UNIVERSITÉ DE RENNES 1

State of the art of tropical cryptography

Arthur Herlédan Le Merdy
Camille Foucault

Tutor : Sylvain Duquesne

1^{st} September 2020 — 3^{th} June 2021

CONTENTS

CHAPTER 1

BACKGROUND

1 Motivation

The objective of this report is to summarily present a recent field of algebra, called *tropical* or *min-plus algebra*, and to introduce its profitable application to cryptography *via* several protocols. These protocols will essentially be enhancements of existing schemes already broken or weakened by linear attacks. Actually, as far as we know, there is no entirely original tropical protocol at the present time.

In this context, the use of tropical algebra as platform for cryptographic schemes has two advantages:

- Security: as tropical algebras are semirings, protocols based on them will be *a priori* less vulnerable to most of already known attacks, in particular linear ones.
- Efficiency: since tropical multiplication (see 2.1) is nothing more than a classical addition, the use of these structure will conduct to a much faster computation.

Therefore, tropical algebra brings an appreciable opportunity to have new approaches on cryptographic schemes, either secure or not, among which some had a true potential. Even if the majority of the protocols presented have been broken, several authors are continuously forging improvements in this area, showing it has been an active research environment since the last ten years.

This presentation will be based on the four protocols presented by D. Grigoriev and V. Shpilrain in [7, 8]. Through the study of these articles, we will be led to work on the different attacks and improvement proposed by various authors. This journey will cover almost the entire literature on this subject at present, to the best of our knowledge.

2 Definitions

2.1 Tropical semiring

We start by introducing essential information on tropical algebra.

Definition (Tropical algebra)**.** Let T be a subset of reals, that contains 0, to which we add a special element we will denote ∞. A tropical algebra $(T, \oplus, \otimes)$ is such a set, closed under addition, endowed with two operations as follow:
$\forall x, y \in T$,

$$\oplus : \begin{cases} x \oplus y = min(x, y) \\ \infty \oplus x = x \oplus \infty = x \end{cases} \qquad \otimes : \begin{cases} x \otimes y = x + y \\ \infty \otimes x = x \otimes \infty = \infty \end{cases}$$

$(T, \oplus, \otimes)$ is called a tropical algebra over T. Afterwards, we will denote by T the tropical algebra if there is no ambiguity.

Remark. This set is also called a min-plus algebra. Note that tropical algebra may be defined as a max-plus algebra as well, where (tropical) addition is defined by $x \oplus y = max(x, y)$.
In what follows we will work on min-plus algebra where T is a subset of the rationals, mainly $\mathbb{Z}$.

Proposition. *T is a commutative semiring.*

Proof. T is closed under $\otimes$ and $\oplus$ because T is closed under addition and $min(x, y) \in T$ $\forall x, y \in T$.
Now we have straightforward the following properties: $\forall x, y, z \in T$,
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:

$$(x \oplus y) \otimes z = (x \otimes z) \oplus (y \otimes z)$$

additive identity:

$$\infty \oplus x = x \oplus \infty = x$$

multiplicative identity:

$$0 \otimes x = x \otimes 0 = x$$ □

Remark. Even if tropical algebra is a relatively new domain of mathematics, it is a very fruitful one, especially tropical geometry. In this report will only be introduced necessary properties for tropical cryptography. We refer the reader to the book [12] if she wants to go deeper on this subject.

2.2 Tropical polynomial

We denote the semiring of tropical univariate polynomials over T in the indeterminate x by $T[x]$. A tropical monomial in $T[x]$ is a linear function in the usual sense and a tropical polynomial is the minimum of a finite number of tropical monomials. Thus a tropical polynomial P in one variable over T is an usual piecewise linear function $P : T \to T$. Moreover, the order of tropical operations follows the classical case.

The tropical polynomials in one indeterminate can be expanded to more variables. We denote by $T[x_1, ..., x_n]$ the semiring of tropical polynomials over T in the indeterminates $(x_i)_{i\in[n]}$ where $[n]$ stands for $\{1, ..., n\}$. For $u = (u_1, ..., u_n) \in \mathbb{N}^n$,

$$x^u = \bigotimes_{i=1}^{n} x_i^{\otimes u_i} := \sum_{i=1}^{n} u_i x_i$$

An element of $T[x_1, ..., x_n]$ is then a polynomial of the form :

$$\bigoplus_{u\in\mathbb{N}^n} c_u \otimes x^u, \text{ where } c_u \in T$$

Notice that a tropical multivariate polynomial restricted to $(T\backslash\{\infty\})^n$ is an usual piecewise linear function $P : T^n \to T$.

Example. *An example of tropical monomial in $T[x, y]$ is $15 \otimes x \otimes x \otimes y$. One may use the alternative notation $x^{\otimes 2}$ for $x \otimes x$, which gives us in this case : $15 \otimes x^{\otimes 2} \otimes y$. We can see that this monomial corresponds to the linear usual function $15 + 2 \times x + y$.*

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

2.3 Tropical rational function

Definition (Semifield)**.** A semifield is a semiring $(S, +, \times)$ in which all nonzero elements have a multiplicative inverse.

Proposition. *The operation $\oplus$ is not invertible. On the other hand, the operation $\otimes$ is invertible, it is simply the usual subtraction. This inverse will be denoted $\oslash$ such that $\forall x, y, z \in T$,*

$$x \oslash y = z \text{ if and only if } y \otimes z = x$$

Remark. Tropical algebras are semifields.

Remark. Let x, y, a be three tropical integers, then we have $y^{\otimes(-a)} = -y^{\otimes a}$ and $x \otimes y^{\otimes(-a)} = x \oslash y^{\otimes a}$.

One can define equivalence classes for the expressions of the form $x \oslash y$, $\forall x, y \in T$ from the equivalence of the above proposition. We will denote by $Rat(T)$ the set of all of them, also called the semifield of fractions of T.

Proposition. *Rat(T) is a semifield equipped by the following operations: $\forall (x \oslash y), (z \oslash t) \in Rat(T)$*

$$(x \oslash y) \otimes (z \oslash t) = (x \otimes z) \oslash (y \otimes t)$$
$$(x \oslash y) \oplus (z \oslash t) = ((x \otimes t) \oplus (y \otimes z)) \oslash (y \otimes t)$$

Now, it will be necessary to expand this construction of semifield of fractions to a tropical polynomial semiring over T.

Definition (Tropical rational function)**.** A tropical rational function is a function that can be given as a quotient of two tropical polynomials such that the denominator is not equal to the infinity polynomial. Thus we can always define a tropical rational function over T in the indeterminates $(x_i)_{i\in[n]}$ by an expression $P \oslash Q$ where P and Q are two tropical polynomials in $T[x_1, ..., x_n]$ such that $Q \neq \infty$.

Remark. The semifield of fractions of a tropical polynomial ring will be exclusively considered over the integers during this report. For that reason we will always denote the semifield of fractions of a tropical polynomial semiring over $\mathbb{Z}$ by $Rat[x_1, ..., x_n]$.

2.4 Automorphisms of $Rat[x_1, ..., x_n]$

An introduction to the automorphism group of $Rat[x_1, ..., x_n]$ is needed to present the public key encryption scheme of D. Grigoriev and V. Shpilrain. The goal of this section is mainly to define the two automorphism types used, the protocol itself will be explained in 3.

Definition (Semiring morphism)**.** Let $(S, +, \times)$ and $(R, +', \times')$ be semirings. A morphism α from S to R is a function $\alpha : S \to R$ such that $\forall x, y \in S$,

$$\alpha(x + y) = \alpha(x) +' \alpha(y)$$
$$\alpha(x \times y) = \alpha(x) \times' \alpha(y)$$
$$\alpha(0_S) = 0_R$$
$$\alpha(1_S) = 1_R$$

Proposition. *A semifield morphism $\alpha : S \to R$ also verifies the following property, by the underlying multiplicative group morphism, for all nonzero element x of S,*

$$\alpha(x^{-1}) = (\alpha(x))^{-1},$$

Let $Aut(Rat[x_1, ..., x_n])$ be the automorphism group of $Rat[x_1, ..., x_n]$, we will simply denote it by $\mathcal{A}ut$ in the future. Let α be an element of $\mathcal{A}ut$, we can see it as a tuple of tropical rational functions $(\alpha(x_1),...,\alpha(x_n))$ and use the properties of semifield morphisms to evaluate it.

Example. *Let α be the automorphism of $Rat[x_1, x_2]$ that permutes x_1 and x_2. We can give α as $\alpha = (x_2, x_1)$. Let $P \in Rat[x_1, x_2]$ be the tropical rational function $P(x_1, x_2) = (x_1 \otimes x_2 \oplus 1) \oslash (x_2^{\otimes 2})$. Then*

$$\begin{aligned}
\alpha(P) &= \alpha((x_1 \otimes x_2 \oplus 1) \oslash (x_2^{\otimes 2})) \\
&= \alpha((x_1 \otimes x_2 \oplus 1) \otimes (x_2^{\otimes 2})^{\otimes -1}) \\
&= \alpha(x_1 \otimes x_2 \oplus 1) \otimes \alpha(x_2^{\otimes -2}) \\
&= (\alpha(x_1) \otimes \alpha(x_2) \oplus \alpha(1)) \otimes (\alpha(x_2))^{\otimes -2} \\
&= (x_2 \otimes x_1 \oplus 1) \otimes (x_1)^{\otimes -2} \\
&= (x_2 \otimes x_1 \oplus 1) \oslash (x_1^{\otimes 2})
\end{aligned}$$

Example. *Let α be the automorphism of $Rat[x_1, x_2, x_3]$ given as the tuple of tropical rational functions $(x_1^{\otimes 2} \oslash x_2, x_3, 2)$. Let $P \in Rat[x_1, x_2, x_3]$ be the tropical rational function $P(x_1, x_2, x_3) = ((x_2 \otimes x_3) \oplus x_1) \oslash x_2$.*

$$\begin{aligned}\alpha(P) &= ((2 \otimes x_3) \oplus (x_1^{\otimes 2} \oslash x_2)) \oslash x_3 \\ &= (((2 \otimes x_3) \oslash 0) \oplus (x_1^{\otimes 2} \oslash x_2)) \oslash x_3 \\ &= (((2 \otimes x_2 \otimes x_3) \oplus (0 \otimes x_1^{\otimes 2}) \oslash x_2) \oslash x_3) \\ &= ((2 \otimes x_2 \otimes x_3) \oplus x_1^{\otimes 2}) \oslash (x_2 \otimes x_3)\end{aligned}$$

2.4.1 Monomial automorphism

Definition (Monomial automorphism)**.** A monomial automorphism $\mu \in \mathcal{A}ut$ is an automorphism given as a tuple of tropical rational functionals $(m_1, ..., m_n)$ where m_i is of the form

$$m_i = b_i \otimes x^{a_i},\ 1 \leq i \leq n,$$

with $x := (x_1, ..., x_n), b_i \in \mathbb{Z}, a_i := (a_{i1}, ..., a_{in}) \in \mathbb{Z}^n$
We choose the exponents such that the matrix $A = (a_{ij})_{i,j\in[n]}$ is invertible in the "classical" sense.

Remark. Since monomial automorphisms are analogs of linear automorphisms in the "classical" situation, we can also have a matricial point of view on $\mu(x)$:

$$\mu(x) = \begin{pmatrix} m_i \\ \vdots \\ m_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} + \begin{pmatrix} a_{11} & ... & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & ... & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

Proposition. *Let $\mu \in \mathcal{A}ut$ be a monomial automorphism, denoted with the same notation as above. We can give its inverse automorphism μ^{-1} by the tuple of tropical rational functions $(w_1, ..., w_n)$ where w_i is of the form*

$$w_i = (x \oslash b)^{c_i},\ 1 \leq i \leq n,$$

where the matrix $C = (c_{ij})_{i,j\in[n]}$ is the inverse matrix, in the "classical" sense, of A and $c_i := (c_{i1}, ..., c_{in})$.

Proof. Let $\mu \in \mathcal{A}ut$ be a monomial automorphism given as the tuple of tropical rational functions $(m_1, ..., m_n)$. Let ω be the automorphism given as the tuple of tropical rational functions $(w_1, ..., w_n)$ defined as above. We can look at the composition $\omega \circ \mu(x)$ with the matricial point of view :

$$\begin{aligned}(\omega \circ \mu)(x) &= \begin{pmatrix} c_{11} & ... & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{n1} & ... & c_{nn} \end{pmatrix} \left(\left(\begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} + \begin{pmatrix} a_{11} & ... & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & ... & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \right) - \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} \right) \\ &= \begin{pmatrix} c_{11} & ... & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{n1} & ... & c_{nn} \end{pmatrix} \begin{pmatrix} a_{11} & ... & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & ... & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \\ &= \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}\end{aligned}$$

The composition $\mu \circ \omega(x)$ is analog.

□

2.4.2 Triangular automorphism

Definition (Triangular automorphism)**.** A triangular automorphism $\tau \in \mathcal{A}ut$ is an automorphism given as a tuple of tropical rational functions $(t_1, ..., t_n)$ where t_i is of the form

$$t_i = x_i \otimes p_i(x_{i+1}, ..., x_n),\ 1 \leq i \leq n, \text{where } p_i \in Rat[x_1, ..., x_n]$$

Proposition. *Let $\tau \in \mathcal{A}ut$ be a triangular automorphism, denoted with the same notation as above. τ is a product of elementary triangular automorphisms $\tau_1 \circ ... \circ \tau_n$ where each τ_i is given as the tuple of tropical rational functions $(x_1, ..., x_{i-1}, t_i, x_{i+1}, ..., x_n)$.*
Its inverse automorphism τ^{-1} is given by the product of the inverse of the elementary triangular automorphisms $\tau_n^{-1} \circ ... \circ \tau_1^{-1}$. The inverse of an elementary triangular automorphism τ_i is given by the tuple of triangular rational functions $(x_1, ..., x_{i-1}, x_i \oslash p_i(x_{i+1}, ..., x_n), x_{i+1}, ..., x_n)$.

Proof. Let $\tau \in \mathcal{A}ut$ be a triangular automorphism given as the tuple of tropical rational functions $(t_1, ..., t_n)$. Let η be the automorphism given by the product of the elementary triangular automorphisms $\tau_n^{-1} \circ ... \circ \tau_1^{-1}$ defined as in the above proposition. Clearly

$$\begin{aligned}\tau \circ \eta &= \tau_1 \circ ... \circ \tau_n \circ \tau_n^{-1} ... \circ \tau_1^{-1} \\ &= Id = \tau_n^{-1} \circ ... \circ \tau_1^{-1} \circ \tau_1 ... \circ \tau_n = \eta \circ \tau\end{aligned}$$

Moreover $\forall \tau_i, 1 \leq i \leq n$, we have

$$\begin{aligned}\tau_i(x_j) &= x_j \quad \forall j \neq i \\ \tau_i(x_i \oslash p_i(x_{i+1}, ..., x_n)) &= (x_i \oslash p_i(x_{i+1}, ..., x_n)) \otimes p_i(x_{i+1}, ..., x_n) = x_i\end{aligned}$$

From here, we can easily deduce that τ_i^{-1} is given as the tuple of tropical rational functions $(x_1, ..., x_{i-1}, x_i \oslash p_i(x_{i+1}, ..., x_n), x_{i+1}, ..., x_n)$.

□

2.5 Tropical matrix semiring

The construction of matrices can now be defined over tropical algebras. As T is a semiring, the set of square matrices $\mathcal{M}_n(T) = \mathrm{Mat}_{n \times n}(T)$ is also a semiring and will have the same interest in cryptography as tropical algebra itself. Therefore, it is the key structure of the protocols covered in this report.

Proposition. *The set $\mathcal{M}_n(T)$ endowed by the following operations is a non-commutative semiring:* $\forall A, B \in \mathcal{M}_n(T), \forall i, j \in [n]$,

$$\begin{aligned}(A \oplus B)_{ij} &= a_{ij} \oplus b_{ij} \\ (A \otimes B)_{ij} &= \bigoplus_{k=1}^{n} a_{ik} \otimes b_{kj}\end{aligned}$$

Note that, with the construction above, the additive and multiplicative tropical identity matrices (respectively $\mathbf{0}$ and $\mathbf{1}$) are given by

$$\begin{aligned}\mathbf{0}_{ij} &= \infty \qquad \forall i, j \in [n] \\ \mathbf{1}_{ij} &= \begin{cases} 0 & \text{if } i = j \\ \infty & \text{otherwise} \end{cases}\end{aligned}$$

The tropical matrix semiring is naturally endowed by the tropical scalar multiplication: let $c \in \mathbb{Z}$,

$$(c \otimes A)_{ij} = c \otimes a_{ij} \qquad \forall i, j \in [n]$$

Notation (Power of a matrix)**.** The power of a tropical matrix follows the standard notation: $\forall k \in \mathbb{N} \backslash \{0\}$,

$$A^{\otimes k} = \underbrace{A \otimes A \otimes ... \otimes A}_{\text{k times}} \quad \text{with} \quad A^0 = \mathbf{1}$$

Using this notation, it is now possible to extend tropical scalar polynomials to tropical matrices.

Definition (Matrix polynomial)**.** Let $P(x) = a_0 \oplus a_1 x \oplus a_2 x^{\otimes 2} \oplus ... \oplus a_d x^{\otimes d}$ be an univariate tropical polynomial as defined in 2.2 and $A \in \mathcal{M}_n(T)$, where d is a positive integer, called the *degree* of P.
P evaluated at a matrix A is

$$P(A) = a_0 \otimes A^0 \oplus a_1 A \oplus a_2 A^{\otimes 2} \oplus ... \oplus a_d A^{\otimes d} = \bigoplus_{i=0}^{d} a_i A^{\otimes i}$$

One can also define a partial order on the set of tropical matrices.

Proposition. *The set $\mathcal{M}_n(T)$ is a partially ordered set (poset) for the partial order given by*

$$A \leq B \quad \textit{iff} \quad A \oplus B = A$$

Proof. Let $A, B, C \in \mathcal{M}_n(T)$. The relation proposed is

- reflexive: $A \oplus A = A$.
- antisymmetric: $A \oplus B = A$ and $B \oplus A = B$ leads to $A = B$.
- transitive: $A \oplus B = A$ and $B \oplus C = B$ leads to $A \oplus C = A \oplus B \oplus C = A \oplus B = A$.

Therefore it is a partial order on $\mathcal{M}_n(T)$ □

3 Remarkable structures on tropical matrices

Most of the structures detailed in the section 2 are based on tropical matrices. The point here is to construct sets or maps of matrices with remarkable cryptographic properties. This can be done with an algebraic approach of the problem which will be introduced here.

3.1 Tropical matrix semigroups

The aim of this section is to define new semigroups of matrices which will be used later as a platform for cryptographic schemes (see 2). In this report, we will omit the algebraic construction using semidirect products for the sake of brevity. Indeed, the final structure is sufficient to understand the cryptographic protocol, but we encourage the reader to consult the corresponding section in [8].

Afterwards we will consider the set $S := \mathcal{M}_n(T) \times \mathcal{M}_n(T)$, where T is a tropical algebra over $\mathbb{Z}$ and $\mathcal{M}_n(T)$ the set of matrices of size n over T. This section starts with the definition of a binary operation on $\mathcal{M}_n(T)$ before to introduce the structure of semigroup on S.

Definition (adjoint multiplication)**.** Let $A, B \in \mathcal{M}_n(T)$. The adjoint multiplication of A and B is

$$A \circ B = A \oplus B \oplus A \otimes B$$

It is a binary operation on the set $\mathcal{M}_n(T)$, as $\oplus$ and $\otimes$ are.

This operation is nothing more than the usual adjoint multiplication $a \circ b = a + b + ab$ in $\mathbb{R}$ adapted to tropical algebra. It is known as an associative operation, which makes it a good candidate as a binary semigroup operation. The interest here is that the adjoint multiplication, which is not distributive in respect to $+$, is distributive in T in respect to $\oplus$.

Remark. The adjoint multiplication is associative and distributive in respect to tropical addition. Indeed, let $A, B, C \in \mathcal{M}_n(T)$.
associativity

$$\begin{aligned}(A \circ B) \circ C &= (A \oplus B \oplus AB) \oplus C \oplus (A \oplus B \oplus A \otimes B) \otimes C \\ &= A \oplus B \oplus A \otimes B \oplus C \oplus A \otimes C \oplus B \otimes C \oplus A \otimes B \otimes C \\ &= A \oplus (B \oplus C \oplus B \otimes C) \oplus A \otimes (B \oplus C \oplus B \otimes C) \\ &= A \circ (B \circ C)\end{aligned}$$

distributivity

$$\begin{aligned}(A \oplus B) \circ C &= (A \oplus B) \oplus C \oplus (A \oplus B) \otimes C \\ &= A \oplus B \oplus C \oplus A \otimes C \oplus B \otimes C \\ &= A \oplus B \oplus (C \oplus C) \oplus A \otimes C \oplus B \otimes C \qquad \textit{because } C \oplus C = C \\ &= (A \oplus C \oplus A \otimes C) \oplus (B \oplus C \oplus B \otimes C) \\ &= (A \circ C) \oplus (B \circ C)\end{aligned}$$

Proposition. *Let $A := (A_1, A_2)$ and $B := (B_1, B_2)$ be two elements of $\mathcal{S}$ and let $\circledcirc : \mathcal{S} \to \mathcal{S}$ be a binary operation defined by*

$$A \circledcirc B = ((A_1 \circ B_2) \oplus B_1, A_2 \circ B_2)$$

Then $(\mathcal{S}, \circledcirc)$ is a semigroup.

Proof. First, $\circledcirc$ is a binary operation on $\mathcal{S}$ as $\circ$ and $\oplus$ are binary operations on $\mathcal{M}_n(T)$. Now, $\circledcirc$ is associative. Indeed, let $A, B, C \in \mathcal{S}$,

$$\begin{aligned}(A \circledcirc B) \circledcirc C &= ((A_1 \circ B_2) \oplus B_1, A_2 \circ B_2) \circledcirc (C_1, C_2) \\ &= ((((A_1 \circ B_2) \oplus B_1) \circ C_2) \oplus C_1, (A_2 \circ B_2) \circ C_2) \\ &= ((((A_1 \circ B_2) \oplus B_1) \circ C_2) \oplus C_1, A_2 \circ (B_2 \circ C_2)) \\ &= ((((A_1 \circ B_2) \circ C_2) \oplus (B_1 \circ C_2)) \oplus C_1, A_2 \circ (B_2 \circ C_2)) \\ &= ((A_1 \circ (B_2 \circ C_2)) \oplus ((B_1 \circ C_2) \oplus C_1), A_2 \circ (B_2 \circ C_2)) \\ &= (A_1, A_2) \circledcirc ((B_1 \circ C_2) \oplus C_1, B_2 \circ C_2) \\ &= A \circledcirc (B \circledcirc C)\end{aligned}$$

Hence $(\mathcal{S}, \circledcirc)$ is a semigroup. □

Remark. In a semigroup there is not necessary an identity element. Therefore, one can define as a left-identity (resp. a right-identity) an element e for which $e \cdot a = a$ (resp. $a \cdot e = a$) for each a in this semigroup. In the semigroup S, there is no left-identity but only a right-identity given by $(\mathbf{0}, \mathbf{0})$.

3.2 Jones matrices

In [13], D.L. Jones introduced a new class of matrices, which will be afterwards considered by A. Muanalifah and S. Sergeev [18] as an interesting class of commuting matrices for cryptographic schemes. In particular, Jones matrices are the basis of rich constructions we will introduce below. In this section, T will stand for the tropical (min-plus) algebra on the set of rationals.

3.2.1 Definitions

Definition (Jones matrix)**.** Let A be a matrix in $\mathcal{M}_n(T)$. A is a Jones matrix over T if

$$a_{ij} \otimes a_{jk} \geq a_{ik} \otimes a_{jj} \qquad \forall i, j, k \in [n]$$

We will denote by $\mathcal{J}_n(T)$ the set of Jones matrices of size n over T.

Definition (Deformation)**.** Let $A \in \mathcal{J}_n(T)$ and $\alpha \in [0, 1]$ a rational number. A deformation is a map on the tropical matrices semiring defined as follow:

$$\begin{array}{rccc} \mathcal{D}_\alpha & : & \mathcal{J}_n(T) & \to & \mathcal{J}_n(T) \\ & & A & \mapsto & A^{(\alpha)} \end{array}$$

where

$$A^{(\alpha)} = (a_{ij} \otimes (a_{ii} \oplus a_{jj})^{\otimes(\alpha-1)})_{i,j\in[n]}$$

Remark. $A^{(\alpha)}$ a Jones matrix. Therefore, the set $\mathcal{J}_n(T)$ of Jones matrices is closed under deformation.

Proposition (Commutativity)**.** *Let $A, B \in \mathcal{J}_n(T)$ and $0 \leq \alpha, \beta \leq 1$. Then $A^{(\alpha)} \otimes A^{(\beta)} = A^{(\beta)} \otimes A^{(\alpha)}$*

The proof of the commutativity is given in [18] for max-plus algebra in which Jones matrices follow the inequalities $a_{ij} \otimes a_{jk} \leq a_{ik} \otimes a_{jj}$. This proof can be naturally adapted to min-plus algebras by reversing the inequalities, as defined above.

3.2.2 Quasi-polynomials

The notion of polynomials over $\mathcal{M}_n(T)$ has already been adapted in 2.2. Now, to define a polynomial over the class of Jones matrices, one will use the notion of *deformation*, introduced above, which has the remarkable property to preserve this structure and therefore the commutativity.

Definition (Quasi-polynomial)**.** Let $(\alpha_i)_{i\in[d]}$ be a finite subset of rationals with $0 \leq \alpha_i \leq 1$ for all $i \in [d]$. Let $(a_i)_{i\in[d]}$ be a finite set of tropical numbers. A quasi-polynomial Q in the indeterminate x is given by

$$Q(x) = \bigoplus_{i=1}^{d} a_i \otimes x^{(\alpha_i)}$$

Example. *$Q(x) = x^{(1/2)} \oplus (-1) \otimes x^{(2/3)} \oplus 2 \otimes x$ is a quasi-polynomial – one may notice that $x = x^{(1)}$. Now let A be the following Jones matrix:*

$$A = \begin{pmatrix} 2 & 3 \\ 4 & -10 \end{pmatrix}$$

We can evaluate Q at the matrix A to obtain

$$Q(A) = \begin{pmatrix} 1 & 8 \\ 9 & -5 \end{pmatrix} \oplus (-1) \otimes \begin{pmatrix} 4/3 & 19/3 \\ 22/3 & -20/3 \end{pmatrix} \oplus 2 \otimes \begin{pmatrix} 2 & 3 \\ 4 & -10 \end{pmatrix} = \begin{pmatrix} 1/3 & 5 \\ 6 & -8 \end{pmatrix}$$

Remark. Deformations and quasi-polynomials can be defined more widely with $\alpha \leq 1$ a real. In this report we decided to restrain the definitions to what is necessary for the protocol. We refer the reader to [18] for a more detailed study.

3.2.3 Generation of Jones matrices

Beyond the theory, the necessity to generate Jones matrices has been a strong issue in our work. One may notice that it is sufficient to minimize the coefficients of the diagonal to construct a matrix which satisfies the definition. Nevertheless, in order to implement good cryptographic schemes, the public matrices (see 1.4.1) have to be chosen as random as possible. In this report we will propose an efficient way to generate them.

The first step to generate them is the Answer Set Programming (ASP). In classical programming, one tries to compute a solution to a given problem by describing a sequence of steps to execute (*i.e.* "how to solve it"). On the other hand, declarative programming – which ASP belongs to – consists in modeling the problem and let a solver find a solution (*i.e.* "what is the problem?"). Thus, ASP is particularly suited to solve combinatorial problems, by conducting an exhaustive search in the search space, that may be restricted with constraints, what traditional programming could not handle.

Using this language, we implemented a function *jones_ asp()* [10] to generate a Jones matrix with small coefficients in a reasonable time. It is then easy to be reduced to the initial parameters using the following proposition.

Proposition. *Let $A \in \mathcal{J}_n(T)$ be a Jones matrix and $C, D \in \mathcal{M}_n(T)$ be random diagonal matrices. Then $C \otimes A \otimes D$ is also a Jones matrix.*

The proof is given in [18] for max-plus algebra but can easily be adapted to min-plus algebra.

Proof. Let $A \in \mathcal{J}_n(T)$, $C = (c_1, ..., c_n)$ and $D = (d_1, ..., d_n)$ with $(c_i)_{i\in[n]}, (d_i)_{i\in[n]} \in T^n$. Then

$$\begin{aligned} a_{ij} \otimes a_{jk} &\geq a_{ik} \otimes a_{jj} \\ (c_i \otimes c_j \otimes d_j \otimes d_k) \otimes a_{ij} \otimes a_{jk} &\geq (c_i \otimes c_j \otimes d_j \otimes d_k) \otimes a_{ik} \otimes a_{jj} \\ (c_i \otimes a_{ij} \otimes d_j) \otimes (c_j \otimes a_{jk} \otimes d_k) &\geq (c_i \otimes a_{ik} \otimes d_k) \otimes (c_j \otimes a_{jj} \otimes d_j) \\ (C \otimes A \otimes D)_{ij} \otimes (C \otimes A \otimes D)_{jk} &\geq (C \otimes A \otimes D)_{ik} \otimes (C \otimes A \otimes D)_{jj} \end{aligned}$$

□

To generate a Jones matrix of a certain size, it is now sufficient to generate a Jones matrix with small coefficients – easily doable with ASP – and to apply diagonal matrices with greater coefficients. This is how we do in practice with the corresponding protocol as explained in section 1.4.1. Note that one saves the use of ASP only for solving the combinatorial core of the problems. Here it is more relevant to perform the adjustment of coefficients with imperative programming since it is not a combinatorial problem.

3.3 Linde-De La Puente matrices

In [16], J. Linde and M.J. de la Puente proposed an analysis of a new class of commuting matrices over the max-plus algebra semiring, called *normal* matrices. In [18], this notion has been extended to a new class of commuting matrices, said *Linde-De La Puente*. In this report we will adapt it to the min-plus matrices semiring $\mathcal{M}_n(T)$ as defined in 2.5.

Definition (Linde-De La Puente matrix)**.** Let $A \in \mathcal{M}_n(T)$. A is said *Linde-De La Puente* if for arbitrary rational numbers $r \geq 0$ and $k \leq 0$,

$$\begin{aligned} a_{ii} &= k \qquad \forall i \in [n] \\ a_{ij} &\in [r, 2r] \quad \forall i, j \in [n], i \neq j \end{aligned}$$

The set of such matrices is denoted by $[r, 2r]_n^k$. More widely, the set of *Linde-De La Puente* matrices of size n over T is denoted by $\mathcal{L}_n(T)$.

Remark. Here, $2r$ has to be understood as $2 \times r$ for the standard multiplication. Moreover, $[r, 2r]$ follows the usual interval notation and stands for the set of rational numbers that contains all rational numbers lying between r and $2r$ for the partial order in the standard case.

Proposition. $(\mathcal{L}_n(T), \oplus, \otimes)$ *is a commutative semiring.*

Proof (commutativity). We refer the reader to [18] where the whole proof is given for $k \geq 0$ a and $r \leq 0$ over a max-plus algebra. We provide here a proof of the commutativity over the min-plus algebra which is very similar. Let $A \in [r, 2r]_n^{k_1}$, $B \in [s, 2s]_n^{k_2}$ for any $r, s \geq 0$, $k_1, k_2 \leq 0$. For all i, j,

$$\begin{aligned} (A \otimes B)_{ij} &= a_{ii} \otimes b_{ij} \oplus a_{ij} \otimes b_{jj} \bigoplus_{l \notin \{i,j\}} a_{il} \otimes b_{lj} \\ &= k_1 \otimes b_{ij} \oplus k_2 \otimes a_{ij} \bigoplus_{l \notin \{i,j\}} a_{il} \otimes b_{lj} \\ &= k_1 \otimes b_{ij} \oplus a_{ij} \otimes k_2 \qquad (1.1) \\ &= (k_2 \otimes A \oplus k_1 \otimes B)_{ij} \\ &= (B \otimes A)_{ij} \end{aligned}$$

To achieve the proof, it remains to prove (1.1) $a_{il} \otimes b_{lj} \leq k_1 \otimes b_{ij} \oplus k_2 \otimes a_{ij}, \forall l \notin \{i, j\}, \forall i, j \in [n]$, *i.e.* for all off-diagonal elements. Indeed,

$$a_{il} \otimes a_{lj} \geq r \otimes s \geq 2r \oplus 2s \geq a_{ij} \oplus b_{ij} \geq k_1 \otimes b_{ij} \oplus k_2 \otimes a_{ij}$$

□

The commutativity of *Linde-De La Puente* matrices is the basis of an interesting public key exchange protocol detailed in section 1.4.2.

CHAPTER 2

THREE PROTOCOLS OF TROPICAL CRYPTOGRAPHY

1 Stickel over the tropical matrix semiring

In 1976 [4], W. Diffie and M. Hellman founded the *Public key exchange* cryptography with a new protocol over a commutative finite group based on the *Discrete logarithm problem* (DLP). This protocol is rather secure since there is no known efficient algorithm which solves the DLP yet. Nevertheless, according to E. Stickel [26], *"the permanent increase in computational power requires permanent adjustment of the public-key parameters. Therefore, it is desirable to look for techniques in more complex algebraic structures that possibly require lower key sizes."*

In his article, E. Stickel proposed a new theoretical protocol based on non-commutative groups and suggested the group of invertible matrices over a finite field as platform. It has unfortunately been victim of an efficient attack by V. Shpilrain (see [24]), using the linearity of this structure. Then, D. Grigoriev and V. Shpilrain proposed three new protocols over tropical algebra to build schemes with fast computation and less vulnerable to linear attacks, as introduced as the motivation (1) of this report.

One of these, presented in [7], is based on the Stickel's key exchange protocol. The section is about this scheme and all its variants and attacks, to the best of our knowledge.

1.1 Stickel's general protocol

Before addressing the tropical variant, we need to introduce Stickel's protocol as described in [26]. Here, E. Stickel proposes a new general protocol over a non-abelian finite group, and gives an example of such a group which could – according to him – be a good support for his protocol. Here is the theoretical scheme:

Protocol 1. ***Stickel***

1. *Alice and Bob agree on an non-abelian finite group G and $g, h \in G$, $gh \neq hg$, of order $|g|, |h|$ respectively.*
2. *Alice chooses two positive integers $a_1 < |g|, a_2 < |h|$, and forms $u = g^{a_1}h^{a_2}$. She sends u to Bob.*
3. *Bob chooses two positive integers $b_1 < |g|, b_2 < |h|$, and forms $v = g^{b_1}h^{b_2}$. He sends v to Alice.*
4. *Alice computes $K_a = g^{a_1}vh^{a_2}$ and Bob computes $K_b = g^{b_1}uh^{b_2}$. $K = K_a = K_b$ is the private shared key.*

Recall that the main aim of Stickel is to enhance the security in order to take smaller – or faster to generate – keys for the same security. In his article, Stickel proposed a first platform for his scheme, namely the group of invertible matrices over a finite field. This will not be detailed here, we refer the reader to our implementation [10] for more details. What is important to notice is that it has suffered an

efficient attack by V. Shpilrain in [24], using the linearity of matrices. Our implementation of this attack can be found in [10] but will not be detailed here since it is outside the scope of our study.

This attack comes from the fact that even if Stickel's scheme is very reminiscent of Diffie-Hellman it is not based on the DLP anymore. Indeed V. Shpilrain showed in [24] that obtaining the secret key in Stickel's scheme is not harder than solving the *decomposition search problem* (DSP) restricted to semigroups. Thus the crucial ingredient is the choice of the platform (semi)group. For V. Shpilrain and A. Ushakov in [25], the platform (semi)group needs to be at least non-commutative. Moreover, V. Shpilrain suggested in [24] to use a semigroup with a lot of non-invertible elements instead of a group. Therefore, D. Grigoriev and V. Shpilrain adapted the Stickel's scheme to a tropical protocol in [7] which we will cover in 1.2.

Note that we implemented the theoretical protocol using our own platform. The construction of this non-commutative – with non-trivial center – group is available in [10]. This is only a toy example which does not correspond to any existing protocol and does not aim at describing a secured and efficient cryptosystem.

Definition (Decomposition search problem (DSP))**.** Let G be a semigroup. Given $u, w \in G$ two elements and $A, B \subseteq G$ two subsemigroups. To find $x \in A$ an $y \in B$ such that $u = xwy$ is called the (subsemigroup-restricted)*Decomposition search problem.*

The reduction to this problem for the Stickel's scheme case implies that an eavesdropper only has to find two elements x, y of the platform (semi)group G verifying the following conditions :

$$\begin{cases} x \otimes g = g \otimes x \\ y \otimes h = h \otimes y \\ x \otimes y = K_a \text{ (resp. } K_b) \end{cases}$$

to retrieve the shared key. It will be detailed in the tropical case at 1.3.

Remark. In [24], V. Shpilrain reminds *"There are several key exchange protocols that directly use the alleged hardness of the DSP in various (semi)group"*. Moreover, there are still searches for a (semi)group providing a secure platform for protocols based on DSP.

1.2 Tropical case

The tropical case follows the same steps as the standard Stickel's protocol, except that polynomials are used instead of simple powers.

Protocol 2. ***Stickel over tropical algebra***

1. *Alice and Bob agree on public tropical matrices A, B.*
2. *Alice chooses two tropical polynomials P_1, P_2 and sends $U = P_1(A) \otimes P_2(B)$ to Bob.*
3. *Bob chooses two tropical polynomials Q_1, Q_2 and sends $V = Q_1(A) \otimes Q_2(B)$ to Alice.*
4. *Alice computes $K_a = P_1(A) \otimes V \otimes P_2(B)$ and Bob computes $K_b = Q_1(A) \otimes U \otimes Q_2(B)$. $K = K_a = K_b$ is the shared secret key.*

The final equality is based on the associativity of the tropical polynomials and their commutativity when evaluated on the same matrix. Indeed,

$$\begin{aligned} K_a &= P_1(A) \otimes (Q_1(A) \otimes Q_2(B)) \otimes P_2(B) \\ &= (P_1(A) \otimes Q_1(A)) \otimes (Q_2(A) \otimes P_2(B)) \\ &= (Q_1(A) \otimes P_1(A)) \otimes (P_2(B) \otimes Q_2(B)) \\ &= Q_1(A) \otimes (P_1(A) \otimes P_2(B)) \otimes Q_2(B) = K_b \end{aligned}$$

The parameters suggested by D. Grigoriev and V. Shpilrain are the following:

- Size of the matrices: 10×10
- Entries of the matrices: $[-10^{10}, 10^{10}]$
- Degree of the polynomials: in $[1, 10]$
- Coefficients of the polynomials: $[-10^3, 10^3]$

These parameters lead to $(2 \times 10^{10} + 1)^{10^2}$ possibilities for each public matrix, and $(2 \times 10^3 + 1)^{10}$ for each polynomial. That is $((2 \times 10^3 + 1)^{10^2})^2 \times ((2 \times 10^{10} + 1)^{10})^4 \simeq 7,282$ bits. This is far more than the standard protocols, for a faster computation.

Indeed, faster computation is one of the advantage of the tropical version of the protocol over the classical version. Obviously this higher speed comes from the fact that one does not have to compute usual multiplications as often as in the classical case. One performs an usual multiplication in a tropical algorithm only when one would have to do an usual exponentiation in the classical case. Most of the time, one only computes usual additions and minima between numbers. In general, this observation leads to much faster algorithms in tropical.

The tropical version has also security advantages compared to the classical one. In [24], V. Shpilrain reminds us that retrieving the shared secret K from the suggested version of the classical Stickel's scheme can be done by resolving a system of linear equations (through the DSP as explained in 1.1). In the tropical case, this reduction translates into the following proposition.

Proposition. *An eavesdropper who wants to get the shared secret K of Alice and Bob, knowing A, B, U and V, only needs to find a pair of matrices X, Y satisfying the following conditions :*

$$\begin{cases} X \otimes A = A \otimes X \\ Y \otimes B = B \otimes Y \\ X \otimes Y = U \text{ (resp. } V) \end{cases}$$

then she gets the shared secret $K = X \otimes V \otimes Y$ (resp. $K = X \otimes U \otimes Y$).

Proof. Indeed, if a matrix X commutes with A, it will commute with any power of A or multiple of A thus it will commute with $P(A)$ for any tropical polynomial P. Of course the same properties apply for the matrix Y which will commute with $P(B)$ for any tropical polynomial P.

Then, as $V = Q_1(A) \otimes Q_2(B)$ (resp. $U = P_1(A) \otimes P_2(B)$) and with the above properties,

$$\begin{aligned} X \otimes V \otimes Y &= X \otimes Q_1(A) \otimes Q_2(B) \otimes Y \\ &= Q_1(A) \otimes X \otimes Y \otimes Q_2(B) \\ &= Q_1(A) \otimes U \otimes Q_2(B) \\ &= K \end{aligned}$$

The respective equality for $X \otimes U \otimes Y$ is analog. □

For D. Grigoriev and V. Shpilrain in [7], this is *a priori* not possible to use a linear attack to find efficiently such X, Y in the tropical case:

- Matrices are generally not invertible so the equation $XY = U$ with known U and unknown X, Y does not translate into a system of linear equations.
- The equations $AX = XA$ and $BY = YB$ do translate into a system of linear equations, called "two-sided min-linear system". This kind of system is well-known but currently there is no efficient method to solve it. We refer the reader to [7] for more information.

However, regardless of this security improvement, this protocol is vulnerable to really effective attacks and is no longer a candidate for real application in cryptography, at least in this form.

1.3 Attacks

Two attacks were proposed in [15] by M. Kotov and A. Ushakov and will be introduced in this section.

In the tropical case, one can notice an important weakness of Stickel's scheme. D. Grigoriev and V. Shpilrain suggested to take the coefficients of polynomials P_1, P_2, Q_1, Q_2 in the range $[-10^3, 10^3]$ and to take the entries of the matrices A, B in the range $[-10^{10}, 10^{10}]$. This suggestion directly leads to generate matrices and polynomials with a predictable behavior.

Indeed, tropical matrix with negative entries have a high probability to become entirely negative and to decrease linearly with the power. If polynomial's coefficients are small enough then the leading monomial will directly gives the evaluation, *i.e.* for a tropical matrix A and a tropical polynomial P of tropical degree d with the suggested parameters, we will often have :

$$P(A) = \bigoplus_{i=0}^{d} x_i \otimes A^{\otimes i} = x_d \otimes A^{\otimes d}$$

From this observation and the above proposition, M. Kotov and A. Ushakov proposed a relatively efficient heuristic for the tropical Stickel's scheme.

1.3.1 Heuristic

The heuristic introduced by M. Kotov and A. Ushakov in [15] aims at finding matrices X, Y of the form $X = c \otimes A^{\otimes i}$ and $Y = B^{\otimes j}$, $c \in \mathbb{Z}, i, j \in [1, D]$ verifying the sufficient conditions presented above to have $K = X \otimes U \otimes Y$. We denote by D the upper bound for degrees of polynomials in the protocol, $D = 10$ in the tropical Stickel's scheme proposed in [7]. The existence of such matrices is habitual because of the exposed predictable behavior of the protocol.

Algorithm 1 Heuristic on tropical Stickel's scheme

```
1:  procedure HEURISTIC(A,B,U,V,D)
2:      i ← 1
3:      while i ≤ D do
4:          j ← 1
5:          while j ≤ D do
6:              T_{i,j} ← A^{⊗i} ⊗ B^{⊗j}
7:              if T_{i,j} has only finite entries then
8:                  T_{i,j} ← U − T_{i,j}                ▷ It is an usual matrix subtraction
9:                  if T_{i,j} has all its entries equal then
10:                     X ← T_{i,j}[0,0] ⊗ A^{⊗i}        ▷ T_{i,j}[0,0] designates the first entry of T_{i,j}
11:                     Y ← B^{⊗j}
12:                     return X ⊗ V ⊗ Y
13:             j ← j + 1
14:         i ← i + 1
15:     return FAIL
```

Our implementation of this algorithm is available at [10]. Now, have a look at its efficiency depending on the different parameters suggested in [15].

Range for coefficients of polynomials	$[-10^3, 10^3]$	$[-10^{10}, 10^{10}]$	$[0, 10^{10}]$
Range for entries of matrices	$[-10^{10}, 10^{10}]$	$[-10^{10}, 10^{10}]$	$[0, 10^{10}]$
Our average time (sec)	0.09	0.12	0.20
Our success rate	97.1 %	71.9 %	1.6 %

Table 1: Average running time and success rate of the heuristic attack on the protocol 2 depending on the coefficients of the polynomials and the matrices (based on 1000 runs).

We get appreciably the same success rate than M. Kotov and A. Ushakov in [15]. Clearly, the presence of negative entries of matrices is a weakness exploitable by this heuristic. Nevertheless, even if the average time of execution is always relatively low, the heuristic becomes inefficient when we focus on positive entries for the matrices. We can conclude that simple changes on the suggested parameters given by D. Grigoriev and V. Shpilrain are enough to counter this attack. Notwithstanding, in [15] they also developed a more general attack, with a perfect rate of success based on the Simplex algorithm.

1.3.2 General attack based on the Simplex algorithm

The Simplex algorithm is a well-known algorithm in linear optimization. It aims at maximizing or minimizing complex systems of equations and therefore is particularly well adapted to tropical algebra. Even

if it does not provide a solution in polynomial time, it is an efficient and widely used algorithm in practice. All the useful information about this algorithm can be found in [2].

Suppose than an eavesdropper intercepted the matrix U sent by Alice to Bob. To break the protocol, she needs to find $(x_i)_{i\in\{0,..,D\}}, (y_i)_{i\in\{0,..,D\}}$ in $\mathbb{Z}^{D+1}$ such that

$$\bigoplus_{i,j=0}^{D} x_i \otimes y_j \otimes A^i \otimes B^j = U$$

with A, B the public matrices, D the maximum degree of the polynomial. This leads to the system

$$\min_{i,j\in\{0,\dots,D\}} (x_i + y_j + (A^i + B^j - U)_{kl}) = 0 \quad \forall k, l \in [n]$$

This is exactly the kind of system willing to be solved by the Simplex algorithm. Note however that to perform this attack one needs to find an additive inverse to U, which is not possible if U has infinite entries. To have such a matrix, the matrices A^i and B^j must have many infinite coefficients, and this is even more true for A and B. Therefore this case is not interesting for the moment. We can imagine that further researches could lead to a system exploiting this flaw of the Simplex but it has not be done for now.

It is not detailed here but we propose an implementation of the attack in *Sagemath* – the original implementation in [15] being in *Gap*. Both can be found in [10]. We present below a summary table of the average execution time of our program as a function of the coefficients of the polynomials:

Range for coefficients of polynomials	$[-10^3, 10^3]$	$[-10^{10}, 10^{10}]$	$[0, 10^{10}]$
Range for entries of matrices	$[-10^{10}, 10^{10}]$	$[-10^{10}, 10^{10}]$	$[0, 10^{10}]$
Our average time (sec)	3.31	3.58	4.11
Our success rate	100 %	100 %	100 %

Table 2: Average running time and success rate of the Simplex attack on protocol 2 depending on the coefficients of the polynomials and the matrices (based on 100 runs).

One can remark that the range of the coefficients does not affect significantly the running time of the attack. In fact, taking non-negative values produce single cases which requires more than one loops turn but the running time of a loop remains the same. The size of the matrix does not increase significantly the running time at all, which is around 4.2 sec for matrices of size 20×20. Indeed, what is decisive with the Simplex algorithm is the number of equations studied, *i.e.* the number of constraints. In practice, this can be increased by the degree of the polynomials. This is what A. Muanalifah and S. Sergeev studied in [18], providing the figure 1.

1.4 Considering new classes of commuting matrices

The protocol of key exchange presented in [7] is based on the natural commutativity of $P(A)$ and $Q(A)$ for A a tropical matrix and P, Q tropical polynomials. In this section, we will address more specific classes of commuting matrices in order to avoid the use of classical polynomials. The entirety of the following comes from the work of A. Muanalifah and S. Sergeev in [18] where they proposed two variants of the protocol, a first one based on the thesis of Jones [13], and a second on an article of J. Linde and M.J. De La Puente [16].

1.4.1 Jones matrices

The construction of Jones matrices has already been introduced in 3.2.3. Here we will present the protocol based on them and discuss its advantages and drawbacks. First of all, the steps of the scheme:

Protocol 3. ***Jones matrices.***

1. *Alice and Bob agree on public Jones matrices A, B and an arbitrary tropical matrix W.*
2. *Alice chooses two quasi-polynomials P_1, P_2 and sends $U = P_1(A) \otimes W \otimes P_2(B)$ to Bob*
3. *Bob chooses two quasi-polynomials Q_1, Q_2 and sends $V = Q_1(A) \otimes W \otimes Q_2(B)$ to Alice*
4. *Alice computes $K_a = P_1(A) \otimes V \otimes P_2(B)$ and Bob computes $K_b = Q_1(A) \otimes U \otimes Q_2(B)$. $K = K_a = K_b$ is the shared secret key.*

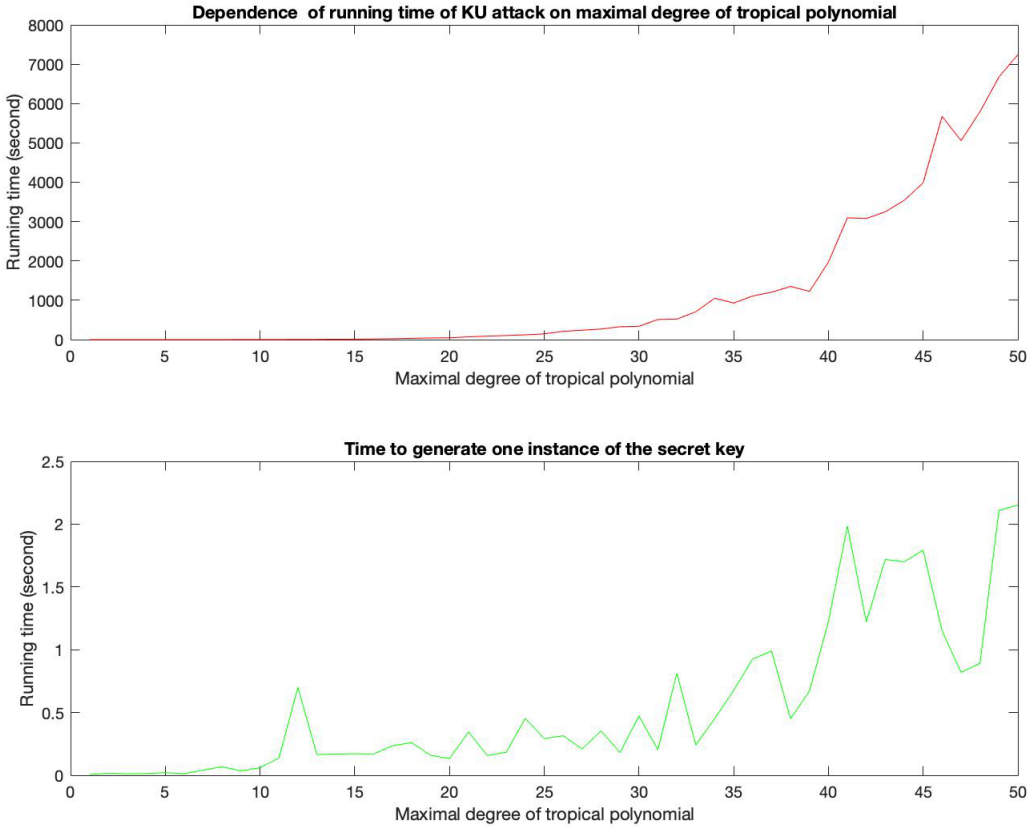

Figure 1: Top: Dependence of running time of the Simplex attack on the maximal degree of tropical polynomials. Bottom: Running time for generating K_a or K_b in Protocol 2 (figures from [18])

One should notice that this protocol is close to the original one presented by D. Grigoriev and V. Shpilrain. The main difference here is the use of quasi-polynomials instead of classical ones. Indeed, as mentioned above, classical polynomials have the big disadvantage of relying on powers of tropical matrices and thus decreasing sequences. Therefore, by using quasi-polynomials, which have no obvious pattern, the authors avoid the heuristic attack proposed by M. Kotov and A. Ushakov.

Moreover, the authors introduced an arbitrary tropical public matrix W in their protocol. This additional may seem anecdotal but is really important. Indeed, the use of particular structures like Jones matrices or Linde-De La Puente matrices gives the possibility to an eavesdropper to use their properties to find another angle of attack on the scheme. The aim of the matrix W is to break these structures to exchange a tropical matrix with less properties as possible.

A. Muanalifah and S. Sergeev presented this protocol in their article [18], without recommending any parameters. However, they produced an analysis of the running time in function of the maximal denominator of the tropical quasi-polynomial. Therefore, we will choose the maximum value of the graph, which is 50, and correspond to a running time of 2.5 sec which is reasonable. For the remaining, we will keep the parameters recommended by D. Grigoriev and V. Shpilrain.

Recommended parameters for this protocol:

- Size of the matrices: 10×10
- Entries of the matrices: $[-10^{10}, 10^{10}]$
- Degree of the polynomials: in $[1, 10]$
- Range of the coefficients: $[-10^{10}, 10^{10}]$

- Maximum denominator of tropical quasi-polynomials: 50

This protocol is more expensive than the original one presented by D. Grigoriev and V. Shpilrain. However, the most expensive step is the generation of the public Jones matrices. In this case, it is fair to consider that a public matrix can be used multiple times and therefore the running time of the protocol can be reduced to the computation of private and secret keys, which is negligible. In conclusion, the practical average time is around 20 ms which is clearly reasonable in practice.

Concerning the Jones matrices, as detailed in 3.2.3 we have to use different advanced tools to generate them. We proceed this way:

Algorithm 2 Generate Jones matrix

1: **procedure** GENERATEJONESMATRIX(n, min, max)
2: $J \leftarrow$ JONESASP(n)
3: $A, B \leftarrow$ diagonal tropical matrices of size n with entries in $[min/2, max/2]$
4: **return** $A \otimes J \otimes B$

The step JONESASP(n) generates a matrix of size n with entries between $-\frac{n^2}{2}$ and $\frac{n^2}{2}$ using Answer Set Programming. It only depends on the size of the matrix. It is the most expensive part of the algorithm, and thus the one that can potentially be improved in future research. The role of the diagonal matrices is to extend the values of the final output to $[min, max]$. We refer the reader to [10] for the full implementation.

1.4.2 Linde-De La Puente matrices

As shown in 3.3, the set of Linde-De La Puente matrices is commutative. It is therefore a good candidate to build a public key exchange scheme. This protocol, which is again a variant of the Protocol 2, is summarized as follow:

Protocol 4. ***Linde-De La Puente matrices***

1. *Alice and Bob agree on an arbitrary tropical matrix W.*
2. *Alice chooses $A_1 \in [a, 2a]_n^{k_1}$ and $A_2 \in [b, 2b]_n^{k_2}$ with random $a, b \geq 0$ and $k_1, k_2 \leq 0$ and sends $U = A_1 \otimes W \otimes A_2$ to Bob.*
3. *Bob chooses $B_1 \in [c, 2c]_n^{l_1}$ and $B_2 \in [d, 2d]_n^{l_2}$ with random $c, d \geq 0$ and $l_1, l_2 \leq 0$ and sends $V = B_1 \otimes W \otimes B_2$ to Alice.*
4. *Alice computes $K_a = A_1 \otimes V \otimes A_2$ and Bob computes $K_b = B_1 \otimes U \otimes B_2$. $K = K_a = K_b$ is the shared secret key.*

A. Muanalifah and S. Sergeev proposed a variant of the protocol above which exploits the commutativity of A and B for A a Linde-De La Puente matrix $[a, 2a]_n^k$ and B with entries in $[0, k]$. This protocol will not be detailed here since it is very close from the first one, but its implementation is available in [10].

Once again, we will keep the parameters proposed by D. Grigoriev and V. Shpilrain:

- Size of the matrices: 10×10
- Entries of the matrices: $[-10^{10}, 10^{10}]$

The running time of the protocol is summarized in the following array. The time is given for entries between -10^{10} and 10^{10} but does not change significantly if we modify this range.

Size of the matrices	10	20	30	50
Average time	2.26 ms	35.6 ms	145 ms	1.27 sec

Table 3: Average running time of the Protocol 4 with entries in $[-10^{10}, 10^{10}]$ depending on the size of the matrices (based on 100 runs).

The results for the variant are similar.

1.4.3 Attacks

First of all, as proved by A. Muanalifah and S. Sergeev in [18], the attack of M. Kotov and A. Ushakov on the Protocol 3 using the Simplex algorithm actually works on the variants presented above. It requires some modifications to adapt it, but a system of inequations using the minimum is still feasible. Moreover, even if the Kotov-Ushakov attack works, the running time is much higher than their original attack on the Protocol 2.

Protocol 3

A. Muanalifah and S. Sergeev proposed in [18] an attack on their own protocol using an adaptation of the Kotov-Ushakov attack. They summarized their results in the Figure 2.

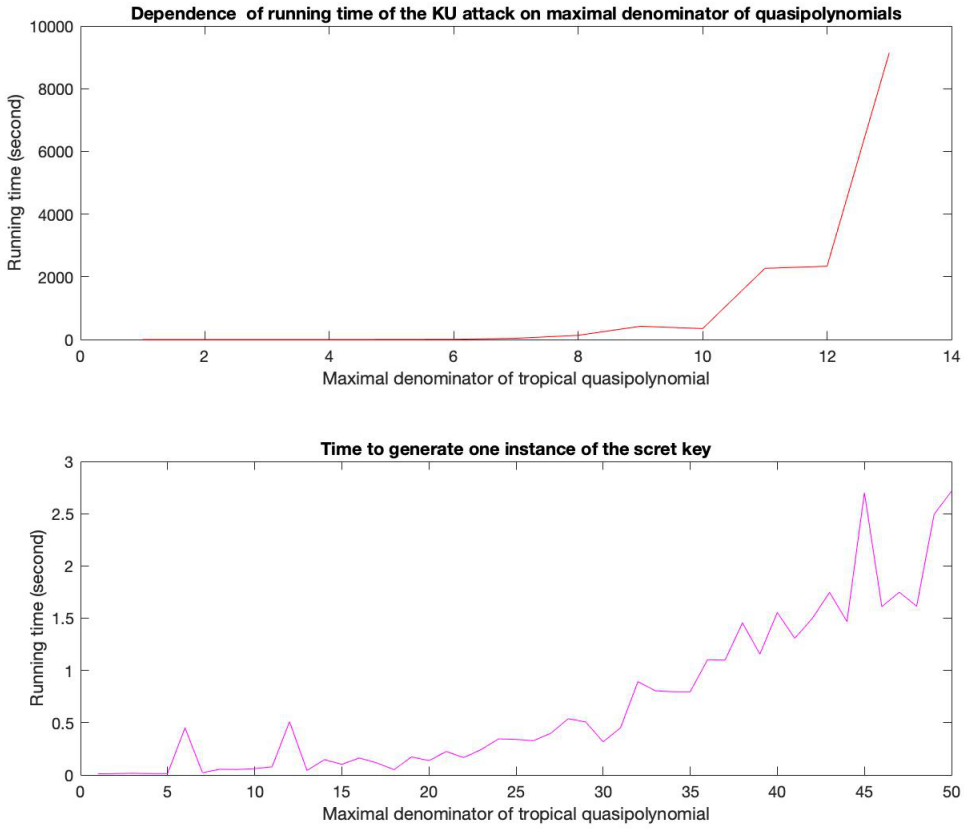

Figure 2: Top: Dependence of running time of the modified Simplex attack on the maximal degree of quasi-polynomials. Bottom: Running time for generating K_a or K_b in Protocol 3 (figures from [18])

The time spent to generate one instance of the secret key in the Protocol 3 seems to grow linearly with the denominator whereas the running time of the attack seems to grow exponentially with the denominator. Thus, this attack is not efficient enough to consider the protocol as broken. According to A. Muanalifah and S. Sergeev, it is therefore interesting to search for more efficient attacks on these protocols.

Protocol 4

It is still possible to find new efficient heuristic attacks on the Protocol 4. These attacks are based on the properties on the matrix W which is the key of the security of the protocol. A. Muanalifah and S. Sergeev highlighted two cases in which a bad choice of W could make the scheme vulnerable to efficient attacks. The main problem is that these "bad choices" depend of all the parameters chosen: W but also A_1, A_2, B_1 and B_2. Therefore, it is impossible to avoid them only by an universal "good choice" of W,

which makes these cases hard to prevent. However, as detailed below, if its entries are chosen too small, W is likely to be vanishing, and if its entries are chosen too large it is likely to be dominant.

Definition (Vanishing)**.** $W \in \mathcal{M}_n(T)$ is said vanishing for $(A_1, A_2), (B_1, B_2) \in \mathcal{M}_n(T) \times \mathcal{M}_n(T)$ if $A_1 \otimes W \otimes A_2 = A_1 \otimes A_2$ and $B_1 \otimes W \otimes B_2 = B_1 \otimes B_2$.

When W is vanishing, the secret key K can be found by the simple expression

$$k_{ij} = v_{st} \otimes w_{st}^{\otimes -1} \otimes U \oplus u_{st} \otimes w_{st}^{\otimes -1} \otimes V$$

for U, V the keys sent respectively by Alice and Bob.

Definition (Dominant)**.** $W \in \mathcal{M}_n(T)$ is said dominant for $(A^{(1)}, A^{(2)})$, $(B^{(1)}, B^{(2)})$, $(A^{(1)}, B^{(2)})$, $(B^{(1)}, A^{(2)}) \in \mathcal{M}_n(T) \times \mathcal{M}_n(T)$ if for all $i, j \in [n]$ and some $s, t \in [n]$ such that $w_{st} = \min\limits_{i,j \in [n]} w_{ij}$, $X, Y \in \{A, B\}$,

$$(X^{(1)} \otimes W \otimes Y^{(2)})_{ij} = x_{is}^{(1)} \otimes w_{st} \otimes y_{tj}^{(2)}$$

When W is dominant, the secret key K can be found with the simple expression

$$k_{ij} = w_{st}^{\otimes -1} \otimes (u_{ij} \otimes v_{st} \oplus u_{st} \otimes v_{ij} \oplus u_{it} \otimes v_{sj} \oplus u_{sj} \otimes v_{it})$$

for $i, j \in [n]$ and U, V the keys sent respectively by Alice and Bob.

Both attacks are really efficient in terms of execution time. It requires to browse once the matrix W, in search of the minimum. That is n^2 comparisons. Then the vanishing attack is completed by a single calculus of usual additions and minima between matrices. The dominant attack is completed by a simple filling of the matrix – *i.e.* n^2 operations – involving usual additions and minima between scalar. Experimentally, for matrices of size 30×30, the running time of both attacks take approximately 0.01 sec.

The existence of both heuristics is problematic for the security of the protocol. Indeed, when the entries of the public matrix W are too small compared to entries of the private matrices A_1, A_2, B_1 and B_2, the matrix W is more likely to be vanishing and then to suffer from the first attack. On the other hand, if the entries of W are chosen too big, the matrix W is likely to be dominant and to suffer from the second attack. We summarize our results in the following array. Parameters a, b are taken in the range $[1, 20]$ and k_1, k_2, l_1, l_2 in $[-100, 0]$.

Size of the matrices	10			20			30		
Entries of W in $+/-$	100	1000	10000	100	1000	10000	100	1000	10000
Vanishing (%)	82	38	38	93	41	36	95	34	34
Dominant (%)	35	78	98	16	44	87	7	30	78
Both (%)	86	80	98	93	50	87	95	37	77

Table 4: Average success rate of the vanishing, the dominant and both attacks on the Protocol 4 depending on the size of the matrices and the entries of W (based on 1000 runs).

Observe that for well-chosen parameters, it is possible to reduce the effectiveness of heuristic attacks to less than 50%.Unfortunately, no matter how much we reduce the success rate of the heuristic, the attack of M. Kotov and A. Ushakov using the Simplex algorithm is still applicable to this protocol, as shown in [18]. The existence of this attack makes this scheme absolutely insecure at the moment. However, according to A. Muanalifah and S. Sergeev, it is a promising area of research.

2 Diffie-Hellman over the tropical matrix semigroup

In 2014, D. Grigoriev and V. Shpilrain introduced the first protocol [7] using the semiring of matrices over a tropical algebra – to the best of our knowledge. Unfortunately, as described in 1.3.2, this protocol has suffered an efficient attack [15] using the Simplex algorithm on a system of equations. This attack exploited the weakness of polynomials as private keys for tropical matrices.

As a respond, they published two new protocols [8] using the semigroup of tropical matrices and involving more advanced mathematics. The point in those protocols is to avoid the use of polynomials by introducing new operations to encrypt the public matrices. In this report we will omit the second one since it has been proven in [11] that the "semigroup" used was not actually associative – *i.e.* not actually a semigroup. As a reminder, the mathematical background is detailed in 3.1 and will not be more developed in this section.

Historically, the idea to extend the Diffie-Hellman's protocol to new algebraic structures has already been explored in many ways. The classical example is the elliptic curves which are a wide research field. Some authors also proposed to construct semigroups using semidirect products [9, 14]. This idea will be presented here through the introduction of tropical algebra in this domain.

In 2013, M. Habeeb *et al.* described a general protocol [9] using a semidirect product to be used on any semigroup. Unfortunately, it has been broken in 2015 by a linear attack [21]. Hence, one can realize how important it is to build a well-chosen semidirect product and therefore a well-chosen group action. The use of semigroups, as interesting as they may be, is not sufficient to avoid linear attacks.

Thereafter, D. Kahrobaei and V. Shpilrain proposed in 2016 [14] a brand new protocol where the set acting on the semigroup G is a subgroup of G. As far as we know, this protocol has not been broken yet and thus is the basis of the tropical scheme presented here. One can consider that this semidirect product is a good candidate, and that it remains to take a well-chosen semigroup to build an efficient protocol. As detailed further, the tropical semigroup is not a relevant choice.

2.1 Diffie-Hellman's standard protocol

First of all, we introduce the Diffie-Hellman key exchange protocol over an abelian group.

Protocol 5. *Diffie-Hellman.*

1. *Alice and Bob agree on a prime number p and g a generating element in of $\mathbb{Z}/p\mathbb{Z}$.*
2. *Alice chooses a private integer x and sends $u = g^x$ to Bob*
   Bob chooses a private integer y and sends $v = g^y$ to Alice
3. *Alice computes $K_A = v^y$ in $\mathbb{Z}/p\mathbb{Z}$*
   Bob computes $K_B = u^x$ in $\mathbb{Z}/p\mathbb{Z}$
4. *The shared secret key is $K = K_A = K_B$*

The security of the protocol is based on the *discrete logarithm problem* (DLP) in abelian groups – typically $\mathbb{Z}/p\mathbb{Z}$. This problem is defined as follow:

Definition. Let G be a finite abelian group, $g, h \in G$ and $a, b \in \mathbb{N}$ such that $h = g^a$. Finding a when g and h are known is called the *discrete logarithm problem*. Finding g^{ab} when g^a and g^b are known is called the *Diffie-Hellman problem*.

This well-known protocol is considered rather secure since the DLP – more specifically the Diffie-Hellman problem (DHP) – is a hard problem that, although weakened, has never been attacked efficiently. Indeed, some algorithm have been found to solve the discrete logarithm problem, like Pohlig-Hellman [20] or Shank [23] to cite the most famous, but never in a reasonable time. Thus, the DHP is not considered as broken yet. However, agreeing with E. Stickel, the size of the key has been chosen increasingly larger.

2.2 Tropical protocol

Now we will discuss the tropical key exchange based on the Diffie-Hellman protocol, introduced in [8]. Recall that the scheme is built on the set of pairs (M, H), where M and H are tropical matrices. Since the protocol only requires tropical exponentiation, the binary operation considered in practice will be $(M, H)^{\odot 2} = ((M \circ H) \oplus M, H^2)$. One may notice that the second part of the pair is simply squared and therefore $(M, H)^{\odot n} = (N, H^n)$, where N depends of M and H. $(M, H)^{\odot n}$ will be simply denoted $(M, H)^n$ afterwards.

The protocol can be summarized by the following steps:

Protocol 6. *Tropical Diffie-Hellman.*

1. *Alice and Bob agree on public matrices* $M, H \in \mathcal{M}_n(T)$.
2. *Alice chooses a private integer* a *and computes* $(M,H)^a = \overbrace{(H,M) \circ ... \circ (H,M)}^{a \text{ times}} := (A, H^a)$
   Bob chooses a private integer b *and computes* $(M,H)^b = (B, H^b)$
3. *Alice and Bob send* A *and* B *to each other.*
4. *Alice computes* $K_a = (B \circ H^a) \oplus A$ *and Bob computes* $K_b = (A \circ H^b) \oplus B$. *The shared secret key is* $K = K_a = K_b$

One verifies that $K_a = K_b$ since $(A \circ H^b) \oplus B$ is equal to the first component of $(A, H^a) \circledcirc (B, H_b)$, such as $(B \circ H^a) \oplus A$. Thus K is equal to the first component of $(M,H)^{a+b}$.

Example. *Assume that* $M = \begin{pmatrix} 1 & -1 \\ 0 & -2 \end{pmatrix}$, $H = \begin{pmatrix} 3 & 1 \\ -1 & -2 \end{pmatrix}$ *and* $a = 3$, $b = 4$.

Alice computes $(M,H)^3 = \left(\begin{pmatrix} -4 & -5 \\ -5 & -6 \end{pmatrix}, \begin{pmatrix} -2 & -3 \\ -5 & -6 \end{pmatrix} \right)$ *and sends* $\begin{pmatrix} -4 & -5 \\ -5 & -6 \end{pmatrix}$ *to Bob.*

Bob computes $(M,H)^4 = \left(\begin{pmatrix} -6 & -7 \\ -7 & -8 \end{pmatrix}, \begin{pmatrix} -4 & -5 \\ -7 & -8 \end{pmatrix} \right)$ *and sends* $\begin{pmatrix} -6 & -7 \\ -7 & -8 \end{pmatrix}$ *to Alice.*

Alice computes $\left(\begin{pmatrix} -6 & -7 \\ -7 & -8 \end{pmatrix} \circ \begin{pmatrix} -2 & -3 \\ -5 & -6 \end{pmatrix} \right) \oplus \begin{pmatrix} -4 & -5 \\ -5 & -6 \end{pmatrix} = \begin{pmatrix} -12 & -13 \\ -13 & -14 \end{pmatrix}$.

Bob computes $\left(\begin{pmatrix} -4 & -5 \\ -5 & -6 \end{pmatrix} \circ \begin{pmatrix} -4 & -5 \\ -7 & -8 \end{pmatrix} \right) \oplus \begin{pmatrix} -6 & -7 \\ -7 & -8 \end{pmatrix} = \begin{pmatrix} -12 & -13 \\ -13 & -14 \end{pmatrix}$

$K = \begin{pmatrix} -12 & -13 \\ -13 & -14 \end{pmatrix}$ *is the private shared key.*

One remarks that the matrices seem to be decreasing for the partial order given by $A \leq B$ *iff* $A \oplus B = A$. It has been proven by D. Rudy and C. Monico in [22] as a basis for their attack described in 3.

2.2.1 Security

In this section will be analyzed the advantages and the drawbacks of this scheme, in term of security and potential improvements.
The parameters suggested by D. Grigoriev and V. Shpilrain are:

- Size of matrices: 30.
- Entries of matrices: $[-1000, 1000]$.
- Exponents: on the order of 2^{200}.

Therefore, there are 2001^{900} possibilities for each matrix that is 2001^{1800} for two matrices. Each private integer is on the order of 2^{200}, thus there are more than $2001^{1800} \times 2^{400} \simeq 20,165$ bits for an exhaustive search attack. This is way more than usual public key exchange protocols. Indeed, if for most of them such parameters would make the scheme impossible to compute for a practical use, the fact that the tropical protocols do not require exponentiation nor multiplication allows a choice of larger parameters.

We can roughly compare this result with the original Diffie-Hellman protocol. Currently, the recommended parameters (from NIST) are 224 bits for the private keys and 2048 bits for the size of the public group – according to [1]. That means 2^{224} possibilities for each secret key and 2^{2048} bits for the public element g, for a total of 2496 bits, which is far less than the 20,000 bits proposed by Grigoriev and Shpilrain. However, the number of possible combinations is not a relevant indicator of the security of a system, as it is shown in 3.

As Diffie-Hellman is based on the DLP, its tropical variant is based on such a problem, defined below:

Definition (Tropical discrete logarithm problem)**.** Let $A, B, U \in \mathcal{M}_n(T)$ and $k \geq 1$ an integer such that

$$U = A \otimes B^{\otimes k}$$

Finding k when A, B and U are known is called the Tropical discrete logarithm problem (TDLP).

Unlike the standard DLP, the TDLP is not a problem anymore since it has been fully described, analyzed and "solved" by A. Muanalifah and S. Sergeev in [19] with an efficient algorithm. The demonstration is based on advanced algebraic tools like tropical eigenvalues and Kleene stars which have been developed with the recent rise of tropical geometry. Therefore, even though the adaptation of Diffie-Hellman to new algebraic structures is potentially a good idea, it is no longer feasible over a tropical matrix semigroup.

2.2.2 Attacks

This protocol has suffered from two attacks using the same kind of pattern: a first one by D. Rudy and C. Monico in [22] and a second one by S. Isaac and D. Kahrobaei in [11]. Their implementations are available in [10]. Then a last attack using the solution of the TDLP has been proposed by A. Muanalifah and S. Sergeev in [19].

The Rudy-Monico attack is based on the fact that – as evoked above – the sequence of matrices is decreasing. Indeed, it is then really easy to find a frame of the values to determine the private exponent. This is the only attack we will study in this section since it is the only one we implemented it ourselves. The algorithm is detailed below.

Recall that (M, H) are public matrices, and Alice computes $(M, H)^n = (M_n, H^n)$ for an integer $n \geq 1$. Suppose that an eavesdropper intercepts the matrix M_n sent by Alice to Bob. Since the sequence $(M_i)_{i\in[n]}$ is (strictly) decreasing, she can efficiently find a frame for M_n by computing $(M_{2^i})_{i\in[n]}$ which only requires squaring. Thus she finds $t \geq 1$ such that $M_{2^{t-1}} \geq M_n \geq M_{2^t}$. It is then easy to retrieve n by a simple binary search.

Algorithm 3 Rudy and Monico's attack

```
1: procedure RudyMonico(A, B, M, H)
2:     t ← 1
3:     (K, J) ← (M, H)
4:     while K > A do
5:         (K, J) ← (K, J)^2
6:         t ← t + 1
7:     P ← BinarySearch(A, M, H, 2^{t−1}, 2^t)
8:     return B ⊕ A ⊕ P ⊕ (B ⊗ P)
```

The algorithm BinarySearch is the adapted case of the classical binary search algorithm – see below.

Algorithm 4 Binary Search

```
1: procedure BinarySearch(A, M, H, min, max)
2:     n ← (min + max)/2
3:     (K, J) = (M, H)^n
4:     if K = A then
5:         return H^n
6:     else if K < A then return BinarySearch(A, M, H, min, n)
7:     else return BinarySearch(A, M, H, n, max)
```

The cost of this algorithm is detailed in [22]. Basically, if k is the number of bits of the private keys of Alice and Bob (n, m), the first step of the algorithm requires k operations in $\mathcal{S}$. Then, the binary search requires $2k$ operations for each call. It has to be called k times in average, which leads to a total of $2k^2 + k$ operations to find the secret key. The conclusion is that Bob and Alice compute $\mathcal{O}(k)$ operations in the semigroup S when an eavesdropper needs to compute $\mathcal{O}(k^2)$ operations. It means that increasing the size of the keys will not prevent the protocol from this attack.

3 Public key encryption scheme

This section is about the public key encryption scheme introduced by D. Grigoriev and V. Shpilrain in [7] based on the TTM (for "tame transform method") cryptosystem proposed by T.T. Moh in [17]. It is

strongly correlated to the family of schemes TPM, whose story will be quickly presented now.

3.1 Origin

As pointed out in [6], the research effort in practical public key cryptography, introduced by R. Rivest, A. Shamir and L. Adleman with univariate polynomials over $\mathbb{Z}/n\mathbb{Z}$, followed two paths. The first, nowadays mainly represented by the elliptic curves, is considering more complex groups. The second is considering multivariate equations.

One of the paradigms for constructing multivariate trapdoor cryptosystems is the triangular construction, proposed initially in an iterated form by H. Fell and W. Diffie in [5]. A triangular construction uses equations that involves $1, 2, ..., n$ variables and are solved sequentially. The special form of those equations is hidden by two linear transformations on inputs and outputs.

A family of schemes based on triangular construction is the TPM (for Triangle construction Plus-Minus) which consists of a triangle construction with u new random equations (the Plus) and r of the beginning equations removed (the Minus). One can denote by $TPM(n, u, r, GF(q))$ the set of cryptosystem constructed as TPM with equations that involves $1, 2, ..., n$ variables on the finite field of q elements, with u random new equations added and r equations from the beginning removed.

In [6], L. Goubin and N. Courtois broke $TPM(n, u, r, GF(q))$ with q^r "small". To do so, they showed that TPM can be reduced to the linear algebra problem called MinRank. MinRank is a famous NP-Hard problem which can be resumed as finding a linear combination of given matrices that has a small rank r. Even if it is NP-Hard it can actually be solve in polynomial times for small entries, we refer the reader to [3] for more details. They managed to introduce a new linear attack called "Kernel Attack" which works for q^r small and thus which can break a $TPM(n, u, r, GF(q))$ scheme with small q^r as well.

In [17], T.T. Moh proposed a public key encryption scheme which is a subcase of TPM. In this cryptosystem, the triangular construction is made from automorphism of finite affine space. With suggested parameters of the TTM, it belongs to the family of $TPM(64, 38, 2, GF(256))$. Then for TTM schemes $q^r = 256^2$ which is small enough to use the linear "Kernel attack", thus this cryptosystem is totally broken. Again, we refer the reader to [6] for more details and for other vulnerabilities of the TTM schemes, the key points being that all attacks are based on linearity flaws.

By adapting the TTM scheme to tropical structures, D. Grigoriev and V. Shpilrain avoid all the existing linear attacks on this protocol and further reduce the computational complexity. Here the considered automorphism group will be of the semifield of fractions of a tropical polynomial semiring over $\mathbf{Z}$. It will be denoted $\mathcal{A}ut$ and the semifield denoted $Rat[x_1, ..., x_n]$. We remind the reader that all these notions have be introduced in 2.3 and 2.4. As in an usual TPM scheme, there is a triangular construction and linear transformations to hide it, especially one on the inputs and one on the outputs. In this case the triangular part will be the triangular automorphisms and the linear part will be the monomial automorphisms, as we defined them in 2.4.2 and 2.4.1.

3.2 How it works

Unlikely the secret-key cryptography, ciphering key and deciphering key are not the same. Suppose Bob wants to send a secret message m to Alice via an public key encryption scheme. First, he needs to encrypt his message using Alice's public key. Then Alice will decrypt the received message using her private key which she is the only one to know.

The encryption function which uses the public key is a one-way function with trapdoor. It is a one-way function in the sense that it is easy to compute but hard to determine its inverse function. Furthermore, to be usable, there is a trapdoor function to decrypt using the private key. Usually one constructs the public key from the private key.

3.2.1 Protocol

The set of plaintexts and ciphertexts are the same, one can theoretically consider the whole set $\mathbb{Z}^n$.
The public key is an automorphism $\alpha \in \mathcal{A}ut$ and the private key is the inverse automorphism α^{-1}.

Remember (2.4) that an automorphism α of $Rat[x_1, ..., x_n]$ is given by a tuple of tropical rational functions $(\alpha(x_1), ..., \alpha(x_n))$.

A communication from Bob to Alice using the Grigoriev-Shpilrain public key encryption scheme follows the steps below:

1. Bob wants to send a secret message $m = (m_1, ..., m_n) \in \mathbb{Z}^n$ to Alice.
2. Bob encrypts the tuple m by applying the public automorphism α to the tuple m.
   He gets the ciphertext $E_\alpha(m) = \alpha((m_1, ..., m_n)) := (\alpha(x_1)(m_1, ..., m_n), ..., \alpha(x_n)(m_1, ..., m_n))$.
3. Bob sends $E_\alpha(m)$ to Alice.
4. Alice decrypts the tuple $E_\alpha(m)$ by applying her private automorphism α^{-1} to the tuple $E_\alpha(m)$.
   She gets the plaintext $\alpha^{-1}(E_\alpha(m)) = ((\alpha^{-1}(x_1))(\alpha(x_1)(m_1, ..., m_n)), ..., (\alpha^{-1}(x_n))(\alpha(x_n)(m_1, ..., m_n))$
   $= ((\alpha^{-1}(x_1) \circ \alpha(x_1))((m_1, ..., m_n)), ..., (\alpha^{-1}(x_n) \circ \alpha(x_n))((m_1, ..., m_n)))$
   $= (x_1(m_1, ..., m_n), ..., x_n(m_1, ..., m_n)) = (m_1, ..., m_n) = m$.

Example. *Let's do an example with a simple automorphism α of $Rat[x, y]$ given by the tuple $(\alpha(x), \alpha(y)) = (10 \otimes x \otimes y^{\otimes 2} \oslash 0, 2 \oslash x^{\otimes 2} \otimes y^{\otimes 3})$. The inverse automorphism α^{-1} is given by the tuple $(\alpha^{-1}(x), \alpha^{-1}(y)) = (x^{\otimes -3} \otimes y^{\otimes -2} \oslash -34, x^{\otimes 2} \otimes y \oslash 22)$.*
Alice's private key is α^{-1} and her public key is α. Suppose Bob wants to send the message $m = (13, 12)$ to Alice.
First Bob encrypts m by applying the public automorphism α to m.

$$\begin{aligned} E_\alpha(m) = \alpha(m) &= ((\alpha(x))(13, 12), (\alpha(y))(13, 12)) \\ &= (10 \otimes 13 \otimes 1^{\otimes 2} \oslash 0, 2 \oslash 13^{\otimes 2} \otimes 12^{\otimes 3}) \\ &= (47, -60) \end{aligned}$$

Bob sends $E_\alpha(m)$.
Alice retrieves m by applying the private automorphism α^{-1} to $E_\alpha(m)$.

$$\begin{aligned} E_{\alpha^{-1}}(E_\alpha(m)) &= \alpha^{-1}(47, -60) \\ &= ((\alpha^{-1}(x))(47, -60), (\alpha^{-1}(y))(47, -60)) \\ &= (47^{\otimes -3} \otimes -60^{\otimes -2} \oslash -34, 47^{\otimes 2} \otimes -60 \oslash 22) \\ &= (13, 12) \end{aligned}$$

3.2.2 Key Generation

Alice's public key is an automorphism $\alpha \in \mathcal{A}ut$ with an underlying triangular construction.

As it has been said before, it is important not to have an explicit triangular construction, its triangularity emanates here from the triangular automorphisms. Hence linear transformations, which are here the monomial automorphisms, are added at least on the inputs and the outputs. Therefore α must be generated as a product of triangular and monomial automorphisms.
For this purpose, D. Grigoriev and V. Shpilrain suggest an automorphism α of the following form :

$$\alpha = \mu_1 \circ \tau_1 \circ \mu_2 \circ \tau_2 \circ \mu_3$$

where $\mu_1, \mu_2, \mu_3 \in \mathcal{A}ut$ are monomial automorphisms and $\tau_1, \tau_2 \in \mathcal{A}ut$ are triangular automorphisms.

Alice's private key is α^{-1} the inverse automorphism of α, thus it has the following form :

$$\alpha^{-1} = \mu_3^{-1} \circ \tau_2^{-1} \circ \mu_2^{-1} \circ \tau_1^{-1} \circ \mu_1^{-1}$$

Notice that Alice can simply consider the list of inverse automorphisms $[\mu_3^{-1}, \tau_2^{-1}, \mu_2^{-1}, \tau_1^{-1}, \mu_1^{-1}]$ as her private key. Using the list of inverses instead of their product, or even directly the list of the factor of α, gives some advantages as we will see in 3.3.

In addition to proposing a precise form for $\alpha = \mu_1 \circ \tau_1 \circ \mu_2 \circ \tau_2 \circ \mu_3$, V. Shpilrain and D. Grigoriev suggest precise parameters for the protocol :

- The number of variables of $Rat[x_1, ..., x_n]$ is $n = 10$.
- The triangular automorphisms τ_1 and τ_2 are product of elementary triangular automorphisms with p_i, defined as in 2.4, of degree 2 and their coefficients selected uniformly randomly in the range $[-10, 10]$.

Remark. Nothing is mentioned on the monomial automorphism in [7]. We will describe explicitly how we managed to generated such automorphisms in the implementation section 3.3.

3.2.3 Alternative version of the protocol

Instead of using the set of plaintext $\mathbb{Z}^n$, it is possible to use directly $Rat[x_1, ..., x_n]$. In this case, Bob's message is a tropical rational function $P \in Rat[x_1, ..., x_n]$ and he sends $\alpha(P)$ as it has been defined in 2.4.

Proposition. *The alternative version of Grigoriev-Shpilrain public key encryption scheme is a totally tropical homomorphic cryptosystem, i.e.* $\forall u_1, u_2 \in Rat[x_1, ..., x_n]$

$$\alpha(u_1 \oplus u_2) = \alpha(u_1) \oplus \alpha(u_2)$$
$$\alpha(u_1 \otimes u_2) = \alpha(u_1) \otimes \alpha(u_2)$$

Proof. The proposition is directly verified by the definition of semiring automorphism (2.4). □

3.3 Implementation

First of all, we have all the information needed to generate triangular automorphisms as suggested by V. Shpilrain and D. Grigoriev in [7] but we have to choose the parameters of the monomial automorphisms. We tried to have relatively similar ones for both type of automorphisms, using the same notation as in 2.4:

- The b_i are selected uniformly randomly in the range $[-10, 10]$.
- The matrix A of dimension n and determinant 1 with entries in the range $[-10, 10]$.

Remark. To construct the matrix A, we apply several random transformations which conserve the determinant on the identity matrix.

3.3.1 Possibilities and limits of our implementation

Unfortunately, the implementation which we propose at [10] does not support the Grigoriev-Shpilrain public encryption key scheme with parameters they suggest. In fact, with our implementation it is possible to manipulate the private key, to encrypt and decrypt, but not to create the public key. It comes from the fact that computing public key, with 10 variables and 5 factor automorphisms in it, involves too much composition of automorphism even for tropical algorithms. At least, that is the case with our program. However, according to V. Shpilrain, this protocol had not yet been implemented and so these parameters may lead to unexpected compilation problems.

These considerations lead us to suggest new parameters for this protocol, which keep good security properties as we will see in the next subsection (3.4):

- The number of variables of $Rat[x_1, ..., x_n]$ is $n = 3$.
- The public automorphism $\alpha \in \mathcal{A}ut$ is given as

  $$\alpha = \mu_1 \circ \tau \circ \mu_2$$

  where μ_1, μ_2 are monomial automorphisms of $Rat[x_1, x_2, x_3]$ and τ is a triangular automorphism of $Rat[x_1, x_2, x_3]$. Thus it conserves the underlying triangular construction and the necessary linear transform on the input and output.
- The triangular automorphisms τ_1 and τ_2 are products of elementary triangular automorphisms with p_i, defined as in 2.4, of degree 2 and their coefficients selected uniformly randomly in the range $[-10, 10]$ – as in [7].

- The monomial automorphisms have entries b_i selected uniformly randomly in the range $[-10, 10]$ and the matrix A of dimension n and determinant 1 with entries in the range $[-10, 10]$, as previously.

We made around 3000 tests to study the rapidity of the protocol and the length of the public and private key with these parameters

average length of the factor list of α	344 char
average length of the inverse factor list of α	1670 char
average length of α	16500 char
average time to create the factor list of α (and their inverse)	0.06 sec
average time to create the explicit form of α	0.92 sec

Table 5: Average running time and length of the different key form of the Grigoriev-Shpilrain public key encryption scheme with our suggested parameters.

Notice the explicit form of α requires about 50 times more characters to be stored and is by far the slowest part of the key creation. Thus it makes sense not to compute the explicit form of α^{-1} and to use only the list of these factors which are very quick to compute and much lighter to store. Especially since the degrees of the α factor automorphisms are very low, we can expect to have much higher ones for their inverse and thus an α^{-1} even larger and more expensive in computation. By the way, it is a good point for the security properties of this protocol to have an α^{-1} difficult to compute .

We also checked the average time to encrypt a message. First using the explicit form of α and then using its factors. We did this on 1000 instances and obtained the following results :

- The average time with explicit α is 10.4 ms.
- The average time with its factors is 0.2 ms.

Clearly, there is no computational advantage to have an explicit form for automorphisms instead of its factor automorphisms. Thus we will suggest to compute explicit form only for α, and not for α^{-1}.

The last thing we studied with our implementation is the size of the encrypted messages set to optimize it. This protocol being a public encryption key scheme, having a huge set of encrypted messages does not add any security. Indeed an attacker who wants to brute force attack the message will simply encrypt all the possible plain messages with the public key.

We did our first experiments on plain texts of 8 bits.

Number of bits	8	8	8
Min values for the plain	$[-3, -3, -1]$	$[0, 0, 0]$	$[-7, -7, -3]$
Max values for the plain	$[4, 4, 2]$	$[7, 7, 3]$	$[0, 0, 0]$
Min values for the cipher	$[-1057, -1204, -1336]$	$[-1951, -1849, -2277]$	$[-2213, -1632, -2154]$
Max values for the cipher	$[1067, 1164, 949]$	$[2179, 1536, 2290]$	$[1865, 1403, 1384]$
Needed bits for the cipher	$\simeq 33$	$\simeq 36$	$\simeq 36$

Table 6: Maximum and minimum values for the cipher tuple depending on the chosen plain set of tuple. Based on 1000 instances of random automorphisms and 100 evaluation of random values for each of them.

Unsurprisingly, we have to try to lower the absolute value in order to have a more compact cipher tuple set.
With this method, the plain tuple set for 128 bits could be from the minimum values $[-2^{42}+1, -2^{42}+1, -2^{41}+1]$ to the maximum values $[2^{42}, 2^{42}, 2^{41}]$. Again, we did 1000 instances of random automorphisms and 100 evaluation of random values in this plain set for each of them. It gives us that the cipher set needs at least 153 bits. In the same way, for 256 bits we need at least 280 bits to store the cipher text. The implementation and its results are available in [10].

3.4 Security

Proposition. *A brute force attack against Grigoriev-Shpilrain public key encryption scheme with at least 3 variables is not practicable.*

Proof. Let α be an automorphism of $Rat[x_1, ..., x_n]$, with $n \geq 2$, generated as required for the Grigoriev-Shpilrain public key encryption scheme, *i.e.* of the form :

$$\alpha = (\bigcirc_{i=1}^{k} (\mu_i \circ \tau_i)) \circ \mu_{k+1}$$

where $k \geq 0$, $\mu_i, \forall i \in [1, k+1]$, are monomial automorphisms in $\mathcal{A}ut$ and $\tau_i, \forall i \in [1, k]$, are triangular automorphisms in $\mathcal{A}ut$.
Notice that if $k = 2$, we get the suggested parameters from D. Grigoriev and V. Shpilrain and if $k = 1$ we get our suggested parameters.

In any case, there is at least a triangular automorphism in the construction of the public key. Thus there is also at least an inverse triangular automorphism in the construction of the private key. There are, of course, as many triangular automorphisms as there are inverse of them.

Let us count the number of possible triangular automorphism in $\mathcal{A}ut$:
Let $\tau \in \mathcal{A}ut$ be a triangular automorphism given as a tuple of tropical rational functions $(t_1, ..., t_n)$ where t_i is of the form

$$t_i = x_i \otimes p_i(x_{i+1}, ..., x_n),\ 1 \leq i \leq n, \qquad \text{where } p_i \in Rat[x_1, ..., x_n]$$

The suggested parameters for the numerator and the denominator of each p_i require a tropical degree 2 and coefficients of their monomials selected uniformly randomly in the range $[-10, 10]$. First of all if $i = n$, p_i is a constant monomial and since there are only 21 possible monomials, there are only 21 possible p_n. From there, we will determinate the number of possible monomials in the numerator and denominator of each $p_i, 1 \leq i < n$.

- There are 21 possible constant monomials.
- There are $21 \times (n - i)$ possible monomials of degree 1 or degree 2 with only one variable to the power of 2 because p_i has as its only variables $x_{i+1}, ..., x_n$.
- There are $21 \times \binom{n-i}{2}$ possible monomials of degree 2 with two variable to the power of 1 because we are choosing two variables between x_{i+1} and x_n.

Therefore the set of possible monomials in the numerators or denominators of each p_i is of size $21 \times (1 + 2 \times (n - i) + \binom{n-i}{2})$, we denote $m_{i,n}$ this quantity. Not to neglect the equivalent rational functions, we will focus only on the possible numerators. Therefore, there are at least

$$21 \times \prod_{i=1}^{n-1} 2^{m_{i,n}}$$

triangular automorphisms. This formula gives us a large lower bound of the number of possibilities of triangular automorphisms with n variables.

Here is a table of the approximation of the number of triangular automorphism in $Rat[x_1, ..., x_n]$, denoted $|\{\tau \in \mathcal{A}ut\}|$, for different value of n:

n	$\lvert\{\tau \in \mathcal{A}ut\}\rvert$	n	$\lvert\{\tau \in \mathcal{A}ut\}\rvert$
1	21	6	$\simeq 2^{1160}$
2	$\simeq 2^{68}$	7	$\simeq 2^{1748}$
3	$\simeq 2^{194}$	8	$\simeq 2^{2504}$
4	$\simeq 2^{404}$	9	$\simeq 2^{3449}$
5	$\simeq 2^{719}$	10	$\simeq 2^{4604}$

Table 7: Approximation of the number of possible triangular automorphisms of $Rat[x_1, ..., x_n]$ in function of n.

By corollary, there are more than 2^{194} possibilities for the automorphism α^{-1} if $n \geq 3$. Obviously the private key set is large enough to be out of a basic bruteforce attack range. □

We can have a more precise approximation of possible automorphisms α^{-1} by looking at the possible inverse monomial automorphisms. Again, note that there are as many inverse monomial automorphisms as there are monomial automorphisms.

We will take a look here only at the number of monomial automorphisms with our suggested parameters. To have an approximation of possible monomial automorphisms of $Rat[x_1, x_2, x_3]$, we started by looking at the proportion of invertible matrices with entries in the range $[-10, 10]$. To do so, we randomly generated one million of matrices and checked if they were invertible – this implementation is available in [10]. We found out there were around 1073 invertible matrices. If we generalise this proportion to the all set of matrices with entries in the range $[-10, 10]$, we find around $\frac{1073}{10^6} \times 21^9 \simeq 2^{30}$ invertible matrices. Moreover, there are $21 \times 3 = 63 \simeq 2^6$ possible sets of b_i for a monomial automorphism. In conclusion there are approximately $2^6 \times 2^{30} = 2^{36}$ possible monomial automorphism of $Rat[x_1, x_2, x_3]$ with our suggested parameters.

From there, we know there are at least $2^{36} \times 2^{194} \times 2^{36} = 2^{266}$ possible private keys α^{-1} of the suggested form $\alpha^{-1} = \mu_2^{-1} \circ \tau^{-1} \circ \mu_1^{-1}$ with our parameters. Again notice that this is a very large lower bound.

Two other attacks are enunciated and showed in theory not practicable in [7]:

- Directly computing the inverse of the public key to retrieve the private key :
  An eavesdropper who wants to attempt this will be hitting the fact that the degree of α^{-1} may be exponentially greater than the degree of α.
  For example, we instantiated 1000 triangular and monomial automorphisms and their inverse to compare their degree.

automorphisms	maximum degree
triangular	3
monomial	9
triangular inverse	19
monomial inverse	87

Table 8: Maximum degree of triangular and monomial automorphisms compared to their inverse. Made on 1000 instances for each kind of automorphisms.

- Retrieving the plaintext m from $\alpha(m)$:
  An eavesdropper who wants to attempt this will have to solve a system of min-plus polynomials equations. Solving such a system is an NP-hard problem as it has been show in [7].

Remark. At the present time, to the knowledge of V. Shpilrain, there is not a single article of cryptanalysis which presents an attack on this protocol or any weakness of it. The entirety of the tropical cryptanalysis is focused of the two other protocols.

Since the Grigoriev-Sphilrain public key encryption scheme does not have exposed weakness yet and has more than 2^{266} private key possibilities with our suggested parameters, it could be a good candidate to do public cryptography, for example to exchange AES – 128 or 256 bits – keys, with fast computation time. Unfortunately, the lack of known weakness probably stems from the absence of cryptanalysis on this protocol.

Conclusion

Tropical algebra is a new field of research. Widely used in geometry, it is even more recent in the cryptography world. However, in six years, already seven articles have been published on the subject, including four in the last two years. In these papers, five protocols and six attacks have been proposed. There are two reasons for this enthusiasm: the high computational speed offered by the min-plus structure, and its (supposed) higher security due to its semiring structure.

Indeed the choice of tropical algebra as platform for protocols seems to be a good way to work on structures with less properties and thus less weaknesses. It turns out that some powerful algorithms are still able to break them through linear attacks. At the present time, the Simplex attack is clearly the nemesis of tropical cryptography and countering it would give a real second wind to the field.

Since it is a fruitful field, the majority of attacks have resulted in new ideas to build variants. Among these we can notably cite the new classes of commutativity which stem from an in-depth work on tropical algebra in general. We can also cite the idea to add infinite entries to the matrices which has not been completed at this time but could lead to further researches.

Despite the novelty of this subject, the tropical cryptosystems are based on already existing protocols. This often leads to schemes that are actually not well adapted to the tropical semiring, especially when working on matrices. In addition to trying to improve existing protocols, one can also imagine that an entirely tropical protocol could appear and thus fully exploit the potential of this domain.

Finally, there is a total lack of cryptanalysis on the Grigoriev-Shpilrain public key encryption scheme, even if it seems to offer a good possibility for asymmetric cryptography at first glance. It is indeed a use of the tropical in a context where its speed advantage would really make the difference with much slower other public protocols.

Tropical cryptography will be interesting to watch !

Acknowledgments

We would first like to thank Sylvain Duquesne who gave us the opportunity to write this report and helped us throughout our work.
We would like to acknowledge Vladimir Shpilrain, Sergey Sergeev and Any Muanalifah who accepted to answer our questions about their articles.
We would also like to thank Clara Delahaye who introduced us to ASP and who reviewed our work.
Finally, we would like to thank Grégory Gobin who suggested us to work on the tropical algebras.

References

[1] BlueKrypt. Cryptography key length recommendation. `https://www.keylength.com/`.

[2] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms, 3rd. edition*. The MIT Press, 3rd edition, 2009.

[3] N. Courtois. *La sécurité des primitives cryptographiques basées sur des problèmes algébriques multivariables MQ, IP, MinRank, HFE*. PhD thesis, Université de Paris 6 - Pierre et Marie Curie, 2001.

[4] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE transactions on information theory*, 1976.

[5] H. Fell and W. Diffie. Analysis of a public key approach based on polynomial substitution. *Springer, Berlin, Heidelberg*, 218, 08 1985.

[6] L. Goubin and N. Courtois. Cryptanalysis of the ttm cryptosystem. *Springer, Berlin, Heidelberg*, 2000. Advances in Cryptology ASIACRYPT 2000.

[7] D. Grigoriev and V. Shpilrain. Tropical cryptography. *Communications in Algebra*, July 2014.

[8] D. Grigoriev and V. Shpilrain. Tropical cryptography II: extensions by homomorphisms. *arXiv:1811.06386*, Nov. 2018.

[9] M. Habeeb, D. Kahrobaei, C. Koupparis, and V. Shpilrain. Public key exchange using semidirect product of (semi)groups. *Springer, Berlin, Heidelberg*, Apr. 2013.

[10] A. Herledan Le Merdy and C. Foucault. Tropical cryptography. `https://github.com/Riemannujan/tropical_cryptography`, 2021.

[11] S. Isaac and D. Kahrobaei. A closer look at the tropical cryptography. *CoRR*, 2020.

[12] I. V. Itenberg, G. Mikhalkin, and E. Shustin. *Tropical algebraic geometry*. Springer Science & Business Media, 2007.

[13] D. L. Jones. *Special and structured matrices in max-plus algebra*. PhD thesis, University of Birmingham, 2017.

[14] D. Kahrobaei and V. Shpilrain. Using semidirect product of (semi)groups in public key cryptography. *Springer, Cham*, Apr. 2016.

[15] M. Kotov and A. Ushakov. Analysis of a key exchange protocol based on tropical matrix algebra. *Journal of Mathematical Cryptology*, Sept. 2018.

[16] J. Linde and M. J. de la Puente. Matrices commuting with a given normal tropical matrix. *Linear Algebra and its Applications*, Dec. 2014.

[17] T. T. Moh. A public key system with signature and master key functions. *Communications in Algebra*, 1999.

[18] A. Muanalifah and S. Sergeev. Modifying the tropical version of Stickel's key exchange protocol. *Applications of Mathematics*, 2019.

[19] A. Muanalifah and S. Sergeev. On the tropical discrete logarithm problem and security of a protocol based on tropical semidirect product. *arXiv:2101.02781*, Jan. 2021.

[20] S. Pohlig and M. Hellman. An improved algorithm for computing logarithms over GF(p) and its cryptographic significance (corresp.). *IEEE Transactions on Information Theory*, 1978.

[21] V. Roman'kov. Linear decomposition attack on public key exchange protocols using semidirect products of (semi)groups. *arXiv: Cryptography and Security*, 2015.

[22] D. Rudy and C. Monico. Remarks on a tropical key exchange system. *De Gruyter*, Sept. 2020.

[23] D. Shanks. Class number, a theory of factorization, and genera. *American Mathematical Society*, 1971.

[24] V. Shpilrain. Cryptanalysis of Stickel's key exchange scheme. *Springer, Berlin, Heidelberg*, June 2008.

[25] V. Shpilrain and A. Ushakov. Thompson's group and public key cryptography. *Springer, Berlin, Heidelberg*, 3531:175–217, 05 2005.

[26] E. Stickel. A new method for exchanging secret keys. *IEEE*, 2005. Third International Conference on Information Technology and Applications.