TROPICAL CRYPTOGRAPHY III: DIGITAL SIGNATURES

JIALE CHEN, DIMA GRIGORIEV, AND VLADIMIR SHPILRAIN

Abstract. We use tropical algebras as platforms for a very efficient digital signature protocol. Security relies on computational hardness of factoring one-variable tropical polynomials; this problem is known to be NP-hard.

Keywords: tropical algebra, digital signature, factoring polynomials

1. Introduction

In our earlier papers [2], [3], we employed *tropical algebras* as platforms for cryptographic schemes by mimicking some well-known classical schemes, as well as newer schemes like [4], [5], in the "tropical" setting. What it means is that we replaced the usual operations of addition and multiplication by the operations $\min(x, y)$ and $x + y$, respectively.

An obvious advantage of using tropical algebras as platforms is unparalleled efficiency because in tropical schemes, one does not have to perform any multiplications of numbers since tropical multiplication is the usual addition, see Section 2. On the other hand, "tropical powers" of an element exhibit some patterns, even if such an element is a matrix over a tropical algebra. This weakness was exploited in [7] to arrange a fairly successful attack on one of the schemes in [2].

In this paper, we offer a digital signature scheme that uses tropical algebra of one-variable polynomials. Security of this scheme is based on computational hardness of factoring one-variable tropical polynomials. This problem is known to be NP-hard, see [6].

2. Preliminaries

We start by giving some necessary information on tropical algebras here; for more details, we refer the reader to the monograph [1].

Consider a tropical semiring $\mathbf{A}$, also known as the min-plus algebra due to the following definition. This semiring is defined as a linearly ordered set (e.g., a subset of reals) that contains 0 and is closed under addition, with two operations as follows:

$x \oplus y = \min(x, y)$

$x \otimes y = x + y$.

It is straightforward to see that these operations satisfy the following properties:

associativity:
$x \oplus (y \oplus z) = (x \oplus y) \oplus z$
$x \otimes (y \otimes z) = (x \otimes y) \otimes z$.

commutativity:
$x \oplus y = y \oplus x$
$x \otimes y = y \otimes x$.

distributivity:
$(x \oplus y) \otimes z = (x \otimes z) \oplus (y \otimes z)$.

There are some "counterintuitive" properties as well:
$x \oplus x = x$

$x \otimes 0 = x$

$x \oplus 0$ could be either 0 or x.

There is also a special "ϵ-element" $\epsilon = \infty$ such that, for any $x \in S$,

$\epsilon \oplus x = x$

$\epsilon \otimes x = \epsilon$.

2.1. **Tropical polynomials.** A (tropical) monomial in S looks like a usual linear function, and a tropical polynomial is the minimum of a finite number of such functions, and therefore a concave, piecewise linear function. The rules for the order in which tropical operations are performed are the same as in the classical case, see the example below. Still, we often use parenthesis to make a tropical polynomial easier to read.

Example 1. *Here is an example of a tropical monomial: $x \otimes x \otimes y \otimes z \otimes z$. The (tropical) degree of this monomial is 5. We note that sometimes, people use the alternative notation $x^{\otimes 2}$ for $x \otimes x$, etc.*

An example of a tropical polynomial is: $p(x, y, z) = 5 \otimes x \otimes y \otimes z \oplus x \otimes x \oplus 2 \otimes z \oplus 17 = (5 \otimes x \otimes y \otimes z) \oplus (x \otimes x) \oplus (2 \otimes z) \oplus 17$. This polynomial has (tropical) degree 3, by the highest degree of its monomials.

We note that, just as in the classical case, a tropical polynomial is canonically represented by an ordered set of tropical monomials (together with finite coefficients), where the order that we use here is deglex.

We also note that some tropical polynomials may look "weird":

Example 2. *Consider the polynomial $p(x) = (0 \otimes x) \oplus (0 \otimes x \otimes x)$. All coefficients in this polynomial are 0, and yet it is not the same as the polynomial $q(x) = 0$.*

Indeed, $q(x) \otimes r(x) = r(x)$ for any polynomial $r(x)$. On the other hand, if, say, $r(x) = 2 \otimes x$, then $p(x) \otimes r(x) = (2 \otimes x \otimes x) \oplus (2 \otimes x \otimes x \otimes x) \neq r(x)$.

In the following example, we show in detail how two tropical polynomials are multiplied and how similar terms are collected.

Example 3. *Let $p(x) = (2 \otimes x) \oplus (3 \otimes x \otimes x)$ and $q(x) = 5 \oplus (1 \otimes x)$. Then $p(x) \otimes q(x) = [(2 \otimes x) \otimes 5] \oplus [(2 \otimes x) \otimes (1 \otimes x)] \oplus [(3 \otimes x \otimes x) \otimes 5] \oplus [(3 \otimes x \otimes x) \otimes (1 \otimes x)] = (7 \otimes x) \oplus (3 \otimes x \otimes x) \oplus (8 \otimes x \otimes x) \oplus (4 \otimes x \otimes x \otimes x) = (7 \otimes x) \oplus (3 \otimes x \otimes x) \oplus (4 \otimes x \otimes x \otimes x)$.*

In this paper, our focus is on one-variable tropical polynomials, although one can use multivariate tropical polynomials instead.

3. Digital signature scheme description

Let $\mathbf{T}$ be the tropical algebra of one-variable polynomials over $\mathbf{Z}$, the ring of integers. The signature scheme is as follows.

Private: two polynomials $X, Y \in \mathbf{T}$ of the same degree d, with all coefficients in the range $[0, r]$, where r is one of the parameters of the scheme.

Public:
– polynomial $M = X \otimes Y$
– a hash function H (e.g., SHA-512) and a (deterministic) procedure for converting values of H to one-variable polynomials from the algebra $\mathbf{T}$ (see Section 4.1).

Signing a message m:

(1) Apply a hash function H to m. Convert $H(m)$ to a polynomial P of degree d from the algebra $\mathbf{T}$ using a deterministic public procedure.

(2) Select two random private polynomials $U, V \in \mathbf{T}$ of degree d, with all coefficients in the range $[0, r]$. Denote $N = U \otimes V$.

(3) The signature is the triple of polynomials $(P \otimes X \otimes U,\ P \otimes Y \otimes V,\ N)$.

Verification:

1. The verifier computes the hash $H(m)$ and converts $H(m)$ to a polynomial P of degree d from the algebra $\mathbf{T}$ using a deterministic public procedure.

2. (a) The verifier checks that the degrees of the polynomials $P \otimes X \otimes U$ and $P \otimes Y \otimes V$ (the first two polynomials in the signature) are both equal to $3d$, and the degree of the polynomial N is equal to $2d$. If not, then the signature is not accepted.

(b) The verifier checks that neither $P \otimes X \otimes U$ nor $P \otimes Y \otimes V$ is a constant multiple (in the tropical sense) of $P \otimes M$ or $P \otimes N$. If it is, then the signature is not accepted.

(c) The verifier checks that all coefficients in the polynomials $P \otimes X \otimes U$ and $P \otimes Y \otimes V$ are in the range $[0, 3r]$, and all coefficients in the polynomial N are in the range $[0, 2r]$. If not, then the signature is not accepted.

3. The verifier computes $W = (P \otimes X \otimes U) \otimes (P \otimes Y \otimes V)$. The signature is accepted if and only if $W = P \otimes P \otimes M \otimes N$.

Correctness is obvious since $W = (P \otimes X \otimes U) \otimes (P \otimes Y \otimes V) = P \otimes P \otimes (X \otimes Y) \otimes (U \otimes V) = P \otimes P \otimes M \otimes N$.

Remark 1. *Step 2 in the verification algorithm is needed to prevent trivial forgery, e.g. signing by the triple of polynomials* $(P \otimes M,\ P \otimes N,\ N)$*, in which case* $(P \otimes M) \otimes (P \otimes N) = P \otimes P \otimes M \otimes N$.

Remark 2. *Here is how one can check whether or not one given tropical polynomial, call it* $R(x)$*, is a constant multiple (in the tropical sense) of another given tropical polynomial (of the same degree), call it* $S(x)$.

Let $r_i \in \mathbf{Z}$ *denote the coefficient at the monomial* $x^{\otimes i}$ *in* $R(x)$*, and* $s_i \in \mathbf{Z}$ *denote the coefficient at the monomial* $x^{\otimes i}$ *in* $S(x)$*. If* $R(x) = c \otimes S(x)$ *for some* $c \in \mathbf{Z}$*, then* $r_i = s_i + c$ *for every* i*. Here "+" means the "classical" addition in* $\mathbf{Z}$.

Therefore, to check if $R(x)$ is a constant multiple of $S(x)$, one checks if $(r_i - s_i)$ is the same integer for every i.

4. Key generation and suggested parameters

The suggested degree d of each polynomial X, Y, U, V is 150.

All coefficients of monomials in the polynomials X, Y, U, V are selected uniformly at random from integers in the range $[0, r]$, where $r = 127$. We emphasize that, in contrast with the "classical" case, if the coefficient at a monomial is 0, this does not mean that this monomial is "absent" from the polynomial.

4.1. **Converting $H(m)$ to a tropical polynomial over Z.** We suggest using a hash functions from the SHA-2 family, specifically SHA-512. We assume the security properties of SHA-512, including collision resistance and preimage resistance. We also assume that there is a standard way to convert $H(m)$ to a bit string of length 512. Then a bit string can be converted to a tropical polynomial $P = P(x)$ over $\mathbf{Z}$ using the following *ad hoc* procedure.

Let $B = H(m)$ be a bit string of length 512. We will convert B to a one-variable tropical polynomial P of degree $d = 150$ over $\mathbf{Z}$. We therefore have to select 151 coefficients for monomials in P, and we want to have these coefficients in the range [0, 127]. With 7 bits for each coefficient, we need $151 \cdot 7 = 1057$ bits in total.

(1) Concatenate 3 copies of the bit string B to get a bit string of length 1536.

(2) Going left to right, convert 7-bit block $\#j$ to an integer and use it as the coefficient at the monomial $x^{\otimes j}$.

(3) After we use $7 \cdot 151 = 1057$ bits, all monomials in the polynomial $P = P(x)$ will get a coefficient.

4.2. **Multiplying two tropical polynomials.** Let $R(x)$ and $S(x)$ be two one-variable tropical polynomials of degree d and g, respectively. We want to compute $R(x) \otimes S(x)$.

Note that a one-variable tropical monomial, together with a coefficient, can be represented by a pair of integers (k, l), where k is the coefficient and l is the degree. Our goal is therefore to compute the coefficient at every monomial of degree from 0 to $d + g$ in the product $R(x) \otimes S(x)$.

Suppose we want to compute the coefficient at the monomial of degree m, $0 \leq m \leq d+g$. Then we go over all coefficients r_i at the monomials of degrees $i \leq m$ in the polynomial $R(x)$ and add (in the "classical" sense) r_i to s_j, where s_j is the coefficient at the monomial of degree $j = m - i$ in the polynomial $S(x)$.

Having computed all such sums $r_i + s_j$, we find the minimum among them, and this is the coefficient at the monomial of degree m in the polynomial $R(x) \otimes S(x)$.

5. What is the hard problem here?

The (computationally) hard problem that we employ in our construction is factoring one-variable tropical polynomials. This problem is known to be NP-hard, see [6].

Since recovering the private tropical polynomials X and Y from the public polynomial $M = X \otimes Y$ is exactly the factoring problem, we see that inverting our candidate one-way function $f(X, Y) = X \otimes Y$ is NP-hard.

However, the private tropical polynomials X and Y are involved also in the signature. For example, from the polynomial $W = P \otimes X \otimes U$ the adversary can recover $X \otimes U$ because the polynomial P is public. The polynomial U is private, so it looks like the adversary is still facing the factoring problem. However, the adversary now knows two products involving the polynomial X, namely $X \otimes Y$ and $X \otimes U$. Therefore, we have a somewhat different problem here: finding a common divisor of two given polynomials.

This problem is easy for "classical" one-variable polynomials over $\mathbf{Z}$. In particular, any classical one-variable polynomial over $\mathbf{Z}$ has a unique factorization (up to constant multiples) as a product of irreducible polynomials. In contrast, a one-variable tropical polynomial can have an exponential number of incomparable factorizations [6]. Furthermore, it was shown in [6] that two one-variable tropical polynomials may not have a unique g.c.d. All this makes it appear likely that the problem of finding the g.c.d. of two (or more) given one-variable tropical polynomials is computationally hard. No polynomial-time algorithm for solving this problem is known. More about this in Section 6.

6. Possible attacks

The most straightforward attack is trying to factor the tropical polynomial $M = X \otimes Y$ as a product of two tropical polynomials X and Y. As we have pointed out before, this problem is known to be NP-hard [6]. In our situation, there is an additional restriction on X and Y to be of the same degree d, to pass Step 2 of the verification procedure.

If one reduces the equation $M = X \otimes Y$ to a system of equations in the coefficients of X and Y, then one gets a system of $2d+1$ quadratic (tropical) equations in $2(d+1)$ unknowns. With d large enough, such a system is unapproachable; in fact, solving a system of quadratic tropical equations is known to be NP-hard [9]. The size of the key space for X (or Y) with suggested parameters is $128^{151} = 2^{1057}$, so the brute force search is infeasible.

It is unclear whether accumulating (from different signatures) many tropical polynomials of the form $M_i = X \otimes U_i$, with different (still unknown) polynomials U_i can help recover X. With each new M_i, the attacker gets $d+1$ new unknowns (these are coefficients of U_i) and $2d+1$ new equations. There is a well-known trick of reducing a system of quadratic equations to a system of linear equations by replacing each product of two unknowns by a new unknown. However, the number of pairs of unknowns increases by d^2 with each new U_i. Therefore, a system of linear equations like that will be grossly underdetermined, resulting in a huge number of solutions for the new unknowns, thus making solving the original system (in the old unknowns) hard, especially given the restrictions on the old unknowns tacitly imposed by Step 2(c) of the verification procedure.

7. Performance and signature size

For our computer simulations, we used Apple MacBook Pro, M1 CPU (8 Cores), 16 GB RAM computer. Python code is available, see [8].

We note that a one-variable tropical monomial, together with a coefficient, can be represented by a pair of integers (k, l), where k is the coefficient and l is the degree of the monomial. Then a one-variable tropical polynomial of degree 150 is represented by 151 such pairs of integers, by the number of monomials. If k is selected uniformly at random from integers in the range $[0, 127]$, then the size of such a representation is about 2000 bits on average. Indeed, 151

coefficient of the average size of 6 bits give about 900 bits. Then, the degrees of the monomials are integers from 0 to 150. These take up $(\sum_{k=1}^{7} k\cdot(2^k-2^{k-1}+1))+8\cdot(150-127)\approx 1000$ bits. Thus, it takes about 2000 bits to represent a single tropical polynomial with suggested parameters.

Since a private key is comprised of two such polynomials, this means that the size of the private key in our scheme is about 4000 bits (or 500 bytes) on average.

The public key is a product of two polynomials of degree 150 and is therefore a polynomial of degree 300. Coefficients in this polynomial are in the range $[0, 254]$. Using the same argument as in the previous paragraph, we estimate the size of such polynomial to be about 4500 bits (or 562 bytes) on average.

The signature is a triple of polynomials, two of them are products of 3 tropical polynomials of degree 150 and one of them is a product of 2 tropical polynomials of degree 150. Therefore, the signature size is about 16,000 bits (or 2000 bytes) on average.

In the table below, we have summarized performance data for several parameter sets. Most columns are self-explanatory; the last two columns show memory usage during verification and during the whole process of signing and verification.

Performance metrics for various parameter values							
degree of private polynomials	range for coefficients in private polynomials	verification time (sec)	signature size (Kbytes)	public key size (Kbytes)	private key size (Kbytes)	memory usage, verification (Mbytes)	memory usage, whole process (Mbytes)
100	[0,127]	¡0.1	1.3	0.37	0.33	0.4	0.4
150	[0,127]	0.15	2	0.56	0.5	0.37	0.5
200	[0,127]	0.25	2.6	0.74	0.67	0.47	0.6

References

[1] P. Butkovic, *Max-linear systems: theory and algorithms*, Springer-Verlag London, 2010.

[2] D. Grigoriev, V. Shpilrain, *Tropical cryptography*, Comm. Algebra. **42** (2014), 2624–2632.

[3] D. Grigoriev and V. Shpilrain, *Tropical cryptography II: extensions by homomorphisms*, Comm. Algebra. **47** (2019), 4224–4229.

[4] M. Habeeb, D. Kahrobaei, C. Koupparis, V. Shpilrain, *Public key exchange using semidirect product of (semi)groups*, in: ACNS 2013, Lecture Notes Comp. Sc. **7954** (2013), 475–486.

[5] D. Kahrobaei, V. Shpilrain, *Using semidirect product of (semi)groups in public key cryptography*, in: CiE 2016, Lecture Notes Comp. Sc. **9709** (2016), 132–141.

[6] K. H. Kim, F. W. Roush, *Factorization of polynomials in one variable over the tropical semiring*, `https://arxiv.org/pdf/math/0501167.pdf`

[7] M. Kotov, A. Ushakov, *Analysis of a key exchange protocol based on tropical matrix algebra*, J. Math. Cryptology **12** (2018), 137–141.

[8] Python code for the tropical digital signature scheme, `https://shpilrain.ccny.cuny.edu/tropicalDS.txt`

[9] T. Theobald, *On the frontiers of polynomial computations in tropical geometry*, J. Symbolic Comput. **41** (2006), 1360–1375.

Department of Mathematics, The City College of New York, New York, NY 10031
Email address: jchen056@citymail.cuny.edu

CNRS, Mathématiques, Université de Lille, 59655, Villeneuve d'Ascq, France
Email address: dmitry.grigoryev@math.univ-lille1.fr

Department of Mathematics, The City College of New York, New York, NY 10031
Email address: shpilrain@yahoo.com