New Bears at the Bear Market: Introducing Polar Bear and Teddy Bear Prime Fields

Tomer Solberg
tomer@ingonyama.com

1 Introduction

Recent advancements in cryptography, particularly in the realm of zero-knowledge (ZK) proofs, have driven significant interest in small prime fields ($\leq$ 64-bit primes) due to their favorable properties for efficient arithmetic and implementation. Notable examples include the widely used Mersenne-31 (M31) [HLN23], Goldilocks [Por22], Baby Bear [BG23], and the recently introduced Koala Bear [SN24, Wan24] primes. These primes, expressed as:

$$p_{M31} = 2^{31} - 1 \tag{1.1}$$

$$p_G = 2^{64} - 2^{32} + 1 \tag{1.2}$$

$$p_{BB} = 2^{31} - 2^{27} + 1 \tag{1.3}$$

$$p_{KB} = 2^{31} - 2^{24} + 1 \tag{1.4}$$

all belong to the family of Solinas primes. These primes are particularly attractive for their balance of performance and structure, offering key advantages:

1. High 2-adicity. The 2-adicity of a finite field $\mathbb{F}_q$ is the highest power of 2 dividing $q-1$, the order of its multiplicative group. Among the aforementioned primes, Goldilocks exhibits the highest 2-adicity at 2^{32}, while Koala Bear has the lowest at 2^{24}. An exception is M31, which achieves a practical 2-adicity of 2^{31}, only when extended to $\mathbb{F}_{p_{M31}^2}$ and using the circle subgroup of order $p_{M31} + 1$.
2. Modern computing architectures favor primes that fit within common word sizes (16, 32, or 64 bits) for efficient arithmetic. Compared to larger primes used in elliptic curve cryptography or RSA, which span hundreds of bits, small primes significantly reduce the computational cost. Specifically, Baby Bear and Koala Bear are treated as 32-bit fields, as well as the circle group of M31, while Goldilocks is a 64-bit field.
3. Modular reduction is a cornerstone of field arithmetic, and the efficiency of this operation varies significantly among these primes. M31 stands out for its unparalleled efficiency, requiring only a single addition for modular reduction. Goldilocks follows, benefiting from optimizations using 24-bit limbs, while Koala Bear and Baby Bear achieve varying degrees of efficiency depending on implementation strategies.

While these primes have been extensively studied and adopted, there remains room for innovation in selecting small primes that balance these properties. In this paper, we propose two new primes that have thus far received little attention within the zero-knowledge community but exhibit strong potential to rival or outperform existing options in terms of 2-adicity, smallness, and modular arithmetic efficiency.

2 Polar Bear

We call $p_{PB} = 2^{40} - 2^{32} + 1$ the Polar Bear prime. It enjoys high 2-adicity of 2^{32}, while taking only 40 bits (or possibly 48, with the implementation we propose).

As for efficiency of arithmetic operations, we propose the following procedure for modular reduction:

Given a number r which we assume is obtained from multiplying two 40 bit numbers (so it is at most 80 bits), we split it into five 16-bit limbs

$$r = a + 2^{16}b + 2^{32}c + 2^{48}d + 2^{64}e \tag{2.1}$$

Note that

$$\begin{align} 2^{48} &\equiv 2^8(2^{32} - 1) \equiv 2^{32} - 2^8 - 1 && \pmod{p_{PB}} \tag{2.2}\\ 2^{64} &\equiv 2^{32} - 2^{24} - 2^{16} - 2^8 - 1 && \pmod{p_{PB}} \tag{2.3}\end{align}$$

So we can write

$$r = a + 2^8(-d - e) + 2^{16}(b - e) + 2^{24}(-e) + 2^{32}(c + d + e) \tag{2.4}$$

This can be compacted by splitting c, d, e into sublimbs of 8 bits such that $x = 2^8x_h + x_l$ for $x = c, d, e$ and define an 8-bit rotation $x_r = 2^8x_l + x_h$. We then write

$$\begin{align} r &= (a - c_h - d_r - e_r) + 2^{16}(b - e - d_h - e_r) + 2^{32}(c_l + d_l + e_l - e_h) \tag{2.5}\\ &= r_0 + r_1 2^{16} + r_2 2^{32} \tag{2.6}\end{align}$$

Of course, addition and subtraction must be done while being mindful of carries, so that the limbs r_0, r_1 are in the range $[2^{16}]$, and the limb r_2 is in the range $[2^8]$. A conditional subtraction must be executed at the end to ensure the result is indeed in the range $[p_{PB}]$.

Lastly, note that for a secure field extension, it suffices to use a cubic extension, which gives 120 bits. Although slightly less than the 124 bits of the quartic extensions of M31, Baby Bear and Koala Bear, it should still provide enough security for most requirements. We propose working with $\mathbb{F}_{p_{PB}}[x]/(x^3 - 3)$ as an efficient and secure field extension.

3 Teddy Bear

We call $p_{TB} = 2^{32} - 2^{30} + 1$ the Teddy Bear prime. It can also be written as $p_{TB} = 2^{31} + 2^{30} + 1 = 3 \cdot 2^{30} + 1$. This prime has been mentioned in passing by Haböck et al [HLN23], but we weren't able to find other mentions or uses. It is easy to see that this prime is similar to Baby Bear, except with higher 2-adicity.

For efficient field operations, we propose a novel representation. Represent a number $x \in \mathbb{F}_{p_{TB}}$ as $x = 3x_1 + x_0$, where $x_0 \in \{0, 1, 2\}$ and $x_1 \in [2^{30}]$. Note that this can be used to represent any number in $\mathbb{F}_{p_{TB}}$ except -1, which is given the unique representation of

$x_1 = 2^{30} - 1,\ x_0 = 3$ ("all ones"). Modular multiplication in this representation can be done as follows. Given 2 numbers $x, y \in \mathbb{F}_{p_{TB}} \setminus \{-1\}^1$, we can write

$$r = xy = 3 \cdot (3x_1y_1) + 3 \cdot (x_0y_1 + x_1y_0) + x_0y_0 \tag{3.1}$$

Note that since $x_0, y_0 \in \{0, 1, 2\}$ calculating x_0y_1, x_1y_0, x_0y_0 does not require multiplication or even addition, only a conditional bit-shift. A single multiplication is required to calculate x_1y_1. This is done as a full integer multiplication, and the 60-bit result is then broken into 30-bit limbs $x_1y_1 = z_0 + 2^{30}z_1$. We then have $3x_1y_1 = 3z_0 - z_1$. The final result can then be written as $r = 3r_1 + r_0$ where

$$r_0 = x_0y_0 \mod 3 \tag{3.2}$$

$$r_1 = 3z_0 - z_1 + x_0y_1 + x_1y_0 + \lfloor x_0y_0/3 \rfloor \tag{3.3}$$

and lastly a conditional subtraction is required to ensure $r_1 \in [2^{30}]$. Note that this approach requires quite a bit of logic together with the arithmetic, so it might not be optimal on GPUs. However, for CPUs or ASICs this approach can be very efficient.

Acknowledgements

The author would like to thank Karthik Inbasekar, Yuval Domb, Daniel Lubarov, Suyash Bagad and Omer Shlomovits for discussions, suggestions, and reviewing the article.

References

[BG23] Jeremy Bruestle and Paul Gafni. Risc zero zkvm: scalable, transparent arguments of riscv integrity, 2023. `https://dev.risczero.com/proof-system-in-detail.pdf`.

[HLN23] Ulrich Haböck, Daniel Lubarov, and Jacqueline Nabaglo. Reed-solomon codes over the circle group. Cryptology ePrint Archive, Paper 2023/824, 2023. `https://eprint.iacr.org/2023/824`.

[Por22] Thomas Pornin. EcGFp5: a specialized elliptic curve. Cryptology ePrint Archive, Paper 2022/274, 2022. `https://eprint.iacr.org/2022/274`.

[SN24] SyxtonPrime and Jacqueline Nabaglo. Koalabear, a possible alternative to babybear, 2024. `https://github.com/Plonky3/Plonky3/commit/b9d6ea993e79782db9a6d8eaf088578bd98fc01d`.

[Wan24] Kayson Wang. Efficient prime fields for zero-knowledge proof, 2024. `https://hackmd.io/@Voidkai/BkNX3xUZA`.

[1] Of course, multiplication of a number by -1 can be done by simply subtracting it from p_{TB}