Defeating Linearization Attacks with Min-Plus-Modulo Digital Signature

Abstract

A novel digital signature method called Min-Plus-Modulo (MPM) uses tropical algebra and modular reduction to make cryptographic security better against linearisation attacks. The Min-Plus-Modulo method uses polynomial operations, such as coefficientwise minima and modular additions, to create and check signatures for messages that are polynomials. We look at how well the MPM technique works for different polynomial degrees in terms of speed and signature size. Our results show that the polynomial degree quadratically scales signing and verification activities, although the sizes of the signatures stay small. When security parameters d=100, 150, and 200 are used, the technique works well on standard hardware, with verification times of less than 5 seconds and signature sizes of about 12-25 kB. Modular reduction makes systems more resistant to attacks like Brown-Monico linearisation without needing more processing or storage space than standard tropical signature systems. The MPM scheme is a promising post-quantum solution that combines small signatures, security, and speed.

Keywords: Tropical cryptography, Min-plus modulo algebra, Polynomial semirings, Digital signatures, NP-hard problems, Hash-to-polynomial conversion

1 Introduction

The recent development of quantum computing poses a significant threat to modern cryptography systems, necessitating the cryptographic community to investigate mathematical principles beyond conventional number-theoretic assumptions. In this endeavor, different algebraic structures have arisen as viable frameworks for developing post-quantum secure cryptographic primitives. The min-plus semiring, referred to as tropical algebra in cryptography, has attracted considerable attention due to its intrinsic non-linearity and computational complexity characteristics.

The foundation of tropical cryptography was systematically established through a series of seminal contributions. Grigoriev and Shpilrain [1] pioneered this domain by introducing tropical algebra as a viable platform for public-key cryptography, demonstrating its potential through key exchange protocols. Their subsequent work extended this framework via homomorphic constructions [2], while parallel research explored related algebraic structures such as semidirect products of semigroups [3, 4]. However, the cryptanalysis by Kotov and Ushakov [5] revealed vulnerabilities in early tropical matrix-based protocols, highlighting the need for more robust constructions.

A critical theoretical breakthrough came from Kim and Roush [6], who established the NP-hardness of factoring one-variable tropical polynomials. This computational hardness result provided a rigorous foundation for advanced cryptographic applications. Building upon this, Chen, Grigoriev, and Shpilrain [7] recently proposed the first practical digital signature scheme based on tropical polynomial factoring, marking a significant milestone in the field. Their work was further refined in a journal publication [8], presenting a comprehensive framework for tropical digital signatures.

The security landscape for tropical cryptography has evolved rapidly through an ongoing cycle of proposal and analysis. Panny's forging attacks [9] on initial constructions prompted immediate responses, including security enhancements and alternative approaches [10, 11]. This dynamic development mirrors the historical evolution of classical paradigms like the Fiat-Shamir transformation [12], underscoring the maturation process of novel cryptographic primitives.

Recent comprehensive reviews [13] have synthesized the growing body of tropical cryptographic research, while advanced complexity analyses [14] have deepened our understanding of tropical polynomial equation systems. The broader mathematical context, drawing from tropical geometry [15] and max-linear systems theory [16], continues to provide rich structural insights for cryptographic applications.

1.1 Problem Statement and Research Questions

Tropical algebra continues to attract attention as a promising non-linear foundation for post-quantum cryptography. However, several existing constructions have shown susceptibility to powerful algebraic attacks—including linearization, factorization, and structural reduction techniques [5, 9]. These attacks typically exploit the inherent distributive and monotonic properties of classical tropical semirings, revealing a critical gap in the design of secure and efficient tropical cryptographic primitives.

The central problem addressed in this work is the absence of a robust, attack-resistant digital signature scheme within the tropical paradigm that can withstand

such advanced cryptanalytic techniques without compromising practical efficiency. Despite the theoretical advantages of tropical operations, current schemes often fail to provide meaningful resistance against adversaries capable of exploiting algebraic vulnerabilities.

Motivated by this gap, our investigation is guided by the following research questions:

1. **Hybrid Algebraic Design:** Can a hybrid algebraic structure—combining tropical (min-plus) operations with modular arithmetic—be constructed in a way that inherently disrupts or neutralizes linearization-based attacks?
2. **Performance and Practicality:** What are the computational overheads, signature sizes, and overall efficiency characteristics of a digital signature scheme built on this hybrid algebra? More importantly, can such a scheme operate effectively in resource-constrained settings?
3. **Security Comparison:** How does the security of the proposed hybrid scheme compare to existing classical tropical signature schemes and modern lattice-based post-quantum signatures, particularly in terms of qualitative resistance to structural algebraic attacks?

In this paper, we present a novel digital signature scheme that addresses key vulnerabilities in existing tropical cryptographic constructions while advancing the theoretical framework. Our approach introduces a Min-Plus-Modulo (MPM) hybrid algebra that synergistically combines tropical operations with modular arithmetic, preserving the NP-hard foundation while incorporating additional cryptographic safeguards through finite field structure.

This is how the rest of the paper is organized. Section 2 (Preliminaries) provides the necessary background on min-plus modulo algebra and its cryptographic constructions. Section 3 presents the proposed Digital Signature, including its correctness, parameter selection, underlying hard problem, possible attacks, performance, and signature size, and provides a comparison between the Tropical Alternative and the Min–Plus–Modulo (MPM) scheme. Section 5 concludes by summarizing our results and suggesting possible avenues for further study.

2 Preliminaries

In this section, we introduce the algebraic structure used in our digital signature scheme, known as the *Min-Plus-Modulo Algebra*. This structure blends elements of tropical (min-plus) algebra with modular arithmetic to enhance both the theoretical strength and practical security of the protocol.

2.1 The Min-Plus-Modulo Algebra

Let $p \geq 2$ be a fixed modulus. Consider the set

$$S = \mathbb{Z}_p = \{0, 1, 2, \ldots, p-1\}.$$

We define two binary operations on S: tropical addition and modular multiplication (in the tropical sense).

$$x \oplus y = \min(x, y)$$
$$x \otimes y = (x + y) \bmod p$$

The resulting algebraic system is denoted by

$$\mathcal{M}_p = (S, \oplus, \otimes),$$

and referred to as *Min–Plus–Modulo algebra*.

These operations meet the following algebraic rules:

- **Associativity:**

$$x \oplus (y \oplus z) = (x \oplus y) \oplus z,$$
$$x \otimes (y \otimes z) = ((x+y) \bmod p + z) \bmod p = (x+y+z) \bmod p = (x+(y+z) \bmod p) \bmod p = ((x \otimes y) \otimes z)$$

- **Commutativity:**

$$x \oplus y = y \oplus x,$$
$$x \otimes y = (x + y) \bmod p = (y + x) \bmod p = y \otimes x$$

- **Idempotent:**

$$x \oplus x = x$$

Furthermore, additive identity is defined as $\varepsilon = p - 1$, such that $\varepsilon \oplus x = x$, and multiplicative identity is 0, since $0 \otimes x = 0 + x \mod p = x$.

The key question is whether $\otimes$ is distributed over $\oplus$:

$$x \otimes (y \oplus z) \stackrel{?}{=} (x \otimes y) \oplus (x \otimes z).$$

Left-hand side:

$$x \otimes (y \oplus z) = (x + \min(y, z)) \bmod p.$$

Right-hand side:

$$(x \otimes y) \oplus (x \otimes z) = \min\big((x + y) \bmod p,\ (x + z) \bmod p\big).$$

These two expressions are not always equal, since modular reduction can reorder the relative sizes of $(x + y) \bmod p$ and $(x + z) \bmod p$.

Counterexample. Let $p = 5$, $x = 2$, $y = 4$, $z = 0$. Then:

$$x \otimes (y \oplus z) = (2 + \min(4, 0)) \bmod 5 = (2 + 0) \bmod 5 = 2,$$

while

$$(x \otimes y) \oplus (x \otimes z) = \min((2 + 4) \bmod 5,\ (2 + 0) \bmod 5) = \min(1, 2) = 1.$$

Thus, distributivity fails. Therefore, sometimes the algebraic structure $\mathcal{M}_p = (S, \oplus, \otimes)$, is called the near-semiring.

The lack of distributivity has important implications. Many tropical cryptographic protocols rely on distributive laws for their proofs of correctness. Such constructions cannot be directly transplanted into the Min–Plus–Modulo setting. Nevertheless, the algebra remains attractive for cryptography: $\oplus$ introduces idempotent order-preserving behavior, while $\otimes$ provides modular wraparound, injecting nonlinearity. Protocols must therefore be designed to exploit these properties without relying on distributivity.

2.2 Polynomials over the Min-Plus-Modulo Algebra

A polynomial of degree d over $\mathcal{M}_p$ in one variable x is a formal expression.

$$F(x) = c_0 \oplus (c_1 \otimes x) \oplus (c_2 \otimes x^{\otimes 2}) \oplus \cdots \oplus (c_d \otimes x^{\otimes d}),$$

with coefficients $c_i \in S$, A *monomial* in this setting is expressed as:

$$c_i \otimes x^{\otimes i} = (c_i + i \cdot x) \bmod p,$$

so the polynomial

$$F(x) = \bigoplus_{i=0}^{d} \left(c_i \otimes x^{\otimes i}\right)$$

evaluates

$$F(x) = \min_{0 \le i \le d} \big((c_i + i \cdot x) \bmod p\big),$$

which is precisely the functional form, and where powers are defined as iterated $\otimes$-multiplication:

$$x^{\otimes k} := \underbrace{x \otimes x \otimes \cdots \otimes x}_{k \text{ times}}, \qquad x^{\otimes 0} := 0.$$

2.3 Polynomial Addition

For polynomials $F(x) = \bigoplus_{i=0}^{d}(c_i \otimes x^{\otimes i})$ and $G(x) = \bigoplus_{i=0}^{d}(d_i \otimes x^{\otimes i})$, define

$$(F \oplus G)(x) = \bigoplus_{i=0}^{d} \big((c_i \oplus d_i) \otimes x^{\otimes i}\big).$$

This operation is commutative, associative, and idempotent, with neutral element the polynomial having all coefficients equal to $p - 1$.

2.4 Polynomial Multiplication

For $F(x) = \bigoplus_{i=0}^{m}(c_i \otimes x^{\otimes i})$ and $G(x) = \bigoplus_{j=0}^{n}(d_j \otimes x^{\otimes j})$, define

$$(F \otimes G)(x) = \bigoplus_{k=0}^{m+n} \left(\min_{i+j=k} (c_i \otimes d_j) \otimes x^{\otimes k} \right).$$

Here each product $c_i \otimes d_j$ is computed as $(c_i + d_j) \bmod p$.

The product uses modular addition of coefficients and sums the exponents. Afterward, terms of the same degree are combined using the tropical minimum.

The *degree* of a polynomial in $\mathcal{M}_p[x]$ is the highest exponent among its monomials with non-infinite coefficients. Each polynomial has a *canonical form*: a list of monomials where, for each degree, only the term with the minimum coefficient (modulo p) is retained. This ensures uniqueness and simplifies comparisons.

2.5 Example: Polynomial Operations in $\mathcal{M}_5$

Let $p = 5$. Consider

$$F(x) = 1 \otimes x^{\otimes 0} \ \oplus \ 3 \otimes x^{\otimes 1} \ \oplus \ 4 \otimes x^{\otimes 2}, \qquad G(x) = 2 \otimes x^{\otimes 0} \ \oplus \ 1 \otimes x^{\otimes 1} \ \oplus \ 0 \otimes x^{\otimes 2}.$$

Addition.

$$(F \oplus G)(x) = (1 \oplus 2) \otimes x^{\otimes 0} \ \oplus \ (3 \oplus 1) \otimes x^{\otimes 1} \ \oplus \ (4 \oplus 0) \otimes x^{\otimes 2}.$$

Hence,

$$F \oplus G = 1 \otimes x^{\otimes 0} \ \oplus \ 1 \otimes x^{\otimes 1} \ \oplus \ 0 \otimes x^{\otimes 2}.$$

Multiplication.

With coefficient vectors $(c_0, c_1, c_2) = (1, 3, 4)$ and $(d_0, d_1, d_2) = (2, 1, 0)$,

$$(F \otimes G)(x) = \bigoplus_{k=0}^{4} \left(\min_{i+j=k} (c_i \otimes d_j) \otimes x^{\otimes k} \right).$$

Explicit computation yields:

$$\begin{aligned}
k = 0: &\quad (1 + 2) \bmod 5 = 3, \\
k = 1: &\quad \min\{(1 + 1) \bmod 5, (3 + 2) \bmod 5\} = \min\{2, 0\} = 0, \\
k = 2: &\quad \min\{(1 + 0) \bmod 5, (3 + 1) \bmod 5, (4 + 2) \bmod 5\} = \min\{1, 4, 1\} = 1, \\
k = 3: &\quad \min\{(3 + 0) \bmod 5, (4 + 1) \bmod 5\} = \min\{3, 0\} = 0, \\
k = 4: &\quad (4 + 0) \bmod 5 = 4.
\end{aligned}$$

Therefore,

$$(F \otimes G)(x) = 3 \otimes x^{\otimes 0} \ \oplus \ 0 \otimes x^{\otimes 1} \ \oplus \ 1 \otimes x^{\otimes 2} \ \oplus \ 0 \otimes x^{\otimes 3} \ \oplus \ 4 \otimes x^{\otimes 4}.$$

2.6 Illustrative Examples

We now provide concrete examples to demonstrate how polynomial operations behave in the Min–Plus–Modulo (MPM) algebra.

Example 1.

Let $p = 251$, and define two polynomials

$$P(x) = 3 \otimes x^{\otimes 2} \ \oplus \ 7 \otimes x^{\otimes 1} \ \oplus \ 2 \otimes x^{\otimes 3}, \qquad Q(x) = 4 \otimes x^{\otimes 1} \ \oplus \ 1 \otimes x^{\otimes 0}.$$

Step 1: Compute pairwise tropical–modular products between terms. Each product contributes a term with degree $i + j$ and coefficient $(a_i + d_j) \bmod p$.

Term from $P(x)$	Term from $Q(x)$	Degree	Coefficient
$3 \otimes x^2$	$4 \otimes x^1$	3	$(3+4) \bmod 251 = 7$
$3 \otimes x^2$	$1 \otimes x^0$	2	$(3+1) \bmod 251 = 4$
$7 \otimes x^1$	$4 \otimes x^1$	2	$(7+4) \bmod 251 = 11$
$7 \otimes x^1$	$1 \otimes x^0$	1	$(7+1) \bmod 251 = 8$
$2 \otimes x^3$	$4 \otimes x^1$	4	$(2+4) \bmod 251 = 6$
$2 \otimes x^3$	$1 \otimes x^0$	3	$(2+1) \bmod 251 = 3$

Step 2: Group by degree and apply $\oplus = \min$:

$$\deg(1) : 8, \quad \deg(2) : \min(4, 11) = 4, \quad \deg(3) : \min(7, 3) = 3, \quad \deg(4) : 6.$$

Thus,

$$P(x) \otimes Q(x) = 8 \otimes x^{\otimes 1} \ \oplus \ 4 \otimes x^{\otimes 2} \ \oplus \ 3 \otimes x^{\otimes 3} \ \oplus \ 6 \otimes x^{\otimes 4}.$$

Example 2.

Consider the monomial

$$f(x, y, z) = x \otimes x \otimes y \otimes z \otimes z = (x + x + y + z + z) \bmod p.$$

For $x = 3$, $y = 5$, $z = 2$, $p = 7$ we obtain

$$f = (3 + 3 + 5 + 2 + 2) \bmod 7 = 15 \bmod 7 = 1.$$

The Min–Plus–Modulo degree of this monomial is 5, equal to the number of variables summed.

Example 3.

Let

$$p(x) = x \otimes x \ \oplus \ x \otimes x \otimes x.$$

For $x = 4$ and $p = 11$:

$$p(x) = \min\big((4+4) \bmod 11,\ (4+4+4) \bmod 11\big) = \min(8,\ 12 \bmod 11) = \min(8, 1) = 1.$$

Although the coefficients in both monomials are zero, the polynomial evaluates non-trivially because of modular wrap-around, highlighting the obfuscating power of the modulo operation.

Example 4.

Define

$$p(x) = x \otimes 2 \ \oplus \ x \otimes x \otimes 3, \qquad q(x) = x \ \oplus \ x \otimes 5.$$

For $x = 1$ and $p = 10$:

$$p(x) = \min((1+2) \bmod 10,\ (1+1+3) \bmod 10) = \min(3,5) = 3,$$

$$q(x) = \min(1,\ (1+5) \bmod 10) = \min(1,6) = 1,$$
$$r(x) = p(x) \otimes q(x) = (3+1) \bmod 10 = 4.$$

Thus the resulting value is $r(x) = 4$, illustrating how tropical minimum and modular addition interacts in composed polynomials.

Although the method can be applied to multivariate tropical polynomials as well, we focus on one-variable tropical polynomials in this work.

3 Proposed Digital Signature Technique

This section presents a digital signature technique constructed over the Min–Plus–Modulo (MPM) algebra. The design leverages the combination of tropical addition (minimum) and modular addition, resulting in a lightweight yet a non-linear protocol that is well-suited for post-quantum cryptographic applications.

Let $\mathcal{M}_p = (\mathbb{Z}_p, \oplus, \otimes)$ denote the Min–Plus–Modulo algebra, where p is a prime modulus, and the operations are described as:

$$x \oplus y = \min(x, y), \qquad x \otimes y = (x + y) \bmod p.$$

All polynomials in this scheme are defined over $\mathcal{M}_p[x]$.

Key Generation

1. Select two secret irreducible polynomials $X(x), Y(x) \in \mathcal{M}_p[x]$ so that,

$$\deg(X) + \deg(Y) = 2d,$$

   where d represents a security parameter. The coefficients for X and Y are uniformly selected from the range $[0, r] \subseteq \mathbb{Z}_p$.
2. Calculate the public key polynomial:

$$M(x) = X(x) \otimes Y(x) = (X(x) + Y(x)) \bmod p.$$

3. Fix a cryptographic hash function H (e.g., SHA3–512), together with a deterministic process for transforming a hash output into a polynomial $P(x) \in \mathcal{M}_p[x]$ of degree d.

Signature Generation

To sign a message $m \in \{0,1\}^*$:

1. Compute the message hash and derive

$$P(x) \in \mathcal{M}_p[x], \qquad \deg(P) = d,$$

using the public hash-to-polynomial procedure.

2. Randomly sample two ephemeral polynomials

$$U(x), V(x) \in \mathcal{M}_p[x],$$

so that $\deg(U) = \deg(Y)$, $\deg(V) = \deg(X)$, and the coefficients in $[0, r]$.

3. Compute the intermediate values:

$$N(x) = U(x) \otimes V(x),$$

$$A(x) = P(x) \otimes X(x) \otimes U(x),$$
$$B(x) = P(x) \otimes Y(x) \otimes V(x).$$

4. The signature is a four-tuple:

$$\sigma = (P(x), A(x), B(x), N(x)).$$

Signature Verification

To verify a signature $\sigma = (P, A, B, N)$ in a message m, given the public key $M(x)$, the verifier must perform the following steps:

1. Recompute the message polynomial $P'(x)$ from m. Reject if $P \neq P'$.
2. Check degree constraints:

$$\deg(A) = \deg(B) = 3d, \qquad \deg(N) = 2d.$$

Reject if not satisfied.

3. Ensure that A and B are not constant tropical multiples of $P \otimes M$ or $P \otimes N$. Reject otherwise.
4. Verify coefficient ranges:

$$\text{coeff}(A), \text{coeff}(B) \in [0, 3r], \qquad \text{coeff}(N) \in [0, 2r].$$

5. Compute:

$$W(x) = A(x) \otimes B(x),$$
$$Z(x) = P(x) \otimes P(x) \otimes M(x) \otimes N(x).$$

6. Accept the signature if and only if

$$W(x) = Z(x).$$

Correctness

Correctly generated signatures satisfy the verification condition. Indeed,

$$\begin{aligned} W(x) &= A(x) \otimes B(x) \\ &= \big(P(x) \otimes X(x) \otimes U(x)\big) \otimes \big(P(x) \otimes Y(x) \otimes V(x)\big) \end{aligned}$$

$$
\begin{aligned}
&= \big(P(x) \otimes P(x)\big) \otimes \big(X(x) \otimes Y(x)\big) \otimes \big(U(x) \otimes V(x)\big) \\
&= (P(x) \otimes P(x)) \otimes M(x) \otimes N(x) \\
&= Z(x).
\end{aligned}
$$

Hence the scheme is correct.

Remark 1 It is essential to ensure that the degrees of the private polynomials $X(x), Y(x), U(x), V(x)$ and the hash polynomial $P(x)$ are carefully selected to maintain the correct structure and dimension of the final signature components. Incorrect degree assignments could lead to invalid or trivially forgeable signatures.

Remark 2 One common forgery attempt is to construct signatures of the form $\sigma = (P(x), P(x) \otimes M(x), P(x) \otimes N(x), N(x))$, which may satisfy some superficial checks. However, our verification condition explicitly prevents such cases by comparing full tropical products and requiring consistency across all four signature components.

Remark 3 A distinguishing feature of the Min–Plus–Modulo algebra is the failure of distributivity of $\otimes$ over $\oplus$. While this prevents the algebra from forming a semiring in the strict sense, it also has positive cryptographic implications. Many known tropical algebra attacks—such as linearization or reductions to classical matrix problems—rely fundamentally on distributive laws to simplify expressions. The absence of distributivity in $\mathcal{M}_p$ therefore disrupts these attack strategies, increasing resistance against algebraic cryptanalysis.

4 Parameters Selection

The security and efficiency of the proposed Min–Plus–Modulo signature scheme depend on several key parameters: the modulus p, the degree parameter d, and the coefficient bound r. These must be selected carefully to resist known cryptanalytic techniques while ensuring computational efficiency.

Choice of modulus p***.***

We require p to be a large prime to avoid factorization-based weaknesses and to maximize the uniformity of modular reduction. To provide 128-bit security, we recommend $p \approx 2^{256}$, matching the bit-length of commonly used elliptic curve groups (e.g., secp256r1). For efficiency-constrained deployments, a smaller p (e.g., 2^{192}) may be chosen at the cost of reduced security.

Degree parameter d***.***

The parameter d determines the degree of the hash-derived polynomial $P(x)$ and thus the dimensions of the signature components:

$$\deg(A) = \deg(B) = 3d, \qquad \deg(N) = 2d.$$

To achieve post-quantum security, we require that brute-force enumeration of polynomial relations is infeasible. A conservative choice is $d \geq 128$, ensuring exponential complexity for degree-based attacks. For lightweight applications, $d = 64$ may be acceptable.

Coefficient bound r.

The bound r controls the distribution of coefficients in private polynomials $X(x), Y(x), U(x), V(x)$. A small r risks structural leakage, while a large r increases noise but may reduce efficiency. We recommend $r \approx \sqrt{p}$, which balances entropy against computational cost. For $p \approx 2^{256}$, this corresponds to $r \approx 2^{128}$.

Hash function H.

We require a collision-resistant hash that outputs at least $2d \log_2 p$ bits, ensuring full entropy in the derived polynomial $P(x)$. SHA3–512 is a conservative choice that meets this requirement.

Let $B = H(m)$ represent the 512-bit output derived from applying the SHA3-512 hash function to an input message m. This bit string B must be converted into a one-variable polynomial $P(x)$ of degree d defined in the MPM algebra $\mathbb{M}_p[x]$, where all coefficients are constrained to the interval $[0, r]$ with $r = 2^{128}$, and d corresponds to our chosen security parameter, taking values of 100, 150, or 200.

The conversion algorithm begins with bit string preparation. Given that each polynomial requires $(d + 1)$ coefficients and each coefficient must be represented using $\log_2(r+1)\rceil = (d+1) \times 129$ bits to span the entire range $[0, r]$, the total bit requirement amounts to $(d + 1) \times 129$ bits. To accommodate this requirement using the 512-bit hash output, we compute the number of necessary concatenations as

$$k = \left\lceil \frac{(d + 1) \times 129}{512} \right\rceil$$

We then construct an extended bit string B' by concatenating k copies of the original hash output B, formally expressed as

$$B' = \underbrace{B \parallel B \parallel \cdots \parallel B}_{k \text{ times}}$$

The coefficient extraction phase processes this extended bit string B' in a left-to-right manner. For each polynomial term index j ranging from 0 to d, we extract a contiguous 129-bit block beginning at bit position $129 \times j$ in B'. This bit sequence is converted to its corresponding integer value c_j, which is then reduced modulo $(r + 1)$ to guarantee that the final coefficient falls within the designated range $[0, r]$. The coefficient c_j is subsequently assigned to the monomial x^j in the polynomial $p(x)$. Through this systematic procedure, the hash output is deterministically mapped to a properly formatted polynomial suitable for all subsequent MPM algebraic operations within the signature scheme.

4.1 Underlying Hard Problem

We formalize the algebraic assumption that underpins the security of the suggested signature system. The hardness stems from the difficulty of recovering private factors from a public polynomial constructed under the Min–Plus–Modulo (MPM) algebra.

Definition 1 (MPM Polynomial Factorization Problem (MPM-PFP)) Let $\mathcal{M}_p = (\mathbb{Z}_p, \oplus, \otimes)$ denote the Min–Plus–Modulo algebra, with $a \oplus b = \min(a, b)$ and $a \otimes b = (a + b) \bmod p$. Given a public polynomial

$$M(x) = X(x) \otimes Y(x),$$

where $X(x), Y(x) \in \mathcal{M}_p[x]$ are irreducible polynomials of bounded degree and coefficients, the problem is to recover the pair $(X(x), Y(x))$ from $M(x)$.

For cryptographically relevant parameters (p, d, r), the MPM Polynomial Factorization Problem cannot be solved in sub-exponential time by any classical or quantum algorithm.

This assumption underpins the unforgeability of our signature scheme. The public key $M(x)$ hides its private factors $(X(x), Y(x))$ due to the combined effects of tropical addition and modular reduction, both of which obscure algebraic structure. Moreover, the non-distributivity of $\otimes$ over $\oplus$ prevents linearization attacks commonly used in tropical cryptanalysis. Forging a signature without knowledge of the private keys would require solving systems of tropical–modular equations, which we assume to be computationally infeasible.

The MPM Polynomial Factorization Problem (MPM–PFP) is introduced in this work as a new and central cryptographic assumption. Its presumed hardness arises from the distinctive behaviour of the MPM structure—most notably, the breakdown of distributivity—which blocks the use of standard linear-algebraic tools. This makes the problem fundamentally different from traditional factorization challenges found in either pure tropical settings or classical algebraic rings.

5 Most Possible Attack

The security of the Min-Plus-Modulo scheme arises from the interplay of several key properties. First, the use of modular arithmetic introduces a high degree of non-linearity, effectively obfuscating structural patterns and providing resistance against linear algebra-based attacks. Second, the incorporation of fresh random polynomials into each signature ensures that even when the same message is signed multiple times, the resulting signatures remain unique, thwarting replay and pattern-based attacks. Third, the scheme binds the signature tightly to the specific message through the message polynomial $P(x)$, which is derived directly from the message hash, thereby preventing content substitution attacks. Finally, the tropical sparsity inherent in the min operation naturally limits the amount of information that can be inferred from each polynomial degree, making full polynomial recovery computationally infeasible. Collectively, these properties ensure that the scheme remains lightweight, computationally efficient, and resistant to both classical and quantum adversaries.

5.1 Panny's Factorization-Based Forgery Attack

In the classical tropical semiring, polynomial factorization is non-unique: a product $M = X \otimes Y$ may admit multiple distinct decompositions $M = A \otimes B$ beyond the original (X, Y). Panny's attack exploits this phenomenon by substituting an alternative factorization into the signature equations. By setting $U = B$ and $V = A$, a forged signature can sometimes satisfy the verification relation, because tropical distributivity ensures that

$$(P \otimes A \otimes B) \otimes (P \otimes B \otimes A) = P^{\otimes 2} \otimes M \otimes N, \quad N = A \otimes B = M.$$

In the Min–Plus–Modulo (MPM) algebra, however, the situation changes. Here multiplication is defined as $(a \otimes b) = (a + b) \bmod p$, and the coefficient rule becomes

$$M_k = \min_{i+j=k} \big((A_i + B_j) \bmod p\big).$$

Because modular reduction disrupts distributivity, the classical attack recipe does not carry over verbatim. To further mitigate structural forgeries,one may enforce irreducibility conditions on private polynomials or add explicit verification checks to prevent degenerate cases such as $N = M$ from being accepted. These protective measures, combined with the inherent algebraic structure of MPM operations, significantly enhance the scheme's resistance to factorization-based forgery attacks.

5.2 Brown–Monico Key Recovery via Tropical "Division"

Brown and Monico describe an attack that recovers private polynomial factors by assembling systems of tropical linear equations from multiple observed signatures. Informally, if a signer reveals components of the form

$$A^{(i)}(x) = P^{(i)}(x) \otimes X(x) \otimes U^{(i)}(x),$$

an adversary who can "divide" by the known $P^{(i)}(x)$ may form candidate masked values $D^{(i)}(x) \approx X(x) \otimes U^{(i)}(x)$. Here the notation "$\div$" or "$\oslash$" must be interpreted carefully: in the classical (min,+) tropical setting, one typically performs coefficientwise subtraction in the underlying integer domain (or cancels matched monomials when degrees align).

Accumulating many such relations produces an (overdetermined) system of min–plus constraints in the coefficients of $X(x)$; under unfavorable parameter choices (low-entropy masks, small coefficient ranges, small degrees) Such systems can be solved in practice by combinatorial or heuristic solvers, yielding key recovery. Although the worst-case problem of solving arbitrary tropical linear systems is NP-hard, this worst-case hardness does not preclude efficient attacks on structured or low-entropy instances.

In the Min–Plus–Modulo (MPM) algebra the attack becomes more delicate. Modular reduction $(a + b) \bmod p$ and the failure of distributivity disturb the simple coefficientwise subtraction used for tropical "division", so the adversary's derived

equations are noisier and often ambiguous. Nevertheless, MPM does not eliminate the risk: poor parameter choices (small p, reused masks, narrow coefficient support) still permit practical recovery. Consequently, implementing the scheme securely requires fresh random masks per signature, large modulus and coefficient ranges, degree separation to prevent trivial cancellations, and ideally a proof-of-knowledge layer that avoids revealing raw masked products. Our parameter selection of $p = 2^{256}$, $r = 2^{128}$, and $d \geq 100$ provides substantial protection against this class of attacks by ensuring sufficient entropy and algebraic complexity.

6 Performance and Signature Size

The efficiency of the proposed Min–Plus–Modulo (MPM) signature scheme is primarily determined by operations on polynomials in the Min–Plus–Modulo algebra. Let d denote the degree of the private polynomials, p the prime modulus, and r the coefficient bound.

Key generation involves computing

$$M(x) = X(x) \otimes Y(x), \tag{1}$$

where $\otimes$ denotes the Min–Plus–Modulo convolution. Each multiplication of two degree-d polynomials requires $O(d^2)$ modular additions followed by coefficientwise minima. Since key generation is performed only once per key pair, this cost is negligible in practice.

To sign a message, the signer computes three polynomials:

$$A(x), \quad B(x), \quad N(x), \tag{2}$$

as tropical-modular products of degree up to $3d$. Each polynomial product involves $O(d^2)$ modular operations, resulting in overall signature generation complexity $O(d^2)$. Hashing the message into a polynomial (via SHA-512 plus coefficient extraction) is independent of d and incurs minimal overhead.

Verification requires recomputing

$$W(x) = A(x) \otimes B(x), \quad Z(x) = \big(2P(x) + M(x) + N(x)\big) \bmod p, \tag{3}$$

followed by a single equality check. As with signing, verification complexity is quadratic in d, relying only on modular additions and minima. These lightweight operations make the scheme practical even for high-degree polynomials.

Overall, both signing and verification scale quadratically with the security parameter d, while remaining computationally lighter than classical number-theoretic schemes, where modular multiplications or exponentiations dominate.

A signature in the MPM scheme consists of the tuple:

$$\sigma = \big(P(x), A(x), B(x), N(x)\big). \tag{4}$$

The size of each component is determined by its degree and coefficient bounds:

- $P(x)$: degree d, coefficients in $\mathbb{Z}_p \Rightarrow (d+1)\log_2 p$ bits,
- $A(x), B(x)$: degree $3d$, coefficients in $[0, 3r] \Rightarrow 2(3d+1)\log_2(3r+1)$ bits,
- $N(x)$: degree $2d$, coefficients in $[0, 2r] \Rightarrow (2d+1)\log_2(2r+1)$ bits.

Hence, the total signature size is approximately:

$$|\sigma| \approx (d+1)\log_2 p + 2(3d+1)\log_2(3r+1) + (2d+1)\log_2(2r+1) \text{ bits.} \quad (5)$$

We evaluated the MPM scheme for three polynomial degrees: $d = 100, 150, 200$. Each test was performed in over 100 independent trials. The average results are summarized in Table 1. For our computer simulations, we used Python 3 on Windows 11, Core i3, 4GB RAM laptop.

Table 1 MPM Signature Benchmarks (100 trials, vertical layout)

Parameter	$d = 100$	$d = 150$	$d = 200$
Verif. Time (s)	0.87	2.35	4.83
Sign. Size (kB)	12.25	18.29	24.28
Pub. Key Size (kB)	3.09	4.62	6.14
Priv. Key Size (kB)	3.13	4.68	6.23
Memory (Verif.) MB	5.05	11.22	19.78
Memory (Total) MB	5.61	14.28	27.53

From these results, we observe that verification time, signature size, and memory usage scale approximately linearly with d. Even for $d = 200$, verification completes in under 5 seconds on a standard desktop machine, while memory usage remains below 30 MB. For practical security, we recommend setting the coefficient bound to $r \approx \sqrt{p}$, balancing computational cost and entropy. For $p \approx 2^{256}$, this corresponds to $r \approx 2^{128}$.

Comparison to Classical Tropical Signatures

Compared to the classical tropical signature scheme (without modular reduction), the MPM scheme introduces only minor additional computational overhead. Modular reduction substantially increases security by complicating linearization-based attacks (e.g., Brown–Monico), while signature sizes remain of the same asymptotic order $O(d \log p)$. Overall, the MPM scheme offers a practical trade-off between security and efficiency, suitable for post-quantum cryptographic applications.

7 Discussion, Limitations, and Future Work

This work introduced MPM-Sign, a digital signature scheme built on a new Min–Plus–Modulo hybrid algebra. Our results show that the scheme achieves correctness, maintains competitive efficiency, and provides stronger resistance to linearization attacks than existing purely tropical constructions.

Limitations. The present study establishes the theoretical basis of the scheme and offers an initial performance evaluation. The main limitation lies in the fact that the security of MPM-Sign depends on the hardness of the newly proposed MPM–PFP assumption. While our analysis supports its plausibility, extensive independent cryptanalysis will be necessary to build long-term confidence in this assumption. Moreover, our performance benchmarks are currently derived from software-level simulations. Real-world performance—particularly on constrained platforms such as IoT devices—remains an open question and warrants targeted experimental validation.

Future Work. Looking ahead, we highlight two immediate research directions:

- **Cryptanalysis.** Organizing a public cryptanalytic challenge to encourage deeper investigation into the hardness of the MPM–PFP and to evaluate the scheme's resilience under broader attack models.
- **Protocol Extensions.** Designing a Key Encapsulation Mechanism (KEM) based on the MPM algebra, with the goal of building a more complete suite of post-quantum cryptographic primitives.

References

[1] Grigoriev, D., Shpilrain, V.: Tropical cryptography. Communications in Algebra **42**(6), 2624–2632 (2014)

[2] Grigoriev, D., Shpilrain, V.: Tropical cryptography ii: extensions by homomorphisms. Communications in Algebra **47**(10), 4224–4229 (2019)

[3] Habeeb, M., Kahrobaei, D., Koupparis, C., Shpilrain, V.: Public key exchange using semidirect product of (semi)groups. In: Applied Cryptography and Network Security. Lecture Notes in Computer Science, vol. 7954, pp. 475–486 (2013)

[4] Kahrobaei, D., Shpilrain, V.: Using semidirect product of (semi)groups in public key cryptography. In: Computability in Europe. Lecture Notes in Computer Science, vol. 9709, pp. 132–141 (2016)

[5] Kotov, M., Ushakov, A.: Analysis of a key exchange protocol based on tropical matrix algebra. Journal of Mathematical Cryptology **12**(3), 137–141 (2018)

[6] Kim, K.H., Roush, F.W.: Factorization of polynomials in one variable over the tropical semiring. arXiv preprint math/0501167 (2005)

[7] Chen, J., Grigoriev, D., Shpilrain, V.: Tropical cryptography iii: digital signatures. Cryptology ePrint Archive (2023)

[8] Chen, J., Grigoriev, D., Shpilrain, V.: Tropical cryptography iii: digital signatures. Journal of Mathematical Cryptology **18**(1), 20240005 (2024)

[9] Panny, L.: Forging tropical signatures. Cryptology ePrint Archive (2023)

[10] Brown, D.R.L., Monico, C.: More forging (and patching) of tropical signatures. Cryptology ePrint Archive (2023)

[11] Géraud-Stewart, R., Naccache, D., Yifrach-Stav, O.: Fiat-shamir goes tropical. Cryptology ePrint Archive (2023)

[12] Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Advances in Cryptology—CRYPTO'86, pp. 186–194 (1987)

[13] Ahmed, K., Pal, S., Mohan, R.: A review of the tropical approach in cryptography. Cryptologia **47**(1), 63–87 (2023)

[14] Buchinskiy, I.M., Kotov, M.V., Treier, A.V.: On the complexity of the problem of solving systems of tropical polynomial equations of degree two. In: International Conference on Mathematical Optimization Theory and Operations Research, pp. 73–84. Springer, ??? (2024)

[15] Theobald, T.: On the frontiers of polynomial computations in tropical geometry. Journal of Symbolic Computation **41**(12), 1360–1375 (2006)

[16] Butkovič, P.: Max-linear Systems: Theory and Algorithms. Springer, London (2010)