On the tropical two-sided discrete logarithm and a key exchange protocol based on the tropical algebra of pairs

Sulaiman Alhussaini, Craig Collett and Sergeĭ Sergeev

Abstract

Since the existing tropical cryptographic protocols are either susceptible to the Kotov-Ushakov attack and its generalization, or to attacks based on tropical matrix periodicity and predictive behaviour, several attempts have been made to propose protocols that resist such attacks. Despite these attempts, many of the proposed protocols remain vulnerable to attacks targeting the underlying hidden problems, one of which we call the tropical two-sided discrete logarithm with shift. An illustrative case is the tropical Stickel protocol, which, when formulated with a single monomial instead of a polynomial, becomes susceptible to attacks based on solutions of the above mentioned tropical version of discrete logarithm. In this paper we will formally introduce the tropical two-sided discrete logarithm with shift, discuss how it is solved, and subsequently demonstrate an attack on a key exchange protocol based on the tropical semiring of pairs. This particular protocol is compromised due to the existence of efficient (albeit heuristic) solution of the tropical two-sided logarithm problem, and this highlights the ongoing challenges in search of a "good" key exchange protocol in tropical cryptography.

Keywords: public key cryptography; key exchange protocol; cryptographic attack; tropical cryptography
Classification: 94A60, 15A80

1 Introduction

Tropical cryptography is a new and promising area that seeks to transform traditional public key exchange methods in cryptography using tropical mathematics. Grigoriev and Shpilrain pioneered the use of tropical algebra as an alternative framework for cryptographic protocols [5]. They developed a tropical version of the original Stickel key exchange protocol, which was vulnerable to common linear algebraic attacks. Their motivation came from the non-invertible nature of matrices in tropical algebra, making the tropical implementation resistant to attacks resembling the ones faced by the original Stickel protocol. However, this tropical implementation has been attacked by Kotov and Ushakov [8]. The Kotov-Ushakov attack was generalized in [11] where it was shown how to apply the same idea to other implementations of Stickel protocol based on matrix commutativity. This has prompted the exploration of alternative ideas beyond matrix commutativity in Stickel protocols to implement tropical cryptographic protocols.

In response, Grigoriev and Shpilrain [6] proposed two protocols based on tropical semidirect product, but one of them was shown to be invalid by Isaac and Kahrobaei [7] and the other was successfully attacked by the same authors as well as by [14] and [12]. The underlying problem that was solved and led to compromising the protocol was the tropical one-sided discrete logarithm. Subsequently, several alternative tropical protocols have been proposed that do not rely on this problem or matrix commutativity. Notably, one such protocol is based on the tropical algebra of pairs, as presented by Ahmed, Pal, and Mohan [1].

The main ideas of the present paper are to formulate the problem which we call the "tropical two-sided discrete logarithm with shift" and present some heuristic methods of solving it, and then, based on a reduction of the matrix algebra over the tropical semiring of pairs to the usual tropical matrix algebra, develop some attacks on the above mentioned protocol of Ahmed, Pal, and Mohan [1] and demonstrate their efficiency and success rate. More specifically, the paper is organized as follows. In Section 2 we start with some preliminaries and basic definitions, particularly those related to tropical matrix periodicity. In Section 3 we present the tropical two-sided discrete log with shift problem and two heuristic algorithms to solve it, showing their efficiency and success rate. In Section 4 we recall the tropical semiring of pairs and the implementation of Stickel protocol suggested in [1]. In Section 5 we cryptanalyze this implementation using the solution of the tropical two-sided discrete log with shift problem, and present some numerical experiments showing the efficiency and performance of the attacks.

2 Preliminaries

In this section, we present some of the standard definitions in tropical matrix algebra, for most of this part closely following Butkovič [3]. Note that we use the standard notation $[m] = \{1, \ldots, m\}$ and $[n] = \{1, \ldots, n\}$ for most common index sets.

Definition 2.1 (Tropical Semiring and Tropical Matrix Algebra)**.** We define the *tropical/max-plus semiring* as $\mathbb{R}_{\max} = (\mathbb{R} \cup \{-\infty\}, \oplus, \otimes)$, where traditional addition $+$ and multiplication $\times$ are replaced by tropical addition $\oplus$ and tropical multiplication $\otimes$ respectively. These new arithmetical operations are defined by $x \oplus y = \max\{x, y\}$ and $x \otimes y = x + y$ for all $x, y \in \mathbb{R}_{\max}$.
The tropical arithmetic operations are naturally extended to include matrices and vectors. In particular, the operation $A \otimes \alpha = \alpha \otimes A, \quad$ where $\alpha \in \mathbb{R}_{\max}, A \in \mathbb{R}_{\max}^{m\times n}$ and $(A)_{ij} = a_{ij}$ for $i \in [m]$ and $j \in [n]$, is defined by

$$(A \otimes \alpha)_{ij} = (\alpha \otimes A)_{ij} = \alpha \otimes a_{ij} \quad \forall i \in [m] \text{ and } \forall j \in [n].$$

The *tropical addition* $A \oplus B$ of two matrices $A \in \mathbb{R}_{\max}^{\mathrm{mxn}}$ and $B \in \mathbb{R}_{\max}^{m\times n}$, where $(A)_{ij} = a_{ij}$ and $(B)_{ij} = b_{ij}$ for $i \in [m]$ and $j \in [n]$, is defined by

$$(A \oplus B)_{ij} = a_{ij} \oplus b_{ij} \quad \forall i \in [m] \text{ and } \forall j \in [n].$$

The *tropical multiplication* of two matrices is also similar to the "traditional" algebra. Namely, we define $A \otimes B$ for two matrices, where $A \in \mathbb{R}_{\max}^{mxp}$ and $B \in \mathbb{R}_{\max}^{p\times n}$, as follows:

$$(A \otimes B)_{ij} = \bigoplus_{k=1}^{p} a_{ik} \otimes b_{kj} = (a_{i1} \otimes b_{1j} \oplus a_{i2} \otimes b_{2j} \oplus \ldots \oplus a_{in} \otimes b_{nj}) \quad \forall i \in [m] \text{ and } \forall j \in [n].$$

Note that, despite introducing this tropical arithmetic, we will also quite often utilize the usual arithmetical operations to introduce concepts and explain arguments.

Definition 2.2 (Tropical Matrix Powers)**.** For $M \in \mathbb{R}_{\max}^{n\times n}$, the n-th *tropical power* of M is denoted by $M^{\otimes n}$, and is equal to

$$M^{\otimes n} = \underbrace{M \otimes M \otimes \ldots \otimes M}_{n \text{ times}}$$

We now introduce elements of graph theory to define the upcoming concepts. We begin by defining the directed graph (or digraph) associated with A and recalling the definitions of cycles, strongly connected components and other related concepts.

Definition 2.3 (Digraphs, Walks and Cycles)**.** The *digraph associated with* $A \in \mathbb{R}_{\max}^{n\times n}$, where $(A)_{ij} = a_{ij}$ for $i, j \in [n]$, is the pair $G(A) = (N_A, E_A)$, where $N_A = [n]$ is called the set of *nodes* of $G(A)$ and $E_A = \{(i,j) \colon a_{ij} \neq -\infty\}$ is called the set of *arcs* of $G(A)$.
A *walk* on G_A can be defined as a sequence of nodes $(i_1, \ldots, i_m)$ where each (i_l, i_{l+1}) for $l = 1, \ldots, m-1$ is an arc. A *closed walk* is a walk that starts and finishes at the same node, and a *cycle* is any closed walk that does not contain any repeated nodes, except for the beginning node and the end node.

The tropical analogue of irreducible matrix can be now defined (closely following its classical prototype).

Definition 2.4 (Irreducibility)**.** A matrix $A \in \mathbb{R}_{\max}^{n\times n}$ is called *irreducible* if G_A is strongly connected, i.e, if for each $i, j \in [n]$ there exists a walk on G_A whose starting node is i and end node is j. A is called *reducible* if it is not irreducible.

We now introduce the concept of a maximum cycle mean, which is the value of the cycle that has the greatest average weight. We then also introduce a special subgraph of $G(A)$ called the critical digraph of A. Formal definitions are given below.

Definition 2.5 (Maximum Cycle Mean and Critical Digraph)**.** For $A \in \mathbb{R}_{\max}^{n\times n}$, the *maximum cycle mean* $\lambda(A)$ is defined in the usual notation as

$$\lambda(A) = \max_{k} \max_{i_1,\ldots,i_k \in [n]} \frac{a_{i_1 i_2} + \ldots + a_{i_k i_1}}{k},$$

which is the same as

$$\lambda(A) = \bigoplus_{k} \bigoplus_{i_1,\ldots,i_k \in [n]} \sqrt[\otimes k]{a_{i_1 i_2} \otimes \ldots \otimes a_{i_k i_1}}$$

in the tropical notation.
The *critical graph* of A, denoted by $G_A^c = (N_A^c, E_A^c)$ is the subgraph of G_A which consists of all nodes and arcs of G_A that belong to the critical cycles. The nodes belonging to N_A^c and the arcs belonging to E_A^c are also called *critical.*

The matrix inverses in tropical algebra can be defined only for a limited class of matrices, but the analogue of $(I - A)^{-1}$ is more convenient to define.

Definition 2.6 (Kleene Star and Tropical Identity Matrix)**.** Let $M \in \mathbb{R}_{\max}^{n\times n}$ have $\lambda(M) \leq 0$. The *Kleene star* of M is defined by

$$M^* = I \oplus M \oplus M^{\otimes 2} \oplus \ldots \oplus M^{\otimes(n-1)},$$

where I is the *tropical identity matrix*, whose all diagonal entries are equal to 0 and all off-diagonal entries are equal to $-\infty$.

One of the key ideas of solving the discrete logarithm problems (or, more specifically, Problems 1 – 4 formulated below) is to use the ultimate periodicity properties of tropical matrix powers. Strictly speaking, 1) the ultimate periodicity (in the sense of Theorem 2.1 below) occurs with a certain shift, that is, instead of repeating themselves the powers get multiplied in the tropical sense by some scalar, 2) the ultimate periodicity does not occur for general square matrix A, but is guaranteed only under certain conditions which (at their best) only slightly generalize the irreducibility property. For simplicity, the ultimate periodicity theorem is stated below only for the irreducible case, close to the original formulation in Cohen et al. [4].

Theorem 2.1 (Ultimate Periodicity of Tropical Matrix Powers [4])**.** *Let $A \in \mathbb{R}_{\max}^{n\times n}$ be irreducible and let $\lambda = \lambda(A)$ be the maximum cycle mean of A. Then for some natural numbers T_A and γ we have*

$$A^{\otimes(t+\gamma)} = \lambda^{\otimes\gamma} \otimes A^{\otimes t} = \gamma \cdot \lambda + A^{\otimes t} \quad \forall t \geq T_A \tag{1}$$

Below we will also use the more particular ultimate periodicity of the ith columns and the ith rows:

$$A_{\cdot i}^{\otimes(t+\gamma)} = \lambda^{\otimes\gamma} \otimes A_{\cdot i}^{\otimes t} = \gamma \cdot \lambda + A_{\cdot i}^{\otimes t} \quad \forall t \geq (T_A)^i, \tag{2}$$

$$A_{i\cdot}^{\otimes(t+\gamma)} = \lambda^{\otimes\gamma} \otimes A_{i\cdot}^{\otimes t} = \gamma \cdot \lambda + A_{i\cdot}^{\otimes t} \quad \forall t \geq (T_A)_i, \tag{3}$$

The ultimate periodicity gives rise to a number of important concepts, of which we will define the following one.

Definition 2.7 (Periodicity Transients)**.** The smallest integer T_A for which (1) holds or, respectively, the smallest integers $(T_A)^i$ and $(T_A)_i$ for which (2) and (3) hold, is called the *periodicity transient* of (the tropical matrix powers of) A or, respectively, the periodicity transient of the ith column and the ith row of those powers.

Obviously, the periodicity transients of ith row and ith column can be much smaller than the periodicity transient of A, and this is particularly relevant in the case where $i \in N_A^c$ (i.e., when i is a critical node of A). It is also known that, for $i \in N_A^c$, the properties (2) and (3) take place also for general reducible matrices A. Below we will recall a particularly useful result due to Nachtigall [13].

Theorem 2.2 (Ultimate Periodicity of Critical Rows and Columns [13])**.** *Let $A \in \mathbb{R}^{n\times n}$ and k be a critical node on a critical cycle Z of length l_Z. Then $(T_A)_k \leq (n-1)\cdot l_Z$ and $(T_A)^k \leq (n-1)\cdot l_Z$.*

To explain how to compute the critical rows and columns in the ultimately periodic regime, let us also present the following result, which is a variation of the weak CSR theorems of [9] and can be seen as a slight simplification of [12], Proposition 2.5.

Theorem 2.3 (Weak CSR Expansion [9])**.** *Let $A \in \mathbb{R}_{\max}^{n\times n}$ have $\lambda = \lambda(A) \neq -\infty$, and let Z be a critical cycle of A with length l_Z. Then for some integer T_{weak} we have*

$$\begin{aligned} A^{\otimes t} &= \lambda^{\otimes t} \otimes \left(C_Z \otimes S_Z^{\otimes t} \otimes R_Z\right) \oplus B_Z^{\otimes t} \quad \forall t \geq T_{weak} \\ &= \lambda^{\otimes t} \otimes \left(C_Z \otimes S_Z^{\otimes t(\operatorname{rem} l_Z)} \otimes R_Z\right) \oplus B_Z^{\otimes t} \quad \forall t \geq T_{weak}. \end{aligned}$$

where $t\,(\operatorname{rem} l_Z)$ is the remainder when t is divided by l_Z and C_Z, S_Z, R_Z and B_Z are defined by

$$(C_Z)_{ij} = \begin{cases} (U_Z)_{ij} & \text{if } j \text{ is in } Z \\ -\infty & \text{otherwise} \end{cases}$$

$$(R_Z)_{ij} = \begin{cases} (U_Z)_{ij} & \text{if } i \text{ is in } Z \\ -\infty & \text{otherwise} \end{cases}$$

$$(S_Z)_{ij} = \begin{cases} (a_{ij} \otimes \lambda^{-1}) & \text{if } (i,j) \text{ is in } Z \\ -\infty & \text{otherwise} \end{cases}$$

$$(B_Z)_{ij} = \begin{cases} -\infty \text{ if } i \in Z \text{ or } j \in Z \\ a_{ij} \quad \text{otherwise} \end{cases}$$

where $U_Z = \left((A \otimes \lambda^{-1})^{\otimes l_Z}\right)^*$ (i.e., the Kleene star of $\left((A \otimes \lambda^{-1})^{\otimes l_Z}\right)$).

Combining Theorem 2.2 with Theorem 2.3 as well as with [15] Corollary 3.7, we obtain the following result, which we will be using below in some algorithms:

Theorem 2.4 (CSR Formula for Critical Rows and Columns)**.** *Let $A \in \mathbb{R}_{\max}^{n\times n}$ have $\lambda = \lambda(A) \neq -\infty$, and let Z be a critical cycle of A with length l_Z. Then for any $i \in Z$ we have*

$$A_{i\cdot}^{\otimes t} = \lambda^{\otimes t} \otimes (S_Z^{\otimes t(\operatorname{rem} l_Z)} R_Z)_{i\cdot},\ A_{\cdot i}^{\otimes t} = \lambda^{\otimes t} \otimes (C_Z \otimes S_Z^{\otimes t(\operatorname{rem} l_Z)})_{\cdot i},\ \forall t \geq (n-1)l_Z.$$

This theorem can be utilized to solve the different forms of tropical discrete logarithm problems, which are summarized below.

Problem 1. Given $U, M, D \in \mathbb{R}_{\max}^{n\times n}$ such that $U = M \otimes D^{\otimes t}$ for some $t \in \mathbb{N}$, find this t.

Problem 2. Given $U, M, D \in \mathbb{R}_{\max}^{n\times n}$, such that $U = \alpha \otimes M \otimes D^{\otimes t}$ for some $t \in \mathbb{N}$ and $\alpha \in \mathbb{R}_{\max}$, find τ and t' such that $U = \tau \otimes M \otimes D^{\otimes t'}$.

Solution to Problem 1 was discussed in [12] and further in [10], and Problem 2 can be solved by similar methods. However, since it involves a scalar, Problem 2 typically has an infinite number of solutions, unlike Problem 1, whose solution is typically unique. Also note that one can pose a version of Problem 1, respectively Problem 2, with $D^{\otimes t} \otimes M$ instead of $M \otimes D^{\otimes t}$, which can be regarded as a transposition of Problem 1, respectively Problem 2, and can be solved similarly.

Let us now pose two other problems, which will be referred to as tropical two-sided discrete logarithm problems.

Problem 3. Given $D_1, D_2, M, U \in \mathbb{R}_{\max}^{n\times n}$ such that $U = D_1^{\otimes t_1} \otimes M \otimes D_2^{\otimes t_2}$ for some $t_1, t_2 \in \mathbb{N}$. Find t_1', t_2' such that $U = D_1^{\otimes t_1'} \otimes M \otimes D_2^{\otimes t_2'}$.

Problem 4. Given $D_1, D_2, M, U \in \mathbb{R}_{\max}^{n\times n}$ such that $U = \alpha \otimes D_1^{\otimes t_1} \otimes M \otimes D_2^{\otimes t_2}$ for some $t_1, t_2 \in \mathbb{N}$ and $\alpha \in \mathbb{R}$. Find t_1', t_2' such that $U = \tau \otimes D_1^{\otimes t_1'} \otimes M \otimes D_2^{\otimes t_2'}$, where $\tau \in \mathbb{R}_{\max}$.

An attempt to solve Problem 3 can be found in [10], where it is observed, in particular, that the solution to this problem is in general non-unique. However, the number of solutions to this problem (unlike the number of solutions to Problem 4 is typically finite).

It is natural to solve the problems listed above by exploiting the ultimate periodicity properties of the tropical matrix powers or, in particular, by using the more precise information about those powers in the ultimate periodic regime as given, e.g., by the CSR decomposition. Let us note, however, that this approach has some limitations. Firstly, tropical matrix powers are not ultimately periodic in general (see, e.g., [3]), and both Alice and Bob can opt to use matrices whose powers are not ultimately periodic, which offers them some protection against attacks based on ultimate periodicity of entire tropical matrix powers. Secondly, the periodicity transient of tropical matrix powers depends on the matrix entries and it can be quite high.

In view of the above observations it looks more reasonable to exploit the ultimate periodicity of those columns and rows of tropical matrix powers whose indices correspond to the critical nodes. These columns and rows are ultimately periodic for any matrix and some of the known bounds on their periodicity transients are quadratic or even linear, if such columns and rows belong to a critical cycle of a known length.

This was the approach adopted in [12] and [10] and the same approach will be developed in the next section, see Algorithm 2 and Algorithm 4. The more "naive" approach, based on the assumption that the tropical matrix powers are ultimately periodic, is more relevant to Algorithm 1 and Algrorithm 3.

3 Heuristic attacks on the tropical two-sided discrete logarithm with shift

In this section, we will discuss two approaches to how Problem 4 can be solved. Let us first note that this problem differs from the tropical two-sided discrete logarithm problem (Problem 3) since it includes a coefficient. The introduction of this coefficient enhances the solvability of the problem since it becomes sufficient to find a pair of exponents t_1',t_2' that makes U a shifted version of $D_1^{\otimes t_1'} \otimes M \otimes D_2^{\otimes t_2'}$. For this reason, if there is a pair (t_1', t_2') satisfying this equation, where t_1', respectively t_2', are above the ultimate periodicity thresholds for D_1, respectively D_2, then there are infinitely many such pairs. The first algorithm, which we are going to propose, will utilize this property, since we are relying on the intuition that the smallest pair of exponents t_1' and t_2' satisfying the equation is small enough, which is true if the sequences of matrix powers of D_1 and D_2 are ultimately periodic in the sense of Theorem 2.1 and if the threshold of their ultimate periodicity is small enough.

Algorithm 1 has a perfect success rate in solving Problem 4 when $\max_t$ is equal to the maximum exponent that could be used in the problem, but it is also expected to have a high

Algorithm 1 Solving the tropical two-sided discrete logarithm with shift using the non-uniqueness of exponents

```
Input: $U, D_1, M, D_2, \max_t$
Output: $t_1', t_2', \tau$
for $t_1 = 0$ to $\max_t$ do
    for $t_2 = 0$ to $\max_t$ do
        if $(U - D_1^{\otimes t_1} \otimes M \otimes D_2^{\otimes t_2})_{ij} = \beta \ \forall i, j \in [n]$ for some $\beta \in \mathbb{R}$ then
            $t_1' = t_1$, $t_2' = t_2$, $\tau = \beta$.
```

success rate for lower values of $\max_t$ for the reasons outlined above.

For the second algorithm, we apply Theorem 2.4 for the powers of D_1 and D_2 to solve Problem 4. Note that here we can find t_1', t_2' by only searching in a search space of a size equal to the product of the two critical cycle lengths of the two matrices D_1 and D_2, while the previous algorithm has a maximum search space equal to the product of the maximum exponents that could be used in the problem (or, to be more precise, our estimates of such maximum exponents).

Note that this algorithm is heuristic in nature in particular since the CSR formulas of Theorem 2.4 are only guaranteed to hold when the exponent is larger than $(n-1) \cdot \max(l_Z, l_W)$ where l_Z and, respectively, l_W are the lengths of the critical cycle Z of D_1 and, respectively, the critical cycle W of D_2. The algorithm might not solve the problem if the original exponents t_1, t_2 are lower than this bound. One might add to the algorithm some parts where the exponents lower than this bound are also checked, similarly to the algorithms in [12] and [10].

To justify and explain the second algorithm we observe that we can use the CSR formulas of Theorem 2.4 for the rows of powers of D_1 with indices in Z and for the columns of powers of D_2 with indices in W, and then focus on the submatrix of U extracted from the rows with indices in Z and columns with indices in W. We then obtain the following equation

$$(U)_{ij} = \left(\alpha \otimes \lambda_1^{\otimes t_1} \otimes \lambda_2^{\otimes t_2} \otimes \left(S_Z^{\otimes t_1 \operatorname{rem}(l_Z)} \otimes R_Z \otimes M \otimes C_W \otimes S_W^{\otimes t_2 \operatorname{rem}(l_W)} \right)_{ij} \right.$$
$$\forall (i,j) \text{ where } i \in Z \text{ and } j \in W \text{ and } \ \forall t_1 \geq (n-1) l_Z \text{ and } \forall t_2 \geq (n-1) l_W,$$

where λ_1,λ_2 denote the maximum cycle means of the two matrices.

Then we want to solve the problem of finding (t_1', t_2', τ) for some $t_1', t_2' \in \mathbb{N}$ and $\tau \in \mathbb{R}$ such that

$$(U)_{ij} = \left(\tau \otimes \lambda_1^{\otimes t_1'} \otimes \lambda_2^{\otimes t_2'} \otimes \left(S_Z^{\otimes t_1' \operatorname{rem}(l_Z)} \otimes R_Z \otimes M \otimes C_W \otimes S_W^{\otimes t_2' \operatorname{rem}(l_W)} \right)_{ij} \right.$$
$$\forall (i,j) \text{ where } i \in Z \text{ and } j \in W \text{ and } \ \forall t_1' \geq (n-1) l_Z \text{ and } \forall t_2' \geq (n-1) l_W,$$

and this is achieved by firstly finding a pair of exponents $(t_1' \operatorname{rem}(l_Z), t_2' \operatorname{rem}(l_W))$ among $l_Z \cdot l_W$ possibilities such that it will make the above described submatrix of U and the same submatrix of $S_Z^{\otimes t_1' \operatorname{rem}(l_Z)} \otimes R_Z \otimes M \otimes C_W \otimes S_W^{\otimes t_2' \operatorname{rem}(l_W)}$ "in phase".

We will denote $t_1' \operatorname{rem}(l_Z), t_2' \operatorname{rem}(l_W)$ by $\bar{t}_1, \bar{t}_2$ respectively. In particular, we begin by finding $\bar{t}_1 \in \{1, 2, \ldots, l_Z\}$ and $\bar{t}_2 \in \{1, 2, \ldots, l_W\}$ that makes the submatrix of U extracted from the rows with indices in Z and the columns with indices in W a shifted version of the same submatrix of $\left(S_Z^{\otimes \bar{t}_1} \otimes R_Z \otimes M \otimes C_W \otimes S_W^{\otimes \bar{t}_2}\right)_{ij}$. Then we try to find (t_1', t_2', τ) by solving the following equation

$$\begin{aligned} &\beta = \tau + t_1' \cdot \lambda_1 + t_2' \cdot \lambda_2, \text{ where} \\ &\beta = \left(U - S_Z^{\otimes \bar{t}_1} \otimes R_Z \otimes M \otimes C_W \otimes S_W^{\otimes \bar{t}_2}\right)_{ij} \quad \forall (i,j)\colon i \in Z,\ j \in W. \end{aligned} \tag{4}$$

As it follows from numerical experiments, it is also reasonable to impose some lower bounds on t_1' and t_2' which guarantee that the submatrix of U with rows in Z and columns in W is in the ultimately periodic regime.

We now formulate (4) (with the above mentioned lower bounds on t_1' and t_2') in a more precise way as a mixed-integer linear programming problem. Recall that $\bar{t}_1$ and $\bar{t}_2$ denote the remainders when t_1', respectively t_2' are divided by l_Z and, respectively, l_W. Thus $t_1' = l_Z \cdot x + \bar{t}_1$ and $t_2' = l_W \cdot y + \bar{t}_2$ where $x, y \in \mathbb{N}$. We obtain

$$\beta = \tau + (l_Z \cdot x + \bar{t}_1) \cdot \lambda_1 + (l_W \cdot y + \bar{t}_2) \cdot \lambda_2,$$

which can be rearranged to give

$$\beta - \tau - \lambda_1 \cdot \bar{t}_1 - \lambda_2 \cdot \bar{t}_2 = \lambda_1 \cdot l_Z \cdot x + \lambda_2 \cdot l_W \cdot y$$

We also impose the lower bounds $t_1' \geq (n-1)l_Z$ and $t_2' \geq (n-1)l_W$ since we are counting on the critical rows and columns reaching the ultimate periodic regime, which may not be the case if these inequalities do not hold. We then have

$$t_1' = l_Z \cdot x + \bar{t}_1 \geq (n-1)l_Z \quad \text{and} \quad t_2' = l_W \cdot y + \bar{t}_2 \geq (n-1)l_W,$$

which we can rearrange to obtain

$$x \geq \frac{(n-1)l_Z - \bar{t}_1}{l_Z} \quad \text{and} \quad y \geq \frac{(n-1)l_W - \bar{t}_2}{l_W}$$

Then we can find solutions (t_1', t_2', τ) by solving

$$\begin{cases} \beta - \tau - \lambda_1 \cdot \bar{t}_1 - \lambda_2 \cdot \bar{t}_2 = (\lambda_1 \cdot l_Z)x + (\lambda_2 \cdot l_W)y \\ x \geq \frac{(n-1)l_Z - \bar{t}_1}{l_Z} \\ y \geq \frac{(n-1)l_W - \bar{t}_2}{l_W} \end{cases} \tag{5}$$

for $(x, y) \in \mathbb{N}^2$ and $\tau \in \mathbb{R}$. Note that this is a mixed integer linear programming problem with unknowns x, y, τ. We are now ready to formulate the solution method, see Algorithm 2.

The following numerical experiments show the success rate and time consumption for Algorithm 1 and Algorithm 2 as a function of matrix dimension. For all experiments, the entries of the matrices are random integers in $[-1000, 1000]$, and 100 trials were performed for each dimension.

Algorithm 2 Solving the tropical two-sided discrete logarithm with shift using CSR

```
Input: $U, D_1, M, D_2$
Output: $t'_1, t'_2, \tau$
Calculate $\lambda(D_1) = \lambda_1, \lambda(D_2) = \lambda_2$
Find a critical cycle $Z$ from $D_1$, and $W$ from $D_2$, let their lengths be $l_Z$ and $l_W$, respectively.
Calculate $S_Z$, $R_Z$, $C_W$ and $S_W$ as in Theorem 2.3
for $\bar{t}_1 = 0$ to $l_Z$ do
    for $\bar{t}_2 = 0$ to $l_W$ do
        if $(U - (S_Z^{\otimes \bar{t}_1} \otimes R_Z \otimes M \otimes C_W \otimes S_W^{\bar{t}_2})_{ij} = \beta$ for some $\beta \in \mathbb{R}$ and for all $i, j$ where $i \in Z$ and $j \in W$ then
            Check if (5) is solvable. If it is, then return $(t'_1, t'_2, \tau)$ where $t'_1 = l_Z \cdot x + \bar{t}_1$ and $t'_2 = l_W \cdot y + \bar{t}_2$.
```

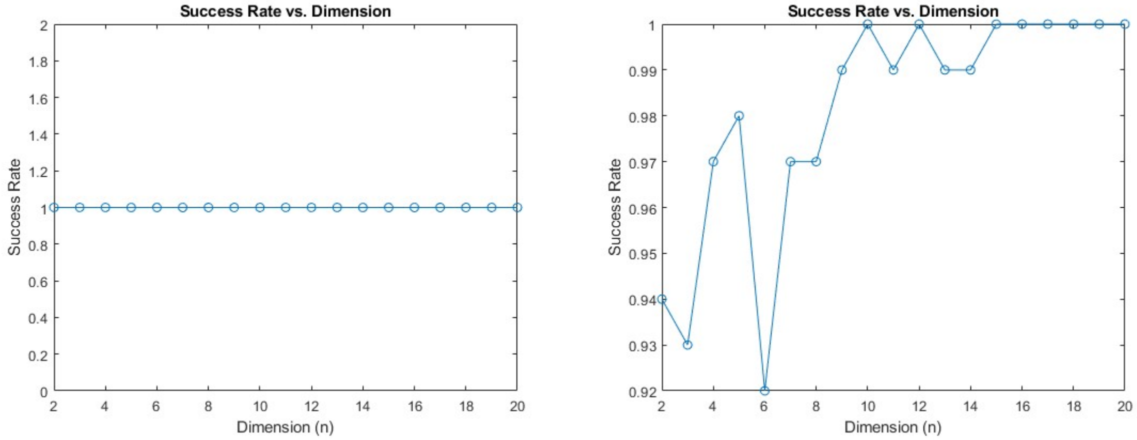

Figure 1: Algorithm 1 success rate with $\max_t = n^5$ (left) and $\max_t = n^3$ (right)

Success rate for Algorithm 1 is shown in Figure 1. The exponents t_1, t_2 in Problem 4 are randomly chosen among integers in $[1, n^5]$, and the scalar value α is in $[1, 1000]$. Algorithm 1 parameter $\max_t$ is n^5 for the left hand side of the figure and n^3 for the right hand side of the figure.

We notice that Algorithm 1 never fails when the maximum searchable exponent $\max_t$ is the same as the one used in Problem 4 since the algorithm searches for all possible matrix exponents that make U in phase or equal to $D_1^{\otimes t'_1} \otimes M \otimes D_2^{\otimes t'_2}$. The guaranteed success results from testing all potential exponent combinations. The efficiency and quickness of the algorithm in finding appropriate exponents depends on the cyclicity pattern of the matrices D_1 and D_2. When we limit $\max_t$ to a lower exponent, we lose the guaranteed success rate, but we still achieve a high success rate with a faster time.

Success rate for Algorithm 2 is shown in Figure 2. The exponents t_1, t_2 in Problem 4 are allowed to be less than $(n-1)l_Z$ and $(n-1)l_W$ for the left hand side, and are required to be larger than $(n-1)l_Z$ and $(n-1)l_W$ for the right hand side, and the scalar value α is in the interval $[1, 1000]$.

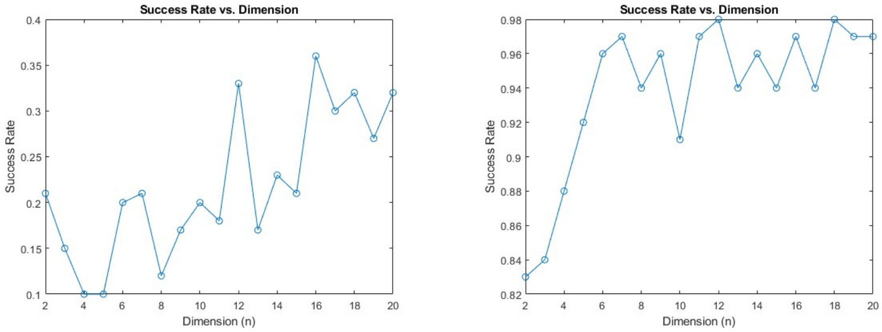

Figure 2: Algorithm 2 success rate when $t_1 < (n-1)l_Z$ and $t_2 < (n-1)l_W$ (left), and when $t_1 \geq (n-1)l_Z$ and $t_2 \geq (n-1)l_W$ (right)

We notice that Algorithm 2 has a high success rate when the exponents used in Problem 4 are larger than $(n-1)l_Z$ and $(n-1)l_W$ for t_1 and t_2 respectively since the critical entries are guaranteed to enter the ultimately periodic regime after such bounds. The attack however does not perform so well when the used exponents are lower than this threshold which is expected since the CSR formulas do not necessarily hold for such exponent values (i.e., entries in the critical rows or columns have not necessarily entered the ultimate periodicity).

The time consumption for the two algorithms is quite different, as shown in Figure 3. Here, the exponents t_1, t_2 in Problem 4 are random integers larger than $(n-1)l_Z$ and $(n-1)l_W$ respectively, and the scalar value α is in the interval $[1, 1000]$.

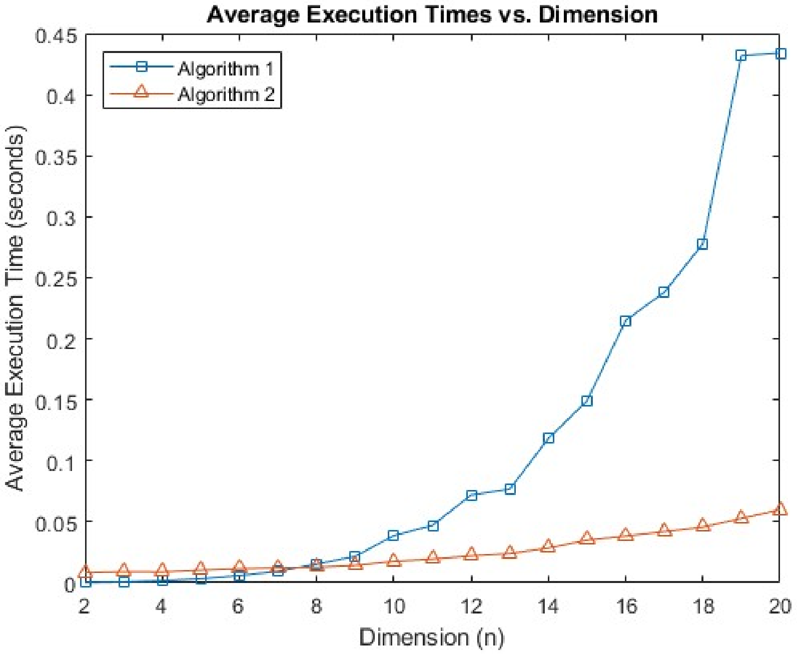

Figure 3: Time taken for Algorithm 1 and Algorithm 2

We notice that Algorithm 2 is faster than Algorithm 1 since it limits its search to a finite set of values equal to the product of the critical cycle sizes of the two matrices D_1, D_2.

However, for lower values of n, we note that Algorithm 1 is faster since the search space is small and Algorithm 2 requires more steps such as computing the CSR terms and maximum cycle means.

Note that one can combine the two algorithms to get better overall performance in terms of success rate and execution time. One possible combination is to apply Algorithm 1 for t_1 until $(n-1)l_Z$ and t_2 until a big enough number, then for t_2 until $(n-1)l_W$ and for t_1 until a big enough number, and then to perform Algorithm 2 for larger exponents.

4 The tropical semiring of pairs and the key exchange protocol

In this section, we examine a key exchange protocol proposed in [1]. The protocol employs a modified tropical structure and is claimed to resist the known attacks on conventional tropical key exchange protocols. We will begin by introducing the modified tropical structure, followed by presenting the associated protocol and outlining its construction.

4.1 The tropical semiring of pairs

Let $\mathbb{R}_{max}$ be the tropical semiring defined in 2.1, then we have the following definition.

Definition 4.1. (Tropical Algebra of Pairs [2], [1]). We define the tropical algbera of pairs $\mathbb{R}^2_{max}$ as

$$\mathbb{R}^2_{max} = (\mathbb{R}_{max} \times \mathbb{R}_{max}, \oplus', \otimes')$$

with elements of the shape $(a^{(1)}, a^{(2)})$ such that $a^{(k)} \in \mathbb{R}_{max}$ for $k \in \{1, 2\}$ denotes the first and second element of a two-dimensional vector in $\mathbb{R}^2_{max}$ with the operations $(\oplus', \otimes')$ defined as

$$(a^{(1)}, a^{(2)}) \oplus' (b^{(1)}, b^{(2)}) = (a^{(1)} \oplus b^{(1)}, a^{(2)} \oplus b^{(2)})$$
$$(a^{(1)}, a^{(2)}) \otimes' (b^{(1)}, b^{(2)}) = ((a^{(1)} \otimes b^{(1)}) \oplus (a^{(2)} \otimes b^{(2)}), (a^{(1)} \otimes b^{(2)}) \oplus (a^{(2)} \otimes b^{(1)}))$$
$$\text{such that} \quad (a^{(1)}, a^{(2)}), (b^{(1)}, b^{(2)}) \in \mathbb{R}^2_{max} \quad \text{and} \quad a^{(1)}, a^{(2)}, b^{(1)}, b^{(2)} \in \mathbb{R}_{max}$$

The following example illustrates these operations.

Example 4.1. Let $(1, 3), (2, 5) \in \mathbb{R}^2_{max}$ then

$$(1, 3) \oplus' (2, 5) = (1 \oplus 2, 3 \oplus 5) = (2, 5)$$
$$(1, 3) \otimes' (2, 5) = ((1 \otimes 2) \oplus (3 \otimes 5), (1 \otimes 5) \oplus (3 \otimes 2)) = (8, 6)$$

The operations of this semiring can also be extended to include matrices following the same manner as the conventional tropical matrix operations in Definition 2.1 but with replacing the elements of $\mathbb{R}_{max}$ by the elements of $\mathbb{R}^2_{max}$ and $(\oplus, \otimes)$ by $(\oplus', \otimes')$. Thus, we will denote the semiring of matrices over $\mathbb{R}^2_{max}$ by $\mathbb{R}^{2^{n\times n}}_{max}$.

4.2 Key exchange protocol based on the semiring of pairs

We now present the key exchange protocol proposed by the authors in [1] which is based on the tropical algebra of pairs $\mathbb{R}_{max}^{2^{n\times n}}$. Note that the authors in this protocol altered the second operation of $\mathbb{R}^2_{max}$ to be $(x,y)\otimes_c(z,w)=((c\otimes x\otimes z)\oplus(c\otimes y\otimes w),(c\otimes x\otimes w)\oplus(c\otimes y\otimes z))$ where $0\neq c\in\mathbb{Z}$. This modification constructs a new semiring $\mathbb{R}^2_{max_c}$ with multiplicative identity $(-c,-\infty)$.

Protocol 1 (Key Exchange Protocol based on the Semiring of Pairs)**.**

1. Alice and Bob agree upon two public matrices $X,Y\in\mathbb{R}_{max}^{2^{n\times n}}$.
2. Alice chooses a semiring $\mathbb{R}^2_{max_c}=(\mathbb{R}_{max}\times\mathbb{R}_{max},\oplus_c,\otimes_c)$ where

$$(a^{(1)},a^{(2)})\oplus_c(b^{(1)},b^{(2)})=(a^{(1)}\oplus b^{(1)},a^{(2)}\oplus b^{(2)})$$
$$(a^{(1)},a^{(2)})\otimes_c(b^{(1)},b^{(2)})=((c\otimes a^{(1)}\otimes b^{(1)})\oplus(c\otimes a^{(2)}\otimes b^{(2)}),(c\otimes a^{(1)}\otimes b^{(2)})\oplus(c\otimes a^{(2)}\otimes b^{(1)}))$$

   Here, $c\in\mathbb{Z}$ is her fixed private parameter. Bob also picks $\mathbb{R}^2_{max_d}=(\mathbb{R}_{max}\times\mathbb{R}_{max},\oplus_d,\otimes_d)$ where

$$(a^{(1)},a^{(2)})\oplus_d(b^{(1)},b^{(2)})=(a^{(1)}\oplus b^{(1)},a^{(2)}\oplus b^{(2)})$$
$$(a^{(1)},a^{(2)})\otimes_d(b^{(1)},b^{(2)})=((d\otimes a^{(1)}\otimes b^{(1)})\oplus(d\otimes a^{(2)}\otimes b^{(2)}),(d\otimes a^{(1)}\otimes b^{(2)})\oplus(d\otimes a^{(2)}\otimes b^{(1)}))$$

   and $d\in\mathbb{Z}$ is his fixed private parameter.
3. Alice picks two private natural numbers k, l and calculates $A=X^{\otimes_c k}\otimes Y^{\otimes_c l}$. She then chooses another private integer p and sends $A_{(p)}$ to Bob, where $A_{(p)}$ is $p\otimes A$.
4. Bob also picks his private natural numbers r,s, calculates $B=X^{\otimes_d r}\otimes Y^{\otimes_d s}$ and sends $B_{(q)}$ to Alice,where $B_{(q)}$ is $q\otimes B$ and $q\in\mathbb{Z}$ is Bob's private parameter.
5. Alice computes her key $K_{Alice}=\left(X^{\otimes_c k}\right)_{(p)}\otimes B_{(q)}\otimes Y^{\otimes_c l}$ and Bob's key is $K_{Bob}=\left(X^{\otimes_d r}\right)_{(q)}\otimes A_{(p)}\otimes Y^{\otimes_d s}$.

The two parties end up with the same key $K_{Alice}=K_{Bob}$ as proved in [1].

The following example illustrates the above key exchange protocol.

Example 4.2. Alice and Bob agree on the public matrices X and Y:

$$X=\begin{bmatrix}(9,6)&(1,-2)\\(8,-4)&(4,1)\end{bmatrix}\text{ and }Y=\begin{bmatrix}(-2,0)&(8,-2)\\(6,10)&(8,-5)\end{bmatrix}$$

and they choose the private parameters $c=3,k=27,l=18,p=-2$ for Alice, and $d=4,r=17,s=23,q=5$ for Bob.

Alice calculates $A=X^{\otimes_3 27}\otimes Y^{\otimes_3 18}=\begin{bmatrix}(532,534)&(532,530)\\(531,533)&(531,529)\end{bmatrix}$ and sends $A_{(-2)}=\begin{bmatrix}(530,532)&(530,528)\\(529,531)&(529,527)\end{bmatrix}$ to Bob.

Bob calculates $B = X^{\otimes_4 17} \otimes Y^{\otimes_4 23} = \left[\begin{array}{cc} (509,511) & (509,511) \\ (508,510) & (508,510) \end{array}\right]$ and sends $B_{(5)} = \left[\begin{array}{cc} (514,516) & (514,516) \\ (513,515) & (513,515) \end{array}\right]$ to Alice.

Then Alice and Bob compute their keys using those received transmissions:

$$K_{Alice} = \left(X^{\otimes_3 27}\right)_{(-2)} \otimes B_{(5)} \otimes Y^{\otimes_3 18} = \left[\begin{array}{cc} (1048,1046) & (1048,1046) \\ (1047,1045) & (1047,1045) \end{array}\right]$$

$$K_{Bob} = \left(X^{\otimes_4 17}\right)_{(5)} \otimes A_{(-2)} \otimes Y^{\otimes_4 23} = \left[\begin{array}{cc} (1048,1046) & (1048,1046) \\ (1047,1045) & (1047,1045) \end{array}\right]$$

Thus they end up with the same shared key.

5 Cryptanalysis of the protocol

In this section, we introduce our attacks on the proposed protocol. It is claimed that it is more resilient than its conventional tropical counterparts since it doesn't reveal a cyclicity pattern for high powers, but we will show otherwise. We begin by presenting an alternative representation of matrices in $\mathbb{R}^{2^{n\times n}}_{max}$, providing a foundation for constructing our attacks which are based on solving Problem 4. Subsequently, we implement our attacks showing their efficiency and success rate.

5.1 Representing matrices in $(\mathbb{R}^2)^{n\times n}_{\max}$ as matrices in $\mathbb{R}^{2n\times 2n}_{\max}$

Observe that a matrix with entries in the tropical semiting of pairs can be associated with a conventional tropical matrix in $\mathbb{R}^{2n\times 2n}_{\max}$ by replacing each entry of the matrix over that semiring with a 2×2 square matrix. More formally this representation is defined as follows.

Definition 5.1. Suppose $A \in (R^2_{\max})^{n\times n}$ is given by

$$A = \begin{bmatrix} (a^{(1)}_{11}, a^{(2)}_{11}) & (a^{(1)}_{12}, a^{(2)}_{12}) & \dots & (a^{(1)}_{1n}, a^{(2)}_{1n}) \\ (a^{(1)}_{21}, a^{(2)}_{21}) & (a^{(1)}_{22}, a^{(2)}_{22}) & \dots & (a^{(1)}_{2n}, a^{(2)}_{2n}) \\ \vdots & \vdots & \ddots & \vdots \\ (a^{(1)}_{n1}, a^{(2)}_{n1}) & (a^{(1)}_{n2}, a^{(2)}_{n2}) & \dots & (a^{(1)}_{nn}, a^{(2)}_{nn}) \end{bmatrix},$$

then we define $\tilde{A} \in \mathbb{R}^{2n\times 2n}_{max}$ as follows:

$$\tilde{A} = \begin{bmatrix} a^{(1)}_{11} & a^{(2)}_{11} & a^{(1)}_{12} & a^{(2)}_{12} & \dots & a^{(1)}_{1n} & a^{(2)}_{1n} \\ a^{(2)}_{11} & a^{(1)}_{11} & a^{(2)}_{12} & a^{(1)}_{12} & \dots & a^{(2)}_{1n} & a^{(1)}_{1n} \\ a^{(1)}_{21} & a^{(2)}_{21} & a^{(1)}_{22} & a^{(2)}_{22} & \dots & a^{(1)}_{2n} & a^{(2)}_{2n} \\ a^{(2)}_{21} & a^{(1)}_{21} & a^{(2)}_{22} & a^{(1)}_{22} & \dots & a^{(2)}_{2n} & a^{(1)}_{2n} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a^{(1)}_{n1} & a^{(2)}_{n1} & a^{(1)}_{n2} & a^{(2)}_{n2} & \dots & a^{(1)}_{nn} & a^{(2)}_{nn} \\ a^{(2)}_{n1} & a^{(1)}_{n1} & a^{(2)}_{n2} & a^{(1)}_{n2} & \dots & a^{(2)}_{nn} & a^{(1)}_{nn} \end{bmatrix}$$

.

We now observe that this representation gives rise to an injective homomorphism from the matrix algebra $(\mathbb{R}^2_{\max})^{n\times n}$ to the matrix algebra $\mathbb{R}^{2n\times 2n}_{\max}$. In other words, each matrix in $(\mathbb{R}^2_{\max})^{n\times n}$ is uniquely represented by a matrix in $\mathbb{R}^{2n\times 2n}_{\max}$ and this representation respects the matrix addition (obviously), and also the matrix multiplication, due to the following proposition.

Proposition 5.1. Let $A, B \in (\mathbb{R}^2)^{n\times n}_{\max}$, then $\tilde{A}\otimes\tilde{B} = \widetilde{A\otimes B}$ where $\tilde{A}, \tilde{B}, (\widetilde{A\otimes B}) \in \mathbb{R}^{2n\times 2n}_{max}$.

Proof. We infer from the definition of the multiplication over $\mathbb{R}^2_{\max}$ (Definition 4.1) that if

$$A = \begin{bmatrix} (a^{(1)}_{11}, a^{(2)}_{11}) & \dots & (a^{(1)}_{1n}, a^{(2)}_{1n}) \\ \vdots & \ddots & \vdots \\ (a^{(1)}_{n1}, a^{(2)}_{n1}) & \dots & (a^{(1)}_{nn}, a^{(2)}_{nn}) \end{bmatrix}$$

and

$$B = \begin{bmatrix} (b^{(1)}_{11}, b^{(2)}_{11}) & \dots & (b^{(1)}_{1n}, b^{(2)}_{1n}) \\ \vdots & \ddots & \vdots \\ (b^{(1)}_{n1}, b^{(2)}_{n1}) & \dots & (b^{(1)}_{nn}, b^{(2)}_{nn}) \end{bmatrix}$$

Then we have

$$(A\otimes B)_{pq} = \left(\bigoplus_{k=1}^{n} a^{(1)}_{pk}\otimes b^{(1)}_{kq} \oplus a^{(2)}_{pk}\otimes b^{(2)}_{kq}, \bigoplus_{k=1}^{n} a^{(1)}_{pk}\otimes b^{(2)}_{kq} \oplus a^{(2)}_{pk}\otimes b^{(1)}_{kq}\right)$$
$$(A\otimes B)_{pq} = (z^{(1)}_{pq}, z^{(2)}_{pq}) \quad \forall p, q \in [n]$$

where $z^{(1)}_{pq} = \bigoplus_{k=1}^{n} a^{(1)}_{pk}\otimes b^{(1)}_{kq} \oplus a^{(2)}_{pk}\otimes b^{(2)}_{kq}$ and $z^{(2)}_{pq} = \bigoplus_{k=1}^{n} a^{(1)}_{pk}\otimes b^{(2)}_{kq} \oplus a^{(2)}_{pk}\otimes b^{(1)}_{kq}$. Using Definition 5.1 to obtain matrices $\tilde{A}$, $\tilde{B}$ and, respectively, $(\widetilde{A\otimes B})$, we then observe that

$$(\tilde{A}\otimes\tilde{B})_{ij} = \bigoplus_{k=1}^{n} a^{(1)}_{pk}\otimes b^{(1)}_{kq} \oplus a^{(2)}_{pk}\otimes b^{(2)}_{kq} = Z^{(1)}_{ij} \quad \forall i, j \in [2n] \text{ such that} (i+j) \mod 2n = 0$$

and

$$(\tilde{A}\otimes\tilde{B})_{ij} = \bigoplus_{k=1}^{n} a^{(1)}_{pk}\otimes b^{(2)}_{kq} \oplus a^{(2)}_{pk}\otimes b^{(1)}_{kq} = Z^{(2)}_{ij} \quad \forall i, j \in [2n] \text{ such that} (i+j) \mod 2n \neq 0.$$

Therefore

$$\tilde{A}\otimes\tilde{B} = \begin{bmatrix} z^{(1)}_{11} & z^{(2)}_{11} & \dots & z^{(1)}_{1n} & z^{(2)}_{1n} \\ z^{(2)}_{11} & z^{(1)}_{11} & \dots & z^{(2)}_{1n} & z^{(1)}_{1n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ z^{(1)}_{n1} & z^{(2)}_{n1} & \dots & z^{(1)}_{nn} & z^{(2)}_{nn} \\ z^{(2)}_{n1} & z^{(1)}_{n1} & \dots & z^{(2)}_{nn} & z^{(1)}_{nn} \end{bmatrix} = (\widetilde{A\otimes B})$$

. □

Example 5.1. Suppose we have

$$A = \begin{bmatrix} (-7,-1) & (6,2) \\ (-4,9) & (4,-8) \end{bmatrix} \text{ and } B = \begin{bmatrix} (2,2) & (-6,7) \\ (1,10) & (3,-2) \end{bmatrix}$$

Then

$$A \otimes B = \begin{bmatrix} (-7,-1) & (6,2) \\ (-4,9) & (4,-8) \end{bmatrix} \otimes \begin{bmatrix} (2,2) & (-6,7) \\ (1,10) & (3,-2) \end{bmatrix} = \begin{bmatrix} (12,16) & (9,5) \\ (11,14) & (16,3) \end{bmatrix}$$

The above operation can be equivalently calculated using the conventional tropical matrix algebra, as we have

$$\tilde{A} = \begin{bmatrix} -7 & -1 & 6 & 2 \\ -1 & -7 & 2 & 6 \\ -4 & 9 & 4 & -8 \\ 9 & -4 & -8 & 4 \end{bmatrix}, \quad \tilde{B} = \begin{bmatrix} 2 & 2 & -6 & 7 \\ 2 & 2 & 7 & -6 \\ 1 & 10 & 3 & -2 \\ 10 & 1 & -2 & 3 \end{bmatrix}$$

$$\begin{aligned} \tilde{A} \otimes \tilde{B} &= \begin{bmatrix} -7 & -1 & 6 & 2 \\ -1 & -7 & 2 & 6 \\ -4 & 9 & 4 & -8 \\ 9 & -4 & -8 & 4 \end{bmatrix} \otimes \begin{bmatrix} 2 & 2 & -6 & 7 \\ 2 & 2 & 7 & -6 \\ 1 & 10 & 3 & -2 \\ 10 & 1 & -2 & 3 \end{bmatrix} \\ &= \begin{bmatrix} 12 & 16 & 9 & 5 \\ 16 & 12 & 5 & 9 \\ 11 & 14 & 16 & 3 \\ 14 & 11 & 3 & 16 \end{bmatrix} = (\widetilde{A \otimes B}) \end{aligned}$$

Notice that the odd numbered rows of $\tilde{A} \otimes \tilde{B}$ are identical to the rows of $A \otimes B$. We will exploit this observation to attack Protocol 1.

5.2 Attacks on Protocol 1 based on solving the tropical two-sided discrete logarithm

After the suggested matrix transformation, we observe that the problem of attacking the proposed protocol is essentially reduced to solving Problem 4. Thus, to attack the protocol, it is sufficient to find a pair of exponents (k', l') that makes the tropical product of the public matrices $X^{\otimes k'} \otimes Y^{\otimes l'}$ in phase with the transmitted matrix $A_{(p)}$, and not necessarily the exact exponents that generated the instance. In particular, we know from the definition of the proposed protocol that

$$\begin{aligned} A &= X^{\otimes_c k} \otimes Y^{\otimes_c l} \\ &= (k-1)c \otimes X^{\otimes k} \otimes (l-1)c \otimes Y^{\otimes l} \end{aligned}$$

Then when the two sides are multiplied by p, we get

$$A_{(p)} = p \otimes (k-1)c \otimes X^{\otimes k} \otimes (l-1)c \otimes Y^{\otimes l}.$$

This $A_{(p)}$ can be intercepted by the attacker, and he can transform it to $\tilde{A}_{(p)} \in \mathbb{R}_{max}^{2n\times 2n}$ using Definition 5.1. Similarly, the attacker also transforms public matrices X and Y to $\tilde{X}$ and $\tilde{Y}$. The attacker can then utilize the following equality, which holds by Proposition 5.1

$$\tilde{A}_{(p)} = p \otimes (k+l-2)c \otimes \tilde{X}^{\otimes k} \otimes \tilde{Y}^{\otimes l} \tag{6}$$

This equation resembles Problem 4 which can be solved by finding a pair of exponents k', l' such that $\tilde{A}_{(p)}$ is a shifted version of $\tilde{X}^{\otimes k'} \otimes \tilde{Y}^{\otimes l'}$ by a fixed integer τ which equals to the scalar part of the above equation ($\tau = p' + (k'+l'-2)c'$). In other words, all entries of $\tilde{A}_{(p)}$ differ by the same amount from the corresponding entry of $\tilde{X}^{\otimes k'} \otimes \tilde{Y}^{\otimes l'}$. There are highly likely infinite number of solutions due to the ultimate periodicity of tropical matrices. Note that the attacker can find the private parameters p', c' by solving the equation $\tau = p' + (k'+l'-2)c'$ for p', c'. However, these parameters are not required to construct the shared key as it suffices to use τ.

The shared key can then be computed using the found exponents (k', l') and τ and the other intercepted massage $B_{(q)}$ as

$$\begin{aligned} K_{\text{Attack}} &= \tau \otimes \left(X^{\otimes k'} \otimes B_{(q)} \otimes Y^{\otimes l'}\right) \\ &= (p' + (k'+l'-2)c' + q + (r+s-2)d) \otimes \left(X^{\otimes k'} \otimes X^{\otimes r} \otimes Y^{\otimes s} \otimes Y^{\otimes l'}\right) \\ &= (p' + (k'+l'-2)c' + q + (r+s-2)d) \otimes \left(X^{\otimes r} \otimes X^{\otimes k'} \otimes Y^{\otimes l'} \otimes Y^{\otimes s}\right) \\ &= (p + (k+l-2)c + q + (r+s-2)d) \otimes \left(X^{\otimes k} \otimes X^{\otimes r} \otimes Y^{\otimes s} \otimes Y^{\otimes l}\right) \\ &= K_{\text{Alice}} = K_{\text{Bob}} \end{aligned} \tag{7}$$

The attack is described in Algorithm 3. We also present an example illustrating this attack.

Algorithm 3 Attacking Protocol 1 by using Algorithm 1

```
Input: $X, Y, A_{(p)}, B_{(q)}, \max_t$
Output: $Key$
Transform $X, Y, A_{(p)}$ to $\tilde{X}, \tilde{Y}, \tilde{A}_{(p)}$ using Definition 5.1
Perform Algorithm 1 with Input:$\tilde{A}_{(p)}, \tilde{X}, I, \tilde{Y}, \max_t$, then we get the output $k', l', \tau$
$Key = \tau \otimes \left(X^{\otimes k'} \otimes B_{(q)} \otimes Y^{\otimes l'}\right)$
```

Example 5.2. Suppose that Alice and Bob agree on the public matrices X and Y that are shown in Example 4.2, with the private parameters being $c = 3, k = 27, l = 18, p = -2$ for Alice and $d = 4, r = 17, s = 23, q = 5$ for Bob. The attacker intercepts $A_{(p)}$ and transforms it to a conventional tropical matrix $\tilde{A}_{(p)}$ in $R_{max}^{2n\times 2n}$. He also transforms the public matrices X and Y to $\tilde{X}$ and $\tilde{Y}$, respectively. Let the results of this transformation be

$$\tilde{A}_{(p)} = \begin{bmatrix} 530 & 532 & 530 & 528 \\ 532 & 530 & 528 & 530 \\ 529 & 531 & 529 & 527 \\ 531 & 529 & 527 & 529 \end{bmatrix}, \quad \tilde{X} = \begin{bmatrix} 9 & 6 & 1 & -2 \\ 6 & 9 & -2 & 1 \\ 8 & -4 & 4 & 1 \\ -4 & 8 & 1 & 4 \end{bmatrix} \quad \tilde{Y} = \begin{bmatrix} -2 & 0 & 8 & -2 \\ 0 & -2 & -2 & 8 \\ 6 & 10 & 8 & -5 \\ 10 & 6 & -5 & 8 \end{bmatrix}$$

The attacker then searches for a pair of exponents k' and l' that satisfies the following equation

$$\left(\tilde{A}_{(p)} - \tilde{X}^{\otimes k'} \otimes \tilde{Y}^{\otimes l'}\right)_{ij} = \tau \text{ for some } \tau \in \mathbb{R} \quad \forall i, j \in [2n]$$

The attacker notices that any pair from $k' \in \{2, 3, 4, \ldots\}$ and $l' \in \{10, 14, 18, \ldots\}$ satisfies the above equality. Suppose the attacker picks $k = 2$ and $l = 10$

$$\left(\tilde{A}_{(p)} - \tilde{X}^{\otimes 2} \otimes \tilde{Y}^{\otimes 10}\right)_{ij} = 424 \quad \forall i, j \in [2n]$$

Then, the shared key can be computed as:

$$\begin{aligned} \tilde{K}_{\text{Attack}} &= \tau \otimes \left(\tilde{X}^{\otimes 2} \otimes \tilde{B}_{(q)} \otimes \tilde{Y}^{\otimes 10}\right) \\ &= 424 \otimes \begin{bmatrix} 624 & 622 & 624 & 622 \\ 622 & 624 & 622 & 624 \\ 623 & 621 & 623 & 621 \\ 621 & 623 & 621 & 623 \end{bmatrix} = \begin{bmatrix} 1048 & 1046 & 1048 & 1046 \\ 1046 & 1048 & 1046 & 1048 \\ 1047 & 1045 & 1047 & 1045 \\ 1045 & 1047 & 1045 & 1047 \end{bmatrix} \end{aligned}$$

which represents

$$\begin{bmatrix} (1048, 1046) & (1048, 1046) \\ (1047, 1045) & (1047, 1045) \end{bmatrix} = K_{\text{Alice}} = K_{\text{Bob}}$$

We notice that the attacker successfully recovered the secret key using smaller exponents than those that generated the protocol instance.

We now describe the attack based on the CSR solution to the two-sided discrete logarithm problem. Starting from (6) we focus on the submatrix extracted from the rows with indices in a critical cycle Z of $\tilde{X}$ and the columns with indices in a critical cycle W of $\tilde{Y}$, for which we obtain

$$\begin{aligned} &\left(\tilde{A}_{(p)} = \tau \otimes \lambda_1^{\otimes k} \otimes \lambda_2^{\otimes l} \otimes S_Z^{\otimes k \operatorname{rem}(l_Z)} \otimes R_Z \otimes C_W \otimes S_W^{\otimes l \operatorname{rem}(l_W)}\right)_{ij} \quad \forall i \in Z, j \in W \\ &k \geq (2n-1)l_Z, \quad l \geq (2n-1)l_W. \end{aligned}$$

Here l_Z, l_W denote the lengths of Z and W, and $\tau = p \otimes (k + l - 2)c$. Then, we want to find a pair of exponents k', l' such that $(\tilde{A}_{(p)})_{ij}$ is a shifted version of $(S_Z^{\otimes k' \operatorname{rem}(l_Z)} \otimes R_Z \otimes C_W \otimes S_W^{\otimes l' \operatorname{rem}(l_W)})_{ij}$ by a fixed integer β which equals to the scalar part ($\beta = \tau \otimes \lambda_1^{\otimes k'} \otimes \lambda_2^{\otimes l'}$). To achieve that, we firstly need to find $\bar{k} = k' \operatorname{rem}(l_Z)$ and $\bar{l} = l' \operatorname{rem}(l_W) \in \{1, 2, \ldots, l_Z\} \times \{1, 2, \ldots, l_W\}$ that satisfies this shift requirement, and then find (k', l', τ) by solving the following mixed integer linear programming problem

$$\begin{cases} \beta - \tau - \lambda_1 \cdot \bar{k} - \lambda_2 \cdot \bar{l} = (\lambda_1 \cdot l_Z)x + (\lambda_2 \cdot l_W)y \\ x \geq \frac{(2n-1)l_Z - \bar{k}}{l_Z} \\ y \geq \frac{(2n-1)l_W - \bar{l}}{l_W} \end{cases} \tag{8}$$

with unknowns τ, x and y. Then we obtain $k' = x \cdot l_Z + \bar{k}$ and $l' = y \cdot l_W + \bar{l}$. If it happens that (6) holds for the pair of exponents thus found, then this method also yields a common key, by applying the same argument as in (7). The attack is formally written as Algorithm 4.

Algorithm 4 Attacking Protocol 1 by Using Algorithm 2

```
**Input:** $X, Y, A_{(p)}, B_{(q)}$
**Output:** $Key$
Transform $X, Y, A_{(p)}$ to $\tilde{X}, \tilde{Y}, \tilde{A}_{(p)}$ using Definition 5.1
Calculate $\lambda(\tilde{X}) = \lambda_1, \lambda(\tilde{Y}) = \lambda_2$
Find a critical cycle $Z$ from $\tilde{X}$, and $W$ from $\tilde{Y}$, let their lengths be $l_Z$ and $l_W$, respectively.
Calculate $S_Z$, $R_Z$, $C_W$ and $S_W$ as in Theorem 2.3
Perform Algorithm 2 with Input:$\tilde{A}_{(p)}, \tilde{X}, I, \tilde{Y}$, then we get the output $k', l', \tau$
$Key = \tau \otimes \left(X^{\otimes k'} \otimes B_{(q)} \otimes Y^{\otimes l'}\right)$
```

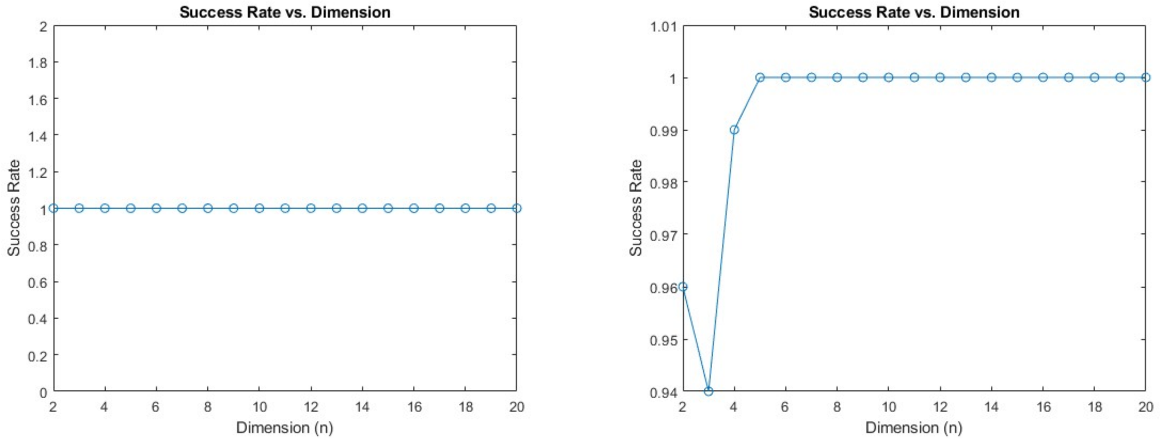

Figure 4: Attack 3 success rate with $\max_t = n^5$ and $\max_t = n^3$

5.3 Numerical experiments

Both of the suggested attacks on Protocol 1 have been tested numerically. Specifically, we looked at the success rate for each attack as a function of matrix dimension. Additionally, we analyzed the time required for the attacker to construct the secret shared key. This analysis serves as a basis for comparison with the process of original secret key generation. For all experiments, the protocol parameters p, q, c, d and the matrix entries are random integers in $[-1000, 1000]$, and 100 trials are performed for each dimension.

Success rate for Algorithm 3 is presented on Figure 4. Here, the private exponents k, l, r, s in the protocol are random integers in $[1, n^5]$, where n is the dimension of the matrices, and the parameter $\max_t$ is n^5 (on the left) and n^3 (on the right).

Since Algorithm 3 uses Algorithm 1, it never fails when the maximum searchable exponent $\max_t$ is the same as the one used in the protocol since the algorithm searches for all possible matrix exponents that make $A_{(P)}$ in phase or equal to $X^{\otimes k} \otimes Y^{\otimes l}$. The guaranteed success results from testing all potential exponent combinations. The efficiency and quickness of the algorithm in finding appropriate exponents depends on the ultimate periodicity threshold of the public matrices X and Y.

Success Rate for Algorithm 4 is shown in Figure 5. The private exponents k, r are random integers less that $(n-1)l_Z$, in $[(n-1)l_Z, (2n-1)l_Z]$, larger than $(2n-1)l_Z$ for the left, middle

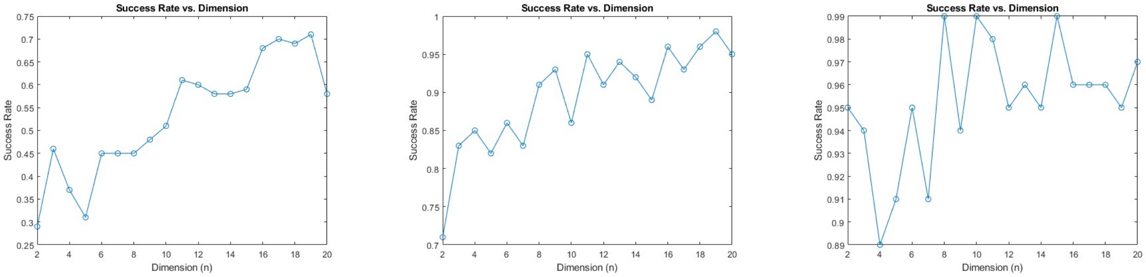

Figure 5: Attack 4 success rate with protocol's exponents less than $(n-1)l_Z$, in $[(n-1)l_Z, (2n-1)l_Z]$ and larger than $(2n-1)l_Z$ respectively

and right figure, respectively. The same follows for l, s but with l_W as the critical cycle length. For easier notations, we will denote both l_Z, l_W as l_Z in what follows.

We notice that it is sufficient for Attack 4 to perform almost optimally when the original protocol's exponents are larger than $(n-1)l_Z$ and not necessarily larger than $(2n-1)l_Z$, which is the guaranteed threshold for the CSR fomulas of Theorem 2.4 to hold. This might indicate that after transforming the matrices from $\mathbb{R}_{max}^{2^{n\times n}}$ to $\mathbb{R}_{max}^{2n\times 2n}$, the periodicity behaviour is still maintained. The small number of failures are probably due the critical graph of 1 or more of the public matrices having multiple strongly connected components. We also notice that Algorithm 4 does not perform as well when the used exponents are lower than $(n-1)l_Z$ which is expected since the CSR decomposition does not necessarily hold for such exponent values (i.e., entries in the critical rows or columns haven't necessarily converged yet to ultimate periodicity).

The time consumption comparison between the protocol and the two attacks is presented on Figure 6.The private exponents k, l, r, s in the protocol are random integers larger than $(2n-1)l_Z$.

We notice that recovering the key with Algorithm 4 is faster than Algorithm 3 since it limits its search to a finite set of values equal to the product of the critical cycle sizes of the two public matrices. We see that there is no significant difference between the attacker's time and the users' time in generating the secret key.

5.4 An attack based on absolute values of tropical pairs

In this section, we describe another attack that does not require the attacker to double the matrices sizes, which makes it more efficient than the previous ones. However, while performing this attack we lose some information and this worsens the success rate. We firstly present the following definition, inspired by the symmeterized semiring introduced in [2].

Definition 5.2 (Absolute Value)**.** Let $(a^{(1)}, a^{(2)}) \in R_{max}^2$. Then the absolute value of this pair is defined by $|(a^{(1)}, a^{(2)})| = a^{(1)} \oplus a^{(2)}$.

For two pairs $(a^{(1)}, a^{(2)}), (b^{(1)}, b^{(2)}) \in R_{max}^2$ the following properties hold:

$$\begin{aligned}|(a^{(1)}, a^{(2)}) \oplus (b^{(1)}, b^{(2)})| &= |(a^{(1)} \oplus b^{(1)}, a^{(2)} \oplus b^{(2)})| \\ &= a^{(1)} \oplus a^{(2)} \oplus b^{(1)} \oplus b^{(2)} = |(a^{(1)}, a^{(2)})| \oplus |(b^{(1)}, b^{(2)})|\end{aligned}$$

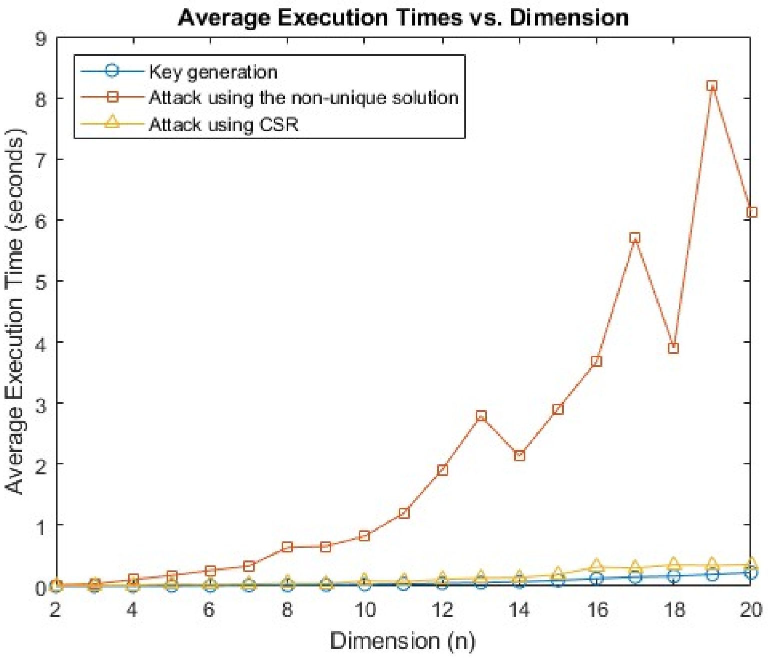

Figure 6: Time taken to attack and to generate Protocol 1

and

$$\begin{aligned}|(a^{(1)},a^{(2)})\otimes(b^{(1)},b^{(2)})| &= |(a^{(1)}\otimes b^{(1)}\oplus a^{(2)}\otimes b^{(2)}),(a^{(1)}\otimes b^{(2)}\oplus a^{(2)}\otimes b^{(1)})| \\ &= (a^{(1)}\otimes b^{(1)}\oplus a^{(2)}\otimes b^{(2)})\oplus(a^{(1)}\otimes b^{(2)}\oplus a^{(2)}\otimes b^{(1)}) \\ &= (a^{(1)}\oplus a^{(2)})\otimes(b^{(1)}\oplus b^{(2)}) = |(a^{(1)},a^{(2)})|\otimes|(b^{(1)},b^{(2)})|\end{aligned}$$

The first of them is rather unusual, since we only have $|a+b|\leq|a|+|b|$ in the usual algebra.

These properties can be also extended to matrices over tropical pairs, as we have

$$|(A\oplus B)_{ij}| = |A|_{ij}\oplus|B|_{ij},$$

which directly follows from the above definition, and

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

Here A,$B \in R_{max}^{2^{n\times n}}$ and $a_{ij}, b_{ij}\in\mathbb{R}^2_{max}\forall i,j\in[n]$.

We can use the above operations to perform the $(\mathbb{R}^2)^{n\times n}_{\max}$ matrix addition and multiplication, and we will get a half sized conventional $\mathbb{R}^{n\times n}_{\max}$ matrix. These operations with half sized result are often sufficient to attack the proposed protocol. The following example illustrates these operations.

Example 5.3. Take

$$A = \begin{bmatrix} (5,3) & (10,10) \\ (3,8) & (10,5) \end{bmatrix}, \quad B = \begin{bmatrix} (6,0) & (6,4) \\ (8,4) & (-1,2) \end{bmatrix}$$

Then we have

$$A \oplus B = \begin{bmatrix} (6,3) & (10,10) \\ (8,8) & (10,3) \end{bmatrix}, \quad A \otimes B = \begin{bmatrix} (18,18) & (12,12) \\ (18,4) & (12,14) \end{bmatrix}.$$

Performing the absolute values of these matrices we see that

$$\begin{aligned} |A| &= \begin{bmatrix} 5 & 10 \\ 8 & 10 \end{bmatrix}, \quad |B| = \begin{bmatrix} 6 & 6 \\ 8 & 2 \end{bmatrix} \\ |A| \oplus |B| &= \begin{bmatrix} 6 & 10 \\ 8 & 10 \end{bmatrix} = |A \oplus B|, \quad |A| \otimes |B| = \begin{bmatrix} 18 & 12 \\ 18 & 14 \end{bmatrix} = |A \otimes B| \end{aligned}$$

The attack utilizing these properties is very similar to the previous attack presented in Attack 4, but they only differ in the type of matrix transformation. Thus, we will describe the attack by presenting Algorithm 5.

```
Algorithm 5 Attacking Protocol 1 using the absolute values
Input: $X, Y, A_{(p)}, B_{(q)}$
Output: $Key$
Transform $X, Y, A_{(p)}$ to $|X|, |Y|, |A_{(p)}|$ using Definition 5.2
Calculate $\lambda(|X|) = \lambda_1, \lambda(|Y|) = \lambda_2$
Find a critical cycle $Z$ from $|X|$, and $W$ from $|Y|$, let their lengths be $l_Z$ and $l_W$, respectively.
Calculate $S_Z$, $R_Z$, $C_W$ and $S_W$ as in Theorem 2.3
Perform Algorithm 2 with Input:$|A_{(p)}|, |X|, I, |Y|$, then we get the output $k', l', \tau$
$Key = \tau \otimes \left(X^{\otimes k'} \otimes B_{(q)} \otimes Y^{\otimes l'}\right)$
```

We expect this attack to have a lower success rate than the previous ones due to the observation that $\alpha \otimes |X|^{\otimes t_1} \otimes |Y|^{\otimes t_2} = |U|$ does not always imply that $\alpha \otimes X^{\otimes t_1} \otimes Y^{\otimes t_2} = U$. Hence, the attack in this case will not successfully recover the secret key since the found exponents do not really solve the original tropical two-sided discrete log with shift problem over the tropical pairs. We will look at the following example, where this attack fails while the previous attacks succeed.

Example 5.4. Suppose Alice chooses the private exponents $k = 43$ and $l = 33$ and private parameters $c = 9$ and $p = -2$, with the two public matrices being:

$$X = \begin{bmatrix} (1,1) & (8,-4) \\ (-8,9) & (5,-7) \end{bmatrix}, Y = \begin{bmatrix} (-5,9) & (2,5) \\ (-8,5) & (8,10) \end{bmatrix}$$

then she computes $A_{(p)}$ and send it to Bob:

$$A_{(p)} = -2 + (43 + 33 - 2)(9) \otimes X^{\otimes 43} \otimes Y^{\otimes 33} = \begin{bmatrix} (1354, 1352) & (1359, 1357) \\ (1351, 1350) & (1356, 1355) \end{bmatrix}$$

The attacker then intercepts $A_{(p)}$ and applies the attacks described in Algorithm 4 and Algorithm 5.
Algorithm 4 starts by computing

$$\tilde{X} = \begin{bmatrix} 1 & 1 & 8 & -4 \\ 1 & 1 & -4 & 8 \\ -8 & 9 & 5 & -7 \\ 9 & -8 & -7 & 5 \end{bmatrix}, \quad \tilde{Y} = \begin{bmatrix} -5 & 9 & 2 & 5 \\ 9 & -5 & 5 & 2 \\ -8 & 5 & 8 & 10 \\ 5 & -8 & 10 & 8 \end{bmatrix}, \quad \tilde{A}_{(p)} = \begin{bmatrix} 1354 & 1352 & 1359 & 1357 \\ 1352 & 1354 & 1357 & 1359 \\ 1351 & 1350 & 1356 & 1355 \\ 1350 & 1351 & 1355 & 1356 \end{bmatrix}.$$

We find $k = 93$ and $l = 100$, which make the original $A_{(p)}$ in phase with $X^{\otimes 93} \otimes Y^{\otimes 100}$. Indeed,

$$A_{(p)} - X^{\otimes 93} \otimes Y^{\otimes 100} = \begin{bmatrix} (-431, -431) & (-431, -431) \\ (-431, -431) & (-431, -431) \end{bmatrix}$$

Algorithm 5 first computes

$$|X| = \begin{bmatrix} 1 & 8 \\ 9 & 5 \end{bmatrix}, \qquad |Y| = \begin{bmatrix} 9 & 5 \\ 5 & 10 \end{bmatrix}, \qquad |A_{(p)}| = \begin{bmatrix} 1354 & 1359 \\ 1351 & 1356 \end{bmatrix}$$

and finds $k = 11$ and $l = 20$, but they do not make the original $A_{(p)}$ in phase with $X^{\otimes 11} \otimes Y^{\otimes 20}$. Indeed, we see that

$$A_{(p)} - X^{\otimes 11} \otimes Y^{\otimes 20} = \begin{bmatrix} (1068, 1064) & (1068, 1064) \\ (1067, 1065) & (1067, 1065) \end{bmatrix}$$

Algorithm 5 does not reconstruct the secret key successfully. We have:

$$Key_{\text{Alice}} = Key_{\text{Bob}} = \begin{bmatrix} (2634, 2632) & (2639, 2637) \\ (2631, 2630) & (2636, 2635) \end{bmatrix}$$

$$Key_{Alg\ 4} = \begin{bmatrix} (2634, 2632) & (2639, 2637) \\ (2631, 2630) & (2636, 2635) \end{bmatrix} = Key_{\text{Alice}} = Key_{\text{Bob}}$$

$$Key_{Alg\ 5} = \begin{bmatrix} (2632, 2634) & (2637, 2639) \\ (2630, 2631) & (2635, 2636) \end{bmatrix} \neq Key_{\text{Alice}} = Key_{\text{Bob}}$$

Due to this counterexample we expect Algorithm 5 to have a lower success rate than the previous ones. However, it is going to be faster since it does not require doubling the matrix size.

We now examine the success rate for this attack as a function of matrix dimension. Additionally, we analyze the time required for the attacker to construct the secret shared key. The protocol parameters p, q, c, d and the matrix entries are random integers from -1000 to 1000, and 100 trails were performed for each dimension. The exponents k, l, r, s are random integers larger than $(2n-1)l_Z$. The success rate for Algorithm 5 is shown on Figure 7. We notice that Algorithm 5 has a relativity lower success rate compared to the

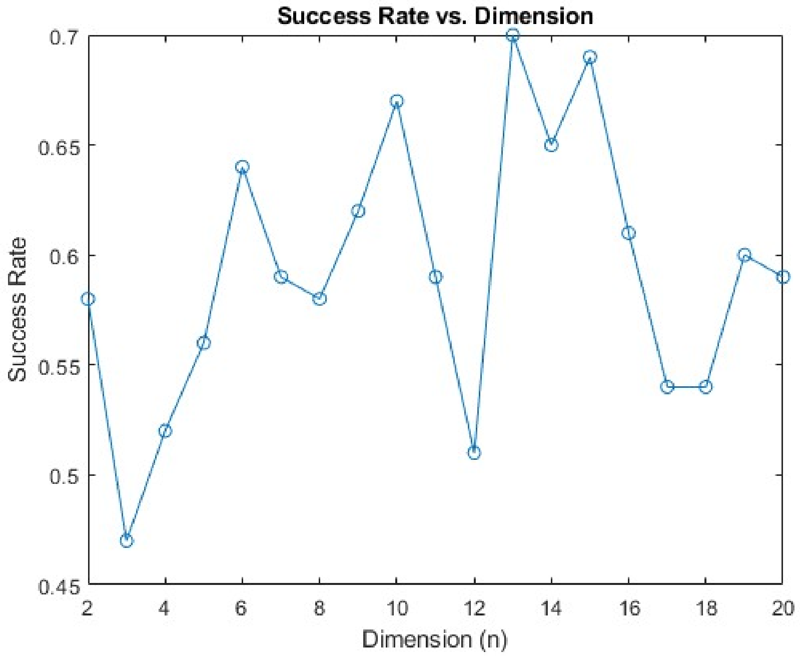

Figure 7: Success rate of Attack 5

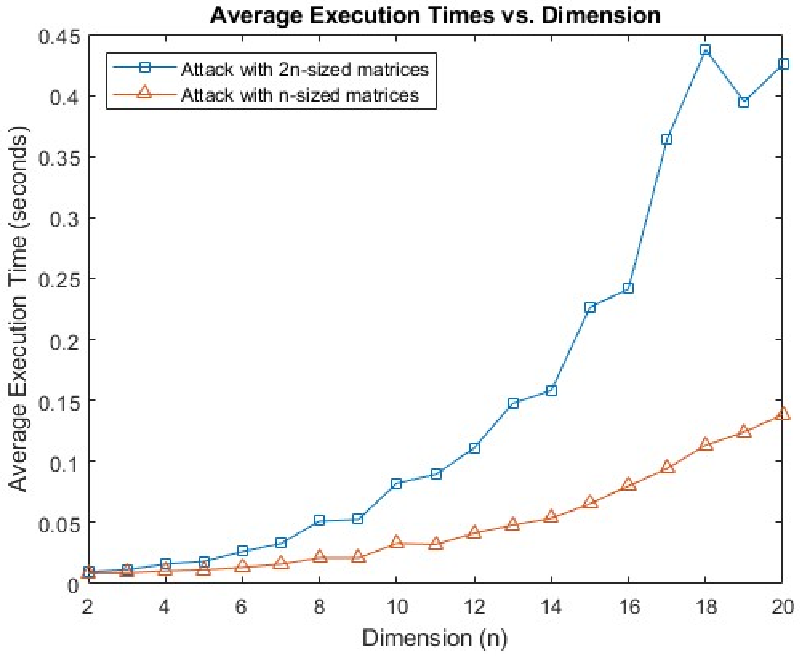

Figure 8: Time taken to perform Algorithm 5 and Algorithm 4

previous ones due to an extra source of failures that was illustrated in Example 5.4. The time comparison between Algorithm 5 and Algorithm 4 is shown in Figure 8.

As anticipated, Algorithm 5 is more efficient as it operates on matrices with half the size of the original ones, which expedites the process of computing the CSR terms and finding the exponents through solving the tropical two-sided discrete logarithm with shift, but it is losing quite a lot in terms of success rate.

6 Conclusion

In this paper, we presented the tropical two-sided discrete logarithm with shift problem and its solution. We showed that two algorithms solve the problem with a high success rate. The first algorithm relies on the non-uniqueness of the exponents that can satisfy the problem's equation, and it showed a perfect success rate but potentially slower convergence. The other algorithm heuristically solves the problem using the CSR decomposition, demonstrating a high success rate and a much faster execution time. We then presented a key exchange protocol that is based on the tropical semiring of pairs. We firstly showed that the matrix operations over the tropical semiring of pairs can be equivalently calculated using the conventional tropical semiring by doubling the matrices' size. After the transformation to the conventional tropical semiring, the problem was reduced to the tropical two-sided discrete logarithm with shift, which can be solved using the two proposed algorithms. Lastly, we introduced an even faster approach that does not require the attacker to double the matrix size, showing roughly twice the speed of the previous heuristic attack, but it has a lower success rate.

We also note that the proposed protocol remains vulnerable to the known tropical cryptographic attacks even if the parties use matrix polynomials over the semiring of pairs. If Alice and Bob use matrix polynomials over the semiring of pairs, then the problem will not reduce to the tropical two-sided logarithm with shift, but in this case it is possible to apply the Kotov-Ushakov attack [8] (which, however, shows an exponential blow up in the number of operations as the maximal degree of the polynomials increases). The main reason for insecurity of the protocols using the matrix algebra over the tropical semiring of pairs is the existence of injective homomorphism from this matrix algebra to the ordinary tropical matrix algebra.

7 Acknowledgement

The authors are grateful to Sergeĭ's M.Sci. student Sarah Elt who gave them the idea to try using also the absolute value of pairs and its extension to matrices.

References

[1] K. Ahmed, S. Pal, and R. Mohan. Key exchange protocol based upon a modified tropical structure. *Communications in Algebra*, 51(1):214–223, 2023.

[2] F.L. Baccelli, G. Cohen, G.J. Olsder, and J.P. Quadrat. Synchronization and linearity - an algebra for discrete event systems. *The Journal of the Operational Research Society*, 45, 01 1994.

[3] P. Butkovič. *Max-linear Systems: Theory and Algorithms*. Springer, London, 2010.

[4] G. Cohen, D. Dubois, J.P. Quadrat, and M. Viot. Analyse du compartement périodique de systémes de production par la théorie des dioïdes. Rr-0191, INRIA, 1983. Research Report RR-0191, INRIA. `https://inria.hal.science/inria-00076367`.

[5] D. Grigoriev and V. Shpilrain. Tropical cryptography. *Communications in Algebra*, 42:2624 – 2632, 2013.

[6] D. Grigoriev and V. Shpilrain. Tropical cryptography II: Extensions by homomorphisms. *Communications in Algebra*, 47(10):4224–4229, 2019.

[7] S. Isaac and D. Kahrobaei. A closer look at the tropical cryptography. *International Journal of Computer Mathematics: Computer Systems Theory*, 6(2):137–142, 2021.

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

[9] G. Merlet, T. Nowak, and S. Sergeev. Weak CSR expansions and transience bounds in max-plus algebra. *Linear Algebra and its Applications*, 461:163–199, 2014.

[10] A. Muanalifah. *Public key cryptography based on tropical algebra*. PhD thesis, University of Birmingham, 2023.

[11] A. Muanalifah and S. Sergeev. Modifying the tropical version of Stickel's key exchange protocol. *Applications of Mathematics*, 65:727–753, 12 2020.

[12] A. Muanalifah and S. Sergeev. On the tropical discrete logarithm problem and security of a protocol based on tropical semidirect product. *Communications in Algebra*, 50(2):861–879, 2022.

[13] K. Nachtigall. Powers of matrices over an extremal algebra with applications to periodic graphs. *Mathematical Methods of Operations Research*, 46:87–102, 1997.

[14] D. Rudy and C. Monico. Remarks on a tropical key exchange system. *Journal of Mathematical Cryptology*, 15(1):280–283, 2020.

[15] S. Sergeev and H. Schneider. CSR expansions of matrix powers in max algebra. *Transactions of the American Mathematical Society*, 364(11):5969–5994, 2012.

Sulaiman Alhussaini
University of Birmingham, School of Mathematics, Birmingham, Edgbaston B15 2TT, UK
saa399@student.bham.ac.uk

Craig Collett
University of Birmingham, School of Mathematics, Birmingham, Edgbaston B15 2TT, UK
CRC957@student.bham.ac.uk

Sergeĭ Sergeev
University of Birmingham, School of Mathematics, Birmingham, Edgbaston B15 2TT, UK
s.sergeev@bham.ac.uk