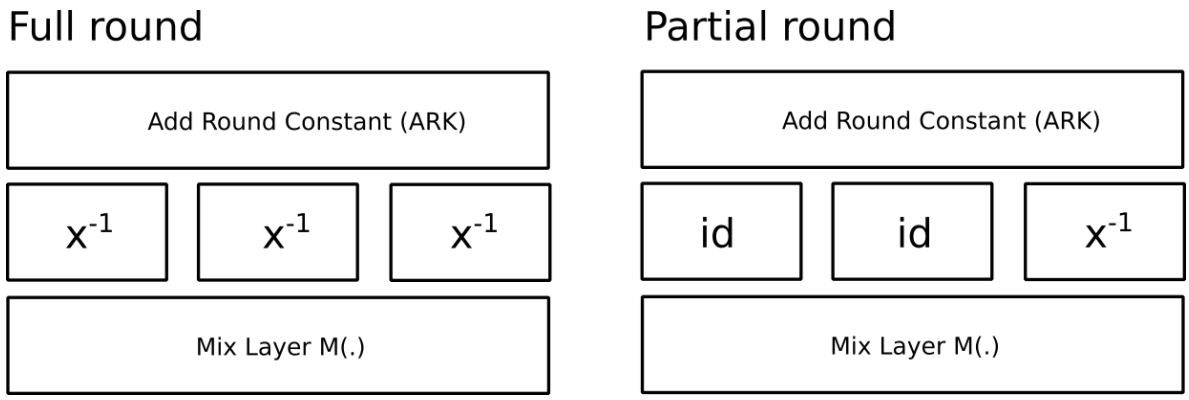
- then the $POSEIDON_\pi$ random permutation is applied to (s_0, s_1, s_2), consisting of

$$\begin{aligned} R_f &= R_F/2 = 4 \qquad \text{full rounds} \\ R_p &= 57 \qquad \text{partial rounds, and another} \\ R_f &= R_F/2 = 4 \qquad \text{full rounds}, \end{aligned}$$

see the pics below.

- The output $PH_F(x_0, x_1)$ is the most "outer" element s'_0 of the internal state (s'_0, s'_1, s'_2).

Domain extension for hashing further pairs of input (x_0, x_1) is done exactly in the same way as the first pair, but by taking the inner state of the previous round as initial state for the next one.



The round dependent vector of round constants

$$\vec{c}_0, \vec{c}_1, \ldots, \vec{c}_{R_F+R_p-1} \in F^3$$

are derived via a linear recursion, an 80 bit Grain LFSR, which is initialized by encoding (or hashing) our context (i.e., the base field $F_{4/6}$, r,c, R_F , R_p, and x^{-1}-S-Box), and the matrix

$$M = \begin{pmatrix} m_{1,1} & m_{1,2} & m_{1,3} \\ m_{2,1} & m_{2,2} & m_{2,3} \\ m_{3,1} & m_{3,2} & m_{3,3} \end{pmatrix}$$

for the Mix Layer is in our instantiations a fixed (3×3)-Cauchy matrix over F.

For efficiency reasons, our M possesses the additional property that all entries have "small" Montgomery representation, see Marcelo's notes or the Mathematica file for details.

The POSEIDON $VerifyPH_{F_{4/6}}$ gadged

$\vec{s}' = VerifyPH_F(\vec{s}, x_1, x_2)$ takes as input

- public $\vec{s} = (s_0, s_1, s_2)$ from F^3 for the initial internal state,
- public variables x_1, $x_2 \in F$ as input to be hashed,
- public variables $\vec{s}' = (s'_0, s'_1, s'_2)$ from F^3 for the internal state after applying the $POSEIDON_\pi$, where $s'_0 \in F$ is the output of the hash,
- private witnesses $(w_{k,1}, w_{k,2}, w_{k,3}) \in F^3$, where $k = 0, 2, \ldots, R_F + R_p$ for the internal wires of each of the rounds.
- enforces

$$\begin{aligned} \vec{w}_0 &= \vec{s} + (x_0, x_1, 0) + c_0, \\ \vec{w}_k &= PHRf(k, \vec{w}_{k-1}) \qquad k = 1, \ldots, R_f, \\ \vec{w}_k &= PHRp(k, \vec{w}_{k-1}) \qquad k = R_f + 1, \ldots, R_f + R_p, \\ \vec{w}_k &= PHRf(k, \vec{w}_{k-1}) \qquad k = R_f + R_p + 1, \ldots, 2 \cdot R_f + R_p - 1, \end{aligned}$$

  and finally

$$\vec{s}' = PHRf(2 \cdot R_f + R_p, \vec{w}_{2 \cdot R_f + R_p}),$$

  where the vector of round constants in the latter is set to $(0, 0, 0)$.

The circuit is as described by Grassi, et al., where S-Box, Mix-Layer, and Add-Round-Constant are merged into a (shifted) round.

Component 1: The modular inversion SBox

$y = SBox(x)$ takes as input a single field element x and enforces y according to

$$SBox(x) = \begin{cases} x^{-1} & x \neq 0, \\ 0 & x = 0, \end{cases}$$

- public variables x and y from F,
- private witness b from F.
- enforces

$$\begin{aligned} 0 &= b \cdot (1 - b), \\ b &= x \cdot y, \\ 0 &= (1 - b) \cdot (x - y). \end{aligned}$$

In these constraints, b is used as a boolean switch. The case $b = 1$ corresponds to the $x \neq 0$ case, in which the second equation enforces $y = x^{-1}$, and the case $b = 0$ enforces by help of both the second and third equation, that $x = y = 0$.

Component 2: The full round $PHRf(x_1, x_2, x_3)$

$(y_1, y_2, y_3) = PHRf(k, x_1, x_2, x_3)$ with index k as round number takes three field elements as input/output and enforces that (y_1, y_2, y_3) corresponds to (x_1, x_2, x_3) applied to the S-Box Layer, Mix Layer, and the Add-Round-Constant Layer:

$$(y_1, y_2, y_3) = ARK \circ M \circ (SBox(x_1), SBox(x_2), SBox(x_3)).$$

- public index to address the round constants $c_{k,1}, c_{k,2}, c_{k,3}$.
- public variable x_1, x_2, x_3, and y_1, y_2, y_3 from F,
- enforces

$$y_i = c_{k,i} + \sum_{j=1}^{3} m_{i,j} \cdot SBox(x_j), \quad i = 1, 2, 3$$

Component 3: The partial round $PHRp(x_1, x_2, x_3)$

$PHRp(k, x_1, x_2, x_3)$ with integer k as round number takes three field elements as input and enforces that its output (y_1, y_2, y_3) corresponds to (x_1, x_2, x_3) applied to the S-Box Layer, the Mix Layer, the Add-Round-Constant Layer and :

$$(y_1, y_2, y_3) = ARK \circ M(x_1, x_2, SBox(x_3)).$$

- public index k to address the round constants $c_{k,1}, c_{k,2}, c_{k,3}$.
- public variable x_1, x_2, x_3, and y_1, y_2, y_3 from F,
- enforces

$$y_i = c_{k,i} + \sum_{j=1}^{2} m_{i,j} \cdot x_j + m_{i,3} \cdot SBox(x_3), \qquad i = 1, 2, 3$$

Security

The role of full rounds

Basically the number $R_F = 2 \cdot R_f$ of full rounds are to prevent statistical attacks for a given security level of M bits:

- classical differential (Biham, Shamer 1991, 1993), linear (Matsui, 1993) and truncated differential (Knudsen, 1994) cryptanalysis,
- rebound attacks (Lamberger, et al. 2009, Mendel, et al., 2009),

- Multiple-of-n and mixed differential cryptanalysis (Grassi, et al., 2017),
- Invariant subspace attack (Leander, et al., 2011),
- Integral/Square attack (Damen, et al., 1997).

We follow the recommendations of Grassi, et al. and choose R_f as three rounds plus one round extra as security margin, i.e. $R_F = 8$.

... and partial Rounds

Once the number of full rounds R_F is settled, the number of partial rounds R_p$ are bounded from below by the maximal solutions of

$$R_F \cdot \log_2(t) + R_P \leq \log_2(t) + \frac{1}{2} \cdot \min(M, n)$$

against interpolation attacks, and in addition

$$(t-1) \cdot R_F + R_P \leq \frac{1}{4} \cdot \min(M, n) - 1$$

against Groebner basis attacks (here, n is the bit length of the modulus of F).

For both instantiations (over F_6 and F_4) if $R_F = 8$, the above equation demand

$$R_p \geq 52.91$$

A choice of $R_p = 57$ therefore yields a security margin of only 7.7%, slightly above the 7.5% recommended by Grassi, et al..