


Article

Attacking Tropical Stickel Protocol by MILP and Heuristic Optimization Techniques

Sulaiman Alhussaini and Sergei Sergeev * 

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

* Correspondence: s.sergeev@bham.ac.uk

Abstract

Known attacks on the tropical implementation of Stickel protocol involve finding minimal covers for a certain covering problem, and this leads to an exponential growth in the worst case time required to recover the secret key as the used polynomial degree increases. The computational inefficiency of this attack is also observed in practice, unless the number of explored covers is limited, on the expense of the success rate of the attack. Consequently, it can be argued that Alice and Bob can still repel these attacks on tropical Stickel protocol by utilizing very high polynomial degrees, a feasible approach due to the efficiency of tropical operations. The same is true for the implementation of Stickel protocol over some other semirings with idempotent addition (such as the max–min or digital semiring). In this paper, we propose alternative methods to attack the Stickel protocols that avoid solving the covering problem. These methods involve framing the attacks as a mixed integer linear programming (MILP) problem or applying certain heuristic global optimization techniques. We also include a number of numerical experiments to analyze the success rate and the time required to execute the suggested attacks in practice.

Keywords: public key cryptography; key exchange protocol; cryptographic attack; tropical cryptography

JEL Classification: 94A60; 15A80



Academic Editors: Yangguang Tian
and Danda B. Rawat

Received: 15 May 2025

Revised: 30 August 2025

Accepted: 11 September 2025

Published: 3 October 2025

Citation: Alhussaini, S.; Sergeev, S.
Attacking Tropical Stickel Protocol by
MILP and Heuristic Optimization
Techniques. *J. Cybersecur. Priv.* **2025**, *5*,
82. [https://doi.org/10.3390/
jcp5040082](https://doi.org/10.3390/jcp5040082)

Copyright: © 2025 by the authors.
Licensee MDPI, Basel, Switzerland.
This article is an open access article
distributed under the terms and
conditions of the Creative Commons
Attribution (CC BY) license
([https://creativecommons.org/
licenses/by/4.0/](https://creativecommons.org/licenses/by/4.0/)).

1. Introduction

A key exchange protocol is a process where two parties, commonly referred to as Alice and Bob, collaboratively generate a shared secret key using public information and messages exchanged over a public channel. The security of a protocol is determined by its ability to prevent an attacker from easily recovering the shared secret key using these public information and intercepted messages, typically by ensuring that the attacker must solve a problem that is computationally hard to succeed in practice. *NP*-hard problems or problems with exponential worst case complexity are natural candidates for these (although *NP*-hardness or exponential worst case complexity are not enough to guarantee the security of protocols). Such protocols often rely on various algebraic tools to achieve the desired security properties.

Polynomials over the tropical (max-plus) semiring are one of the recent tools utilized in key exchange protocols, appearing in the tropical implementation of the Stickel protocol proposed by Grigoriev and Shpilrain [1]. This new implementation followed Shpilrain's successful attack [2] on the initial Stickel protocol [3] and has become one of the most

popular key exchange protocols utilizing tropical operations. The rationale behind suggesting a tropical implementation of the protocol was to avoid obvious attacks involving linear algebra and matrix inverses, which were effective against the original protocol. The Stickel protocol can be similarly implemented over any semiring, and its implementation over max–min and max- T semirings (where the symbol T stands for arbitrary T -norm [4]) is analyzed in [5]. The survey in [6] argues for broadening semiring choices beyond the tropical semiring and reviews the main hard problems in semiring-based cryptography.

Kotov and Ushakov [7] later suggested an attack on the tropical Stickel protocol by transforming the underlying problem into finding a special solution to the protocol's associated system of equations of the form $A \otimes x = b$, the complete solution set to which can be described using solution to a certain covering problem. The attacker still faces a significant challenge: solving the problem to find a solution to the covering problem that satisfies certain conditions. To find such a cover, the attacker potentially needs to check all the minimal covers and find a cover that actually produces the required special solution to $A \otimes x = b$. Therefore, this approach is less effective when Alice and Bob use high-degree polynomials, which can be efficiently managed by Alice and Bob with minimal computational resources due to the efficient nature of tropical operations. An analogue of the Kotov–Ushakov attack against the max–min and, more generally, max- T implementations of the Stickel protocol can be similarly proposed [5]. However, it encounters a similar challenge of finding a minimal solution with special properties, resulting in an exponential increase in the worst case execution time.

The main idea of this paper is to introduce alternative attack strategies that avoid solving the covering problem encountered in a conventional Kotov–Ushakov attack. Specifically, we propose an attack where we instead find a solution x that minimizes the protocol's associated objective function $\sum_i ((A \otimes x)_i - b_i)^2$ using a heuristic optimization technique. We will compare this with a different approach where some of the known attacks are formulated as mixed integer linear programs, allowing the shared key to be recovered using an MILP solver.

This paper is organized as follows: Section 2 covers preliminaries and basic definitions, particularly those related to the matrix algebra over the tropical and max–min semirings, as well as the targeted key exchange protocols based on these semirings. In Section 3, we present our alternative attacks, provide numerical implementations demonstrating their performance, and compare them with a typical Kotov–Ushakov attack. In Sections 4 and 5, we discuss how these proposed attacks can also target a recent implementation of Stickel protocol over a newly introduced semiring known as the “digital semiring” [8] and a recently proposed tropical digital signature protocol [9], respectively. Section 6 is dedicated to conclusions and discussion. Our code implementations have been made available on GitHub: <https://github.com/suliman1n/Attacking-Tropical-Stickel-Protocol-by-MILP-and-Heuristic-Optimization-Techniques> (accessed on 12 September 2025) and were developed using MATLAB R2023b.

2. Preliminaries

In this section, we are going to introduce the matrix algebra over the tropical and max–min semirings, followed by the Stickel protocol over these semirings and two versions of the Kotov–Ushakov attack. Note that we use the standard notation $[m] = \{1, \dots, m\}$ and $[n] = \{1, \dots, n\}$ for most common index sets.

Definition 1 (Matrix Algebra over Semirings [10]). *We define the tropical semiring as $\mathbb{R}_{\max} = (\mathbb{R} \cup \{-\infty\}, \oplus, \otimes)$, and the max–min semiring as $\mathbb{R}_{\max, \min} = (\mathbb{R} \cup \{-\infty\} \cup \{\infty\}, \oplus, \otimes)$, where the arithmetical operations are defined by $x \oplus y = \max(x, y)$ and $x \otimes y = x + y$ for all $x, y \in \mathbb{R}_{\max}$*

in the tropical case, and by $x \oplus y = \max(x, y)$ and $x \otimes y = \min(x, y)$ for all $x, y \in \mathbb{R}_{\max, \min}$ for the max-min case. When addressing both semirings at the same time or any semiring in general, we will use the symbol \mathbb{R}_T (also reminiscent of max-T semirings, of which the max-min semiring and the non-positive part of the tropical semiring are special cases).

The arithmetic operations over any semiring are naturally extended to include matrices and vectors. In particular, the operation $A \otimes \alpha = \alpha \otimes A$, where $\alpha \in \mathbb{R}_T$, $A \in \mathbb{R}_T^{m \times n}$ 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 matrix addition $A \oplus B$ of two matrices $A \in \mathbb{R}_T^{m \times n}$ and $B \in \mathbb{R}_T^{m \times n}$, 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 matrix multiplication of two matrices is also similar to the “traditional” algebra. Namely, we define $A \otimes B$ for two matrices, where $A \in \mathbb{R}_T^{m \times p}$ and $B \in \mathbb{R}_T^{p \times n}$, as follows:

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

The arithmetics of the max-plus and max-min semirings are summarized in Table 1 below.

Table 1. Summary of semiring operations.

Semiring	Ground Set	\oplus	\otimes	Zero Element	Identity Element
Tropical	$\mathbb{R} \cup \{-\infty\}$	$\max(a, b)$	$a + b$	$-\infty$	0
Max-min	$\mathbb{R} \cup \{-\infty\} \cup \{+\infty\}$	$\max(a, b)$	$\min(a, b)$	$-\infty$	∞

Note that, despite introducing this arithmetic, we will also quite often utilize the usual arithmetical operations to introduce concepts and explain arguments, mostly since the optimization methods that we are going to exploit are based on the usual arithmetic.

Definition 2 (Matrix Powers). For $M \in \mathbb{R}_T^{n \times n}$, the n -th power of M is denoted by $M^{\otimes n}$, and is equal to

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

By definition, any square matrix to the power 0 is the identity.

Definition 3 (Identity Matrix). The identity matrix $I \in \mathbb{R}_T^{n \times n}$ is of the form $(I)_{ij} = \delta_{ij}$ where

$$\delta_{ij} = \begin{cases} 0 \text{ for tropical case, or } \infty \text{ for max-min case} & \text{if } i = j \\ -\infty & \text{otherwise} \end{cases}$$

Note that the identity matrix can be defined also for a general semiring: one sets the diagonal entries equal to the semiring unity and the off-diagonal entries to the semiring zero [10].

Subsequently, we define the matrix polynomials.

Definition 4 (Matrix Polynomials). *Matrix polynomial is a function of the form*

$$A \mapsto p(A) = \bigoplus_{k=0}^d a_k \otimes A^{\otimes k}.$$

where $a_k \in \mathbb{R}_T$ for $k = 0, 1, \dots, d$. Here, $A \in \mathbb{R}_T^{n \times n}$ is a square matrix of any dimension n .

Any two matrix polynomials of the same matrix over any semiring commute just like in the classical algebra [10], and this fact was utilized by Grigoriev and Shpilrain [1] to construct a tropical implementation of the Stickel protocol (Protocol 1). Quite obviously, this protocol can be implemented over any semiring (and in particular, over the max–min semiring).

Protocol 1 (Stickel Protocol over Semirings).

1. Alice and Bob agree on public matrices $A, B, W \in \mathbb{R}_T^{n \times n}$.
2. Alice chooses two random tropical polynomials, $p_1(x)$ and $p_2(x)$, and sends $U = p_1(A) \otimes W \otimes p_2(B)$ to Bob.
3. Bob chooses two random tropical polynomials, $q_1(x)$ and $q_2(x)$, and sends $V = q_1(A) \otimes W \otimes q_2(B)$ to Alice.
4. Alice computes her secret key using a public key V obtained from Bob, which is $K_a = p_1(A) \otimes V \otimes p_2(B)$.
5. Bob also computes his secret key using Alice's public key U , which is $K_b = q_1(A) \otimes U \otimes q_2(B)$.

The two parties end up with an identical key in both protocols due to the commutativity of polynomials of the same matrix. Formally, we have $K_a = p_1(A) \otimes V \otimes p_2(B) = p_1(A) \otimes q_1(A) \otimes W \otimes q_2(B) \otimes p_2(B) = q_1(A) \otimes p_1(A) \otimes W \otimes p_2(B) \otimes q_2(B) = q_1(A) \otimes U \otimes q_2(B) = K_b$.

An attack against Protocol 1 over the tropical semiring was published by Kotov and Ushakov [7], and an analog of this attack against Protocol 1 over max–min semiring (and, more generally, max- T semiring with continuous T -norm) was discussed in [5]. In the next section, we will compare their performance with the optimization methods proposed in the present paper. The presented attacks break Protocol 1 by solving the following problem

Problem 1. *Given the public matrices U and W where $U = p_1(A) \otimes W \otimes p_2(B)$ for some unknown $p_1(A)$ and $p_2(B)$, find $p'_1(A)$ and $p'_2(B)$ such that $U = p'_1(A) \otimes W \otimes p'_2(B)$.*

or the following problem for the attack presented in Section 3.3

Problem 2. *Given the public matrices U and W where $U = p_1(A) \otimes W \otimes p_2(B)$ for some unknown $p_1(A)$ and $p_2(B)$, find X and Y such that X commutes with A , Y commutes with B , and $X \otimes W \otimes Y = U$.*

Solving these problems is sufficient but not necessary to compromise the protocol. For example, the attack presented in [11] offers a more efficient approach against this particular version of the protocol. This attack applies under the two conditions detailed in [12]. Consequently, variants of the Stickel protocol that employ broader classes of commuting matrices (beyond polynomials of some public matrices) or alternative semirings, other than the tropical semiring, may not be vulnerable to this attack. Currently, no such variant of the tropical Stickel protocol is known to us. In such generalized settings, solving the aforementioned problems may be the only viable approach for attacking the Stickel protocol,

which is why we still consider them relevant. In [13], the authors proposed a non-Stickel-type protocol based on a tropical multiple-exponentiation problem and showed that known attacks do not directly apply to this construction.

We now turn to the specific goal of the upcoming attacks. The objectives of the attacks is to find the polynomial coefficients x_α, y_β for all $\alpha, \beta \in \{0, \dots, D\}$ where D is the maximum polynomial degree used in the protocols, and hence construct $X = \bigoplus_{\alpha=0}^D (x_\alpha \otimes A^{\otimes \alpha})$ and $Y = \bigoplus_{\beta=0}^D (y_\beta \otimes B^{\otimes \beta})$ that satisfy $X \otimes W \otimes Y = U$. Thus, the attacks aim to recover the shared secret key, by turning $X \otimes W \otimes Y = U$ into the form of a system of linear equations of the shape $A \otimes x = b$ and search for a solution that satisfies a special structure among all possible solutions. Thus, these attacks encounter the problem of finding all minimal solutions of a linear system of the shape $A \otimes x = b$, which is easy to solve when Alice and Bob use low-degree polynomials, as demonstrated numerically in [7,14,15] for the tropical case, or in [5] for the max–min case. However, it becomes significantly more challenging for higher-degree polynomials due to the exponential increase in the number of the minimal solutions of the system. The full details of the Kotov–Ushakov attack are described below.

We are aiming to find two matrices X and Y , where they are expressed as

$$X = \bigoplus_{\alpha=0}^D (x_\alpha \otimes A^{\otimes \alpha})$$

$$Y = \bigoplus_{\beta=0}^D (y_\beta \otimes B^{\otimes \beta}),$$

such that D is sufficiently large to exceed the maximal degree of any polynomial that Alice and Bob might use. Then, we substitute these expressions into $X \otimes W \otimes Y = U$ to obtain

$$U = \bigoplus_{\alpha=0}^D (x_\alpha \otimes A^{\otimes \alpha}) \otimes W \otimes \bigoplus_{\beta=0}^D (y_\beta \otimes B^{\otimes \beta}).$$

Combining the summations, we obtain

$$U = \bigoplus_{\alpha, \beta=0}^D (x_\alpha \otimes A^{\otimes \alpha}) \otimes W \otimes (y_\beta \otimes B^{\otimes \beta}).$$

Rearranging those using the distributivity law will give

$$\bigoplus_{\alpha, \beta=0}^D x_\alpha \otimes y_\beta \otimes (A^{\otimes \alpha} \otimes W \otimes B^{\otimes \beta}) = U.$$

We then denote $R^{\alpha\beta} = A^{\otimes \alpha} \otimes W \otimes B^{\otimes \beta}$ and therefore we can write

$$\bigoplus_{\alpha, \beta=0}^D x_\alpha \otimes y_\beta \otimes (R^{\alpha\beta})_{\gamma\delta} = U_{\gamma\delta} \quad \forall \gamma, \delta \in [n] \times [n]. \quad (1)$$

If we additionally denote $z_{\alpha\beta} = x_\alpha \otimes y_\beta$, we have

$$\bigoplus_{\alpha, \beta=0}^D z_{\alpha\beta} \otimes (R^{\alpha\beta})_{\gamma\delta} = U_{\gamma\delta} \quad \forall \gamma, \delta \in [n] \times [n]. \quad (2)$$

We have arrived at a system of linear equations of the shape $A \otimes x = b$ with coefficients $(R^{\alpha\beta})_{\gamma\delta}$ and unknowns $z_{\alpha\beta}$.

We now need to scan all solutions to this system, and obtain the solution that satisfies $z_{\alpha\beta} = x_\alpha \otimes y_\beta$ for some $x_\alpha, y_\beta \in \mathbb{N} \quad \forall \alpha, \beta \in \{0, 1, \dots, D\}$. Thus, using the theory of $A \otimes x = b$ solvability, we need to find the greatest solution, as well as all minimal solutions. For each minimal solution, we need to search for a vector $(z_{\alpha\beta})$ in the interval between the minimal solution and the greatest solution that solves $z_{\alpha\beta} = x_\alpha \otimes y_\beta$ for some x_α, y_β .

Note that, for the tropical case, a minimal solution can be found by finding a minimal cover (i.e., the minimal number of variables that satisfy all the equations in the system), and the other variables are set to $-\infty$. The following algorithm captures this process.

For the max–min case, we similarly need to compute the greatest solution c (using Lemma 3.2 in [16]) and all minimal solutions $d^{(i)}$'s (using Section 3.3 in [17] or Chapter 3 in [18]), and search for the required solution. The following algorithm captures this process.

Note that system (4) can be transformed into a problem of mixed-integer linear programming as shown in [5].

These attacks succeed under the condition that the attacker is using D that exceeds the greatest polynomial degrees used by Alice and Bob, because, in this case, these attacks produce X and Y that satisfy $X \otimes W \otimes Y = U$. The proof can be found in [5,15]. However, they exhibit exponential growth in computational time relative to the used polynomial degree in the protocol. Numerical experiments showing the time taken by these attacks to compromise the tropical implementation of Protocol 1 can be found in [7,14,15], and for the max–min implementation, see [5]. Table 2 summarizes a representative subset of these runtime results.

Table 2. Comparison of runtimes for Algorithm 1 (tropical) and Algorithm 2 (max–min).

Polynomial Degree D	Algorithm 1 Time (s)	Algorithm 2 Time (s)
3	<0.01	0.04
5	<0.01	2.9
9	<0.01	12,204
30	223	N/A
50	2640	N/A

The most computationally intensive component of the attacks described above is the enumeration of all minimal covers. This problem is fundamentally equivalent to the hypergraph traversal hitting sets enumeration, a widely studied topic in various fields such as combinatorics and optimization. To formalize this connection in the tropical case (Algorithm 1), we firstly present some relevant definitions.

Definition 5 (Hypergraph). *A hypergraph $H = (V, \mathcal{E})$ consists of a vertex set V and a set of hyperedges \mathcal{E} , where each hyperedge $E \in \mathcal{E}$ is a subset of V .*

Definition 6 (Hitting set (e.g., ref. [19])). *A hitting set for a hypergraph $H = (V, \mathcal{E})$ is a subset $K \subseteq V$ such that $K \cap E \neq \emptyset$ for every $E \in \mathcal{E}$. A hitting set is minimal if no proper subset of K is a hitting set.*

The enumeration process of all minimal covers of $[n] \times [n]$ by the computed sets $S_{\alpha\beta}$ in Algorithm 1 is equivalent to the process of enumerating all minimal hitting sets of the hypergraph $H = (\{0, \dots, D\} \times \{0, \dots, D\}, \{G_{11}, G_{12}, \dots, G_{nn}\})$ where $G_{\gamma\delta} = \{(\alpha, \beta) \in \{0, \dots, D\} \times \{0, \dots, D\} : \bigoplus_{\alpha, \beta=0}^D c_{\alpha\beta} \otimes (R^{\alpha\beta})_{\gamma\delta} = U_{\gamma\delta}\}$. This is because we know that a minimal cover $\mathcal{C} \subseteq \{0, \dots, D\} \times \{0, \dots, D\}$ in Algorithm 1 satisfies $\bigcup_{(\alpha, \beta) \in \mathcal{C}} S_{\alpha\beta} = [n] \times [n]$. This is equivalent to \mathcal{C} intersecting every hyperedge $G_{\gamma\delta}$ (i.e., \mathcal{C} is a hitting set for H). Minimality of \mathcal{C} as a hitting set then follows since removing any $(\alpha, \beta) \in \mathcal{C}$ would leave some $G_{\gamma\delta}$ unhit. Similarly, given $H = (\{0, \dots, D\} \times \{0, \dots, D\}, \{G_{11}, G_{12}, \dots, G_{nn}\})$, we know that a minimal hitting set $\mathcal{C} \subseteq \{0, \dots, D\} \times \{0, \dots, D\}$ intersects every $G_{\gamma\delta}$. By

defining $S_{\alpha\beta} = \{(\gamma, \delta) \in [n] \times [n] : (\alpha, \beta) \in G_{\gamma\delta}\}$, the union $\bigcup_{(\alpha, \beta) \in \mathcal{C}} S_{\alpha\beta}$ covers $[n] \times [n]$. Minimality of \mathcal{C} as a cover follows since no smaller subset of \mathcal{C} can cover $[n] \times [n]$. This means there is a one-to-one correspondence between the enumerated minimal covers in Algorithm 1 and the minimal hitting sets of H .

Algorithm 1 Tropical Kotov–Ushakov attack [7]

Input: Public matrices A, B, W , transmitted message U , maximum polynomial degree D

Output: Coefficients x_α, y_β .

1: Compute

$$c_{\alpha\beta} = \min_{\gamma, \delta \in [n]} (U_{\gamma\delta} - R_{\gamma\delta}^{\alpha\beta})$$

$$S_{\alpha\beta} = \arg \min_{\gamma, \delta \in [n]} (U_{\gamma\delta} - R_{\gamma\delta}^{\alpha\beta}).$$

2: Among all minimal covers of $[n] \times [n]$ by $S_{\alpha\beta}$, that is, all minimal subsets $\mathcal{C} \subseteq \{0, \dots, D\} \times \{0, \dots, D\}$ such that

$$\bigcup_{(\alpha, \beta) \in \mathcal{C}} S_{\alpha\beta} = [n] \times [n],$$

find a cover for which the system

$$\begin{aligned} x_\alpha + y_\beta &= c_{\alpha\beta}, & \text{if } (\alpha, \beta) \in \mathcal{C}, \\ x_\alpha + y_\beta &\leq c_{\alpha\beta}, & \text{if otherwise.} \end{aligned} \quad (3)$$

is solvable.

3: **return** (x_α, y_β) .

Algorithm 2 Max–min Kotov–Ushakov attack [5]

Input: Public matrices A, B, W , transmitted message U , maximum polynomial degree D .

Output: Coefficients x_α, y_β .

1: Compute the maximum solution c of system (2) as

$$c_{\alpha\beta} = \min_{\gamma, \delta \in [n]} (U_{\gamma\delta} : R_{\gamma\delta}^{\alpha\beta} > U_{\gamma\delta}) \quad \forall \alpha, \beta \in \{0, \dots, D\}$$

2: Compute all minimal solutions $d^{(i)}$ of system (2).

3: Find a minimal solution $d^{(i)}$ with components $d_{\alpha\beta}^{(i)}$ for which the system

$$d_{\alpha\beta}^{(i)} \leq x_\alpha \otimes y_\beta \leq c_{\alpha\beta} \quad \forall \alpha, \beta \in \{0, \dots, D\} \quad (4)$$

is solvable.

4: **return** (x_α, y_β) .

From this perspective, we know that a hypergraph can have exponentially many minimal hitting sets, so a polynomial-time algorithm for the enumeration process in the above attacks is not possible, but it can be achieved in incremental quasi-polynomial time [19]. This also implies the exponential worst case complexity of the Kotov–Ushakov attacks (Algorithms 1 and 2). Another closely related problem is finding the smallest hitting set, which is known to be NP-hard [19], although the Kotov–Ushakov attacks are not aimed precisely at this problem. Nevertheless, their exponential worst-case complexity presents a major drawback. To address this, we next study the application of some well-known optimization techniques.

3. Attacks Using Optimization

In this section, we explore more efficient approaches to attacking the tropical and max-min implementations of Protocol 1 that avoid the minimal covering problem and the associated exponential complexity, which are evident in Algorithms 1 and 2. For all experiments, we use a matrix dimension of 10, which is the default parameter suggested in [1,7]. This choice allows us to compare the performance of the optimization methods discussed in this paper with the performance of Algorithms 1 and 2. To read this section, the basic knowledge of semiring algebra (see Definitions 1–4) as well as the knowledge of above mentioned Problems and Protocol 1 will be required from the reader. See also Table 1 for a summary of tropical and max-min arithmetics.

3.1. Simulated Annealing

Both Algorithms 1 and 2 aim to find all minimal solutions that satisfy all equations in system (2). In this approach, we aim to find a solution that minimizes the Euclidean distance between the left hand side and the right hand side of the system. Formally, we solve the following:

$$\min_{x_\alpha, y_\beta} \sum_{(\gamma, \delta) \in [n] \times [n]} f_{\gamma\delta}^2$$

where

$$f_{\gamma\delta} = \max_{\substack{\alpha \in \{0,1,\dots,D\} \\ \beta \in \{0,1,\dots,D\}}} (x_\alpha \otimes y_\beta \otimes R_{\gamma\delta}^{\alpha\beta}) - U_{\gamma\delta} \quad (5)$$

This objective function is complex with numerous local minima. However, the simulated annealing algorithm (see, e.g., ref. [20]), when initialized with a sufficiently high temperature parameter, effectively navigates these local minima and converges to the global minimum, where the objective function equals zero. The tropical and max-min objective functions are defined, respectively, in (6) and (7).

We now formally outline how the tropical Stickel protocol (Protocol 1 with $\mathbb{R}_T = \mathbb{R}_{\max}$) is attacked using the simulated annealing method; see Algorithm 3.

To ensure the simulated annealing algorithm escapes local minima, the initial temperature has to be sufficiently large to allow the acceptance of worse points. A practical method for determining this initial temperature is to set it based on the sample variance of multiple randomly evaluated points (e.g., ref. [21]). This captures the variability of the objective function, reducing the risk of getting stuck in local minima.

The performance of simulated annealing is also highly sensitive to the initial point. An optimal initial point can facilitate a quicker convergence to the global minimum. However, in our implementation, we started with a random point, as it seems the high initial temperature helps to mitigate the potential drawback of this non-optimal initialization.

Furthermore, as Alice and Bob increase the range of entries for public matrices and polynomial coefficients, the objective function becomes more complex. Kotov–Ushakov attack (Algorithm 1) is not impacted by this, as it relies on solving a minimal covering problem that is independent of the individual entries (i.e., finding minimal covers using $S_{\alpha\beta}$'s, which are independent of the used entries). We will therefore also examine how Algorithm 3 performs under such conditions. Figure 1 shows the time taken in seconds to compromise Protocol 1 using Algorithm 3 for different degrees and entry ranges. All numerical experiments were executed on Windows 11 64-bit, with an Intel(R) Core(TM) i7-9750H CPU @ 2.60 GHz and 16.0 GB RAM.

This attack achieved a perfect success rate and is significantly faster than Algorithm 1, averaging about 30 times the speed for a polynomial degree of 50 (refer to [14] for detailed experimental results of Algorithm 1). Note that the attack still performs well for higher entry ranges, but it is more likely that we encounter some samples that take significantly

longer than average to converge. This is probably caused by the increased complexity of the objective function and how optimal the probabilistic selection of the next neighboring point in the simulated annealing algorithm is, as well as the number of iterations performed until convergence.

Algorithm 3 Attacking tropical Stickel protocol using simulated annealing

Input: Public matrices A, B, W , transmitted message U , maximum polynomial degree D

Output: Matrices X, Y .

- 1: Compute $T_{\alpha\beta} = A^{\otimes\alpha} \otimes W \otimes B^{\otimes\beta} - U$ for all $0 \leq \alpha, \beta \leq D$.
- 2: Define objective

$$F(x, y) = \sum_{\gamma, \delta} \left(\max_{\alpha, \beta} (x_{\alpha} + y_{\beta} + T_{\alpha\beta}^{\gamma\delta}) \right)^2. \quad (6)$$

- 3: Initialize temperature T and choose a random starting point (x^c, y^c) .
- 4: **repeat**
- 5: Set trial counter $k \leftarrow k + 1$ (initialize $k \leftarrow 0$ before the loop).
- 6: Update the temperature: $T_k \leftarrow T \times 0.95^k$.
- 7: Select a new candidate point (x^{test}, y^{test}) from the neighbourhood of (x^c, y^c) .
- 8: Compute $\Delta \leftarrow F(x^{test}, y^{test}) - F(x^c, y^c)$.
- 9: **if** $\exp\left(-\frac{\Delta}{T_k}\right) > \text{Random}[0, 1)$ **then**
- 10: Accept the candidate: $(x^c, y^c) \leftarrow (x^{test}, y^{test})$.
- 11: **until** $F(x^c, y^c) = 0$
- 12: Let $(\bar{x}, \bar{y}) = (x^c, y^c)$.
- 13: Construct

$$X = \bigoplus_{\alpha=0}^D (\bar{x}_{\alpha} \otimes A^{\otimes\alpha}) \quad \text{and} \quad Y = \bigoplus_{\beta=0}^D (\bar{y}_{\beta} \otimes B^{\otimes\beta}).$$

- 14: **return** (X, Y) .
-

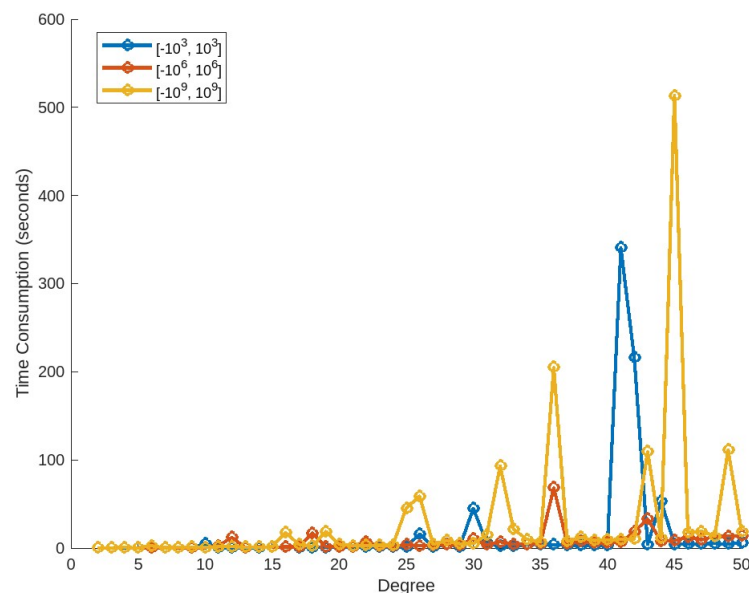


Figure 1. Attacking the tropical version of Protocol 1 using Algorithm 3.

For the max–min implementation of Protocol 1, the simulated annealing algorithm often struggles to reach the zero of the objective function, frequently getting stuck in local minima. Therefore, we have to utilize the lowest local minimum obtained to attempt to recover the secret key; see Step 4 in Algorithm 4.

Algorithm 4 Attacking max–min Stickel protocol using simulated annealing**Input:** Public matrices A, B, W , transmitted message U , maximum polynomial degree D **Output:** Matrices X, Y .

- 1: Compute $R_{\alpha\beta} = A^{\otimes\alpha} \otimes W \otimes B^{\otimes\beta}$ for all $0 \leq \alpha, \beta \leq D$.
- 2: Define objective

$$F(x, y) = \sum_{\gamma, \delta} \left(\max_{\alpha, \beta} (x_\alpha + y_\beta + R_{\alpha\beta}^{\gamma\delta} - U_{\gamma\delta}) \right)^2. \quad (7)$$

- 3: Initialize temperature T and choose a random starting point (x^c, y^c) .
- 4: **repeat**
- 5: Set trial counter $k \leftarrow k + 1$ (initialize $k \leftarrow 0$ before the loop).
- 6: Update the temperature: $T_k \leftarrow T \times 0.95^k$.
- 7: Select a new candidate point (x^{test}, y^{test}) from the neighbourhood of (x^c, y^c) .
- 8: Compute $\Delta \leftarrow F(x^{test}, y^{test}) - F(x^c, y^c)$.
- 9: **if** $\exp\left(-\frac{\Delta}{T_k}\right) > \text{Random}[0, 1)$ **then**
- 10: Accept the candidate: $(x^c, y^c) \leftarrow (x^{test}, y^{test})$.
- 11: **until** $F(x^c, y^c)$ does not change after N loops
- 12: Let $(\bar{x}, \bar{y}) = (x^c, y^c)$.
- 13: Construct

$$X = \bigoplus_{\alpha=0}^D (\bar{x}_\alpha \otimes A^{\otimes\alpha}) \quad \text{and} \quad Y = \bigoplus_{\beta=0}^D (\bar{y}_\beta \otimes B^{\otimes\beta}).$$

- 14: **return** (X, Y) .

In the experiments, we set $N = 300$. Although this attack does not achieve a perfect success rate, it frequently recovers the majority of the entries of the secret key. The average number of recovered entries and the average execution time are respectively illustrated in Figures 2 and 3. The flowchart of the attacks based on the simulated annealing is shown in Figure 4.

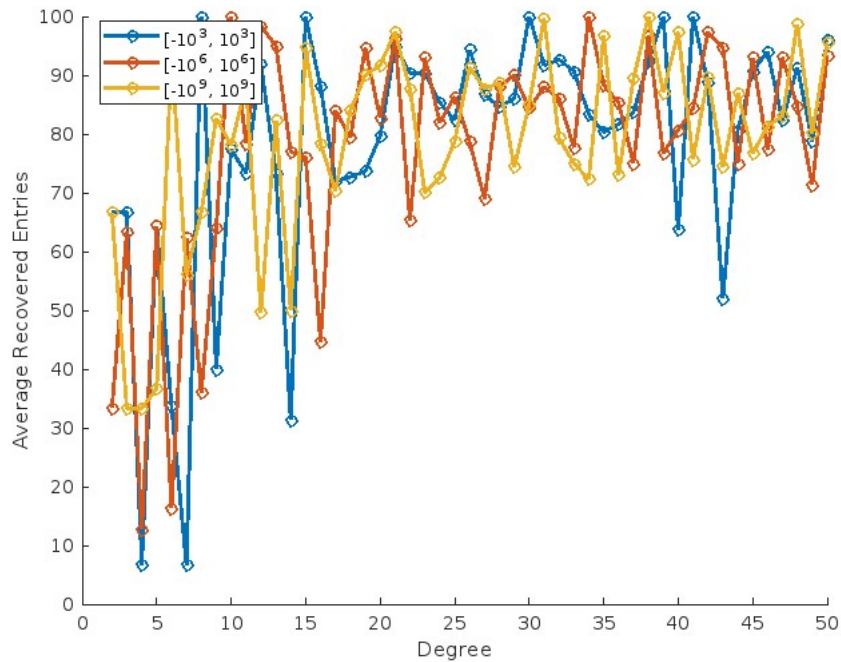


Figure 2. Attacking the max–min version of Protocol 1 using Algorithm 4: Recover entries.

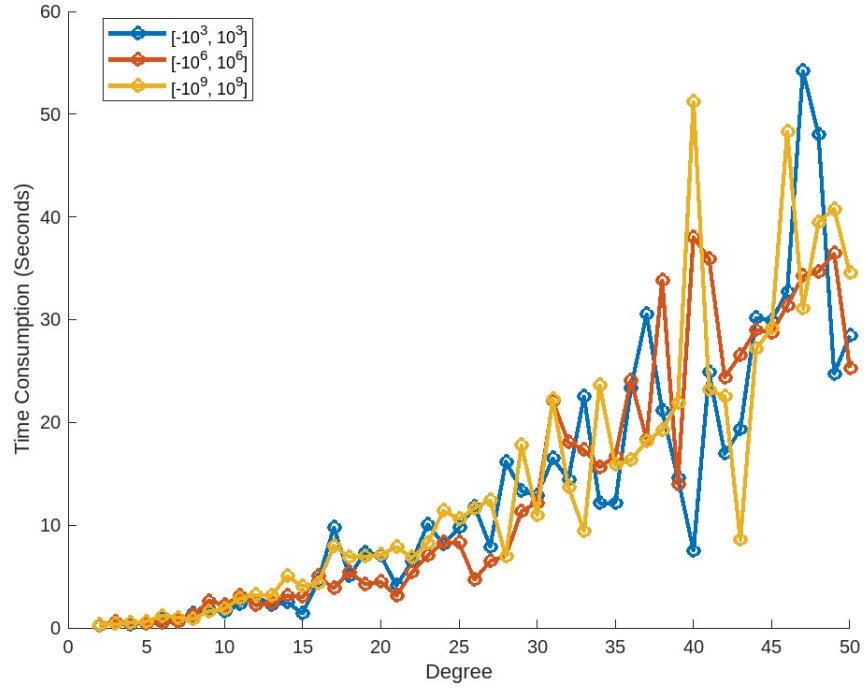


Figure 3. Attacking the max–min version of Protocol 1 using Algorithm 4: Time taken.

Note that this attack is significantly faster than Algorithm 2 (for detailed experimental results of Algorithm 2, refer to [5]). However, as shown experimentally, it does not guarantee the successful recovery of the entire secret key. Furthermore, the algorithm maintains consistent performance with higher entry ranges, largely due to the appropriate adjustment of the initial temperature.

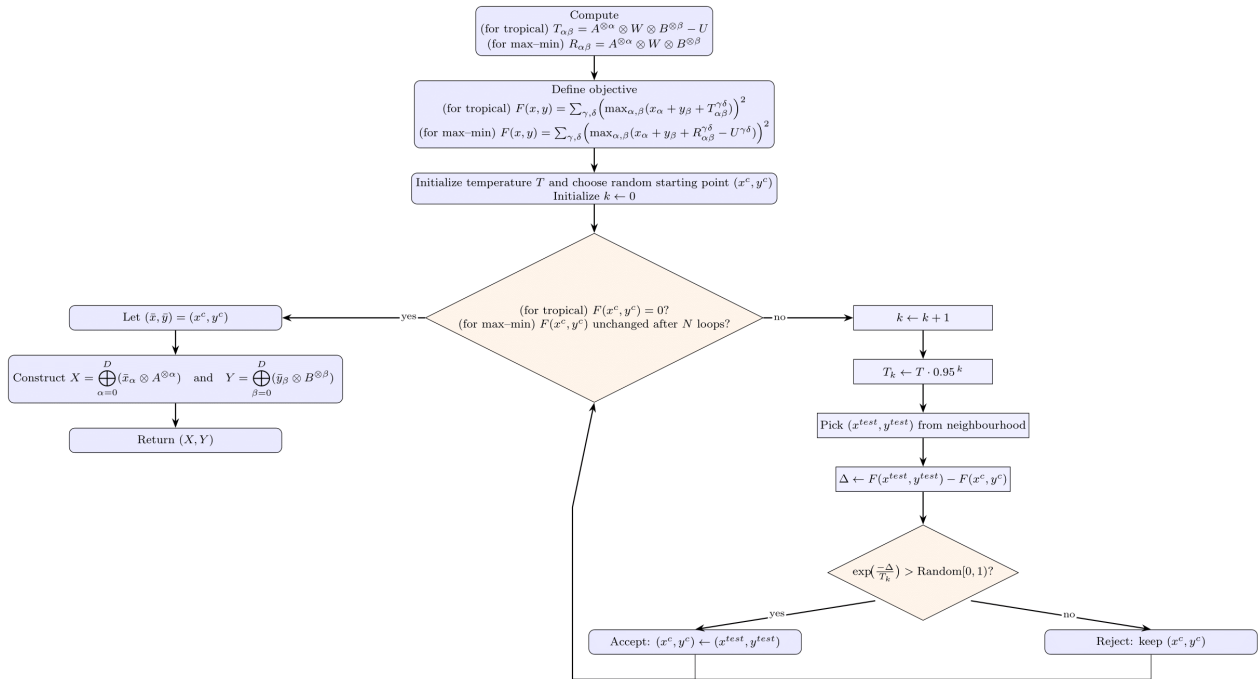


Figure 4. Flowchart of Algorithms 3 and 4.

3.2. Kotov-Ushakov Attack Using MILP Solver

We now propose an attack that recovers the secret key by solving a mixed integer linear program (MILP), following an observation by [22]. Specifically, we start by transforming

system (1) in the Kotov–Ushakov attack into a linear system by converting the disjunctive constraints into linear constraints by using Boolean variables and a big parameter. This approach allows us to avoid dealing with system (2) and the associated challenge of enumerating all minimal solutions. Then we solve this system of inequalities using the Gurobi solver [23] (but we could use any other available MILP solver instead) employing the default parameters of this solver. The tropical and max–min encodings of system (1) are presented in (8)–(10), respectively. See Algorithms 5 and 6 for a detailed description.

Algorithm 5 Kotov–Ushakov MILP attack on tropical Stickel protocol

Input: Public matrices A, B, W , transmitted message U , maximum polynomial degree D

Output: Matrices X, Y .

- 1: Compute $T_{\alpha\beta} = A^{\otimes\alpha} \otimes W \otimes B^{\otimes\beta} - U$ for all $0 \leq \alpha, \beta \leq D$.
- 2: Find x, y and z that satisfy the following system where M is a big enough number, α and β range from 0 to D , and γ and δ range from 1 to n :

$$\begin{aligned} x_\alpha + y_\beta + T_{\gamma\delta}^{\alpha\beta} &\leq 0 \quad \forall \alpha, \beta, \gamma, \delta, \\ x_\alpha + y_\beta + T_{\gamma\delta}^{\alpha\beta} + (1 - z_{\alpha\beta\gamma\delta})M &\geq 0 \quad \forall \alpha, \beta, \gamma, \delta, \\ z_{\alpha\beta\gamma\delta} &\in \{0, 1\} \quad \forall \alpha, \beta, \gamma, \delta, \\ \sum_{(\alpha, \beta)} z_{\alpha\beta\gamma\delta} &= 1 \quad \forall \gamma, \delta. \end{aligned} \tag{8}$$

- 3: Using these x and y construct

$$X = \bigoplus_{\alpha=0}^D (x_\alpha \otimes A^{\otimes\alpha}) \quad \text{and} \quad Y = \bigoplus_{\beta=0}^D (y_\beta \otimes B^{\otimes\beta}).$$

- 4: **return** (X, Y) .
-

The parameter M acts as a tunable variable whose value can be adjusted to ensure the correct and efficient solution of the MILP. In practice, we used a value of M that exceeded 1000 multiplied by the biggest possible entry of A, B , and W . Note that the number of variables in system (8) increases both with the matrix dimension and the polynomial degree used in the protocol. Specifically, the number of variables would be $2(D+1) + n^2(D+1)^2$. Also, the number of equations in this system is $2n^2(D+1)^2 + n^2$. Figure 5 illustrates the time taken by Algorithm 5 when applied to the tropical Stickel protocol.

The attack on the max–min version of Protocol 1 can be similarly described: see Algorithm 6.

Note that the number of variables in this system similarly increases with both the matrix dimension and the polynomial degree used in the protocol. Specifically, the number of variables is $2(D+1) + n^2(D+1)^2 + 3n^2(D+1)^2$. Also, the number of equations in this system is $7n^2(D+1)^2 + n^2$. The time taken by Algorithm 6 when applied to the max–min Stickel protocol is illustrated in Figure 6.

We observe that the computational time required for this approach is worse than that of the tropical case (Figure 5).

Therefore, both Algorithms 5 and 6 require significantly more time even for lower polynomial degrees compared to the tropical and max–min Kotov–Ushakov attacks (Algorithms 1 and 2). This is likely due to the high number of variables involved in the linear system. Consequently, these attacks do not provide any significant advantage over the previously described Kotov–Ushakov attacks.

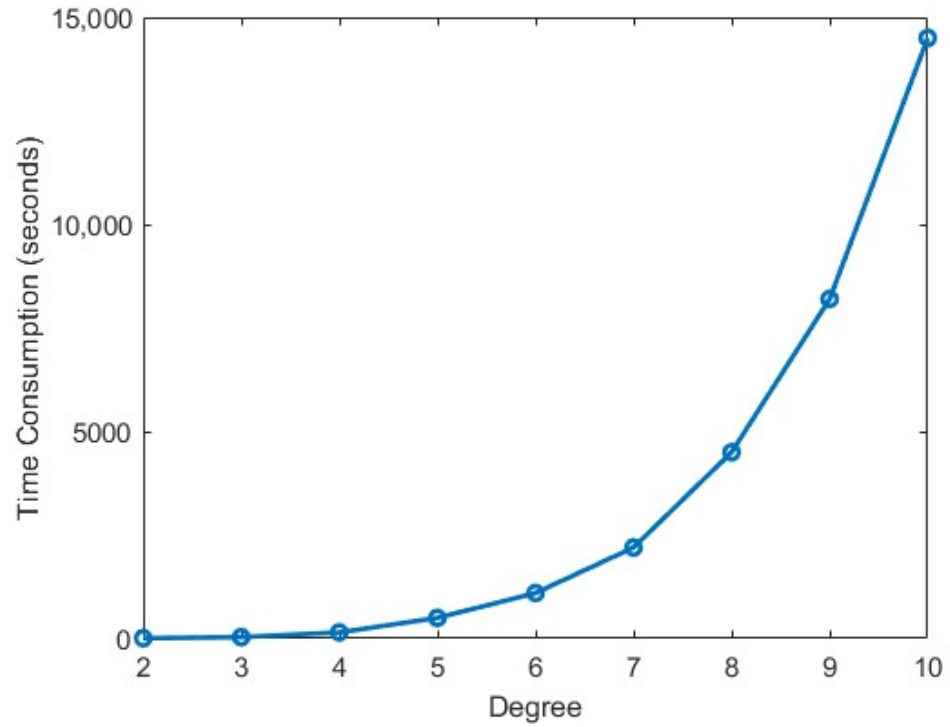


Figure 5. Attacking tropical version of Protocol 1 using Algorithm 5.

Algorithm 6 Kotov–Ushakov MILP attack on max–min Stickel protocol

Input: Public matrices A, B, W , transmitted message U , maximum polynomial degree D

Output: Matrices X, Y .

- 1: Compute $R_{\alpha\beta} = A^{\otimes\alpha} \otimes W \otimes B^{\otimes\beta}$ for all $0 \leq \alpha, \beta \leq D$.
- 2: Find x, y and z that satisfy the following system where M is a big enough number, α and β range from 0 to D , and γ and δ range from 1 to n :

$$\begin{aligned}
 x_\alpha - (1 - z_{\alpha\beta\gamma\delta}^{(1)})M &\leq U_{\gamma\delta} \\
 y_\beta - (1 - z_{\alpha\beta\gamma\delta}^{(2)})M &\leq U_{\gamma\delta} \\
 R_{\gamma\delta}^{\alpha\beta} - (1 - z_{\alpha\beta\gamma\delta}^{(3)})M &\leq U_{\gamma\delta} \\
 z_{\alpha\beta\gamma\delta}^{(i)} &\in \{0, 1\} \quad \text{and} \quad \sum_{i=1}^3 z_{\alpha\beta\gamma\delta}^{(i)} = 1
 \end{aligned} \tag{9}$$

$$\begin{aligned}
 x_\alpha + (1 - z_{\alpha\beta\gamma\delta})M &\geq U_{\gamma\delta} \\
 y_\beta + (1 - z_{\alpha\beta\gamma\delta})M &\geq U_{\gamma\delta} \\
 R_{\gamma\delta}^{\alpha\beta} + (1 - z_{\alpha\beta\gamma\delta})M &\geq U_{\gamma\delta} \\
 z_{\alpha\beta\gamma\delta} &\in \{0, 1\} \quad \text{and} \quad \sum_{(\alpha,\beta)} z_{\alpha\beta\gamma\delta} = 1
 \end{aligned} \tag{10}$$

- 3: Solve the MILP, and construct

$$X = \bigoplus_{\alpha=0}^D (x_\alpha \otimes A^{\otimes\alpha}) \quad \text{and} \quad Y = \bigoplus_{\beta=0}^D (y_\beta \otimes B^{\otimes\beta}).$$

- 4: **return** (X, Y) .
-

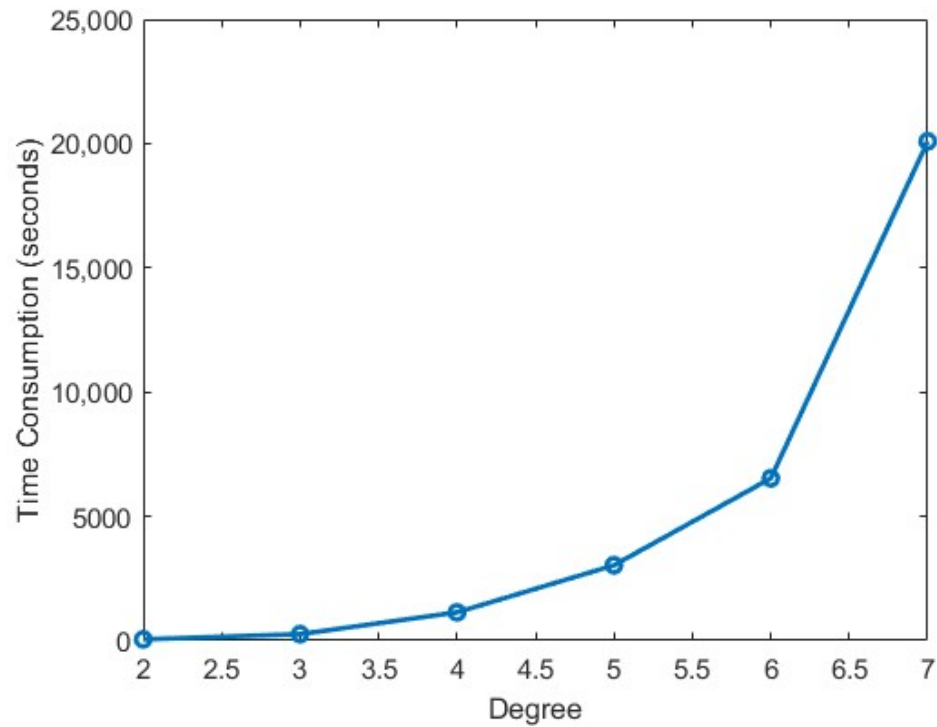


Figure 6. Attacking max–min version of Protocol 1 using Algorithm 6.

3.3. Shpilrain Attack Using MILP Solver

We now propose an alternative method to formulate the MILP to attack the tropical and max-min implementations of Protocol 1. Specifically, we introduce the tropical and max-min versions of the Shpilrain attack [2], where our objective is to find X and Y such that

$$\begin{cases} X \otimes A = T \\ A \otimes X = T \\ Y \otimes B = R \\ B \otimes Y = R \\ X \otimes W \otimes Y = U \end{cases} \quad (11)$$

where matrices T and R are composed of newly introduced auxiliary variables t_{ij}, r_{ij} for $(i, j) \in [n] \times [n]$. Then, the MILP can similarly be formulated by converting the disjunctive constraints into linear constraints with Boolean variables. In particular, for the first equation of (11), with a_{ij} being the entries of A , we have

$$\max_{k \in [n]} (x_{ik} \otimes a_{kj}) = t_{ij} \quad \forall (i, j) \in [n] \times [n],$$

which can be represented as the following set of inequalities

$$x_{ik} \otimes a_{kj} \leq t_{ij} \quad \forall i, j, k \in [n],$$

and with M being a sufficiently large number

$$x_{ik} \otimes a_{kj} + (1 - z_{kij})M \geq t_{ij} \quad \forall i, j, k \in [n],$$

$$\sum_k z_{kij} = 1, \quad z_{kij} \in \{0, 1\} \quad \forall i, j, k \in [n].$$

The rest of inequalities can similarly be formulated using the other equations in (11), and then we solve the system using MILP solver. The tropical and max–min versions of the attack are described below in Algorithms 7 and 8. We observe that the number of variables in the system increases only with the matrix dimension, but not the polynomial degree used in the protocol. Specifically, for the tropical case, the number of variables in this system is $4n^2 + 4n^3 + n^4$, and the number of equations is $5n^2 + 8n^3 + 2n^4$. For the max–min case, the number of variables is $4n^2 + 12n^3 + 4n^4$, and the number of equations is $5n^2 + 20n^3 + 7n^4$. The tropical and max–min encodings of system (11) are displayed in (12)–(16) and (17)–(21), respectively.

Algorithm 7 MILP Shpilrain attack on tropical Stickel protocol

Input: Public matrices A, B, W , transmitted message U

Output: Matrices X, Y .

1: Represent (11) (over the tropical semiring) by the following system:

$$\begin{aligned} x_{ik} + a_{kj} &\leq t_{ij} \quad \forall i, j, k \in [n], \\ x_{ik} + a_{kj} + (1 - z_{1kij})M &\geq t_{ij} \quad \forall i, j, k \in [n], \\ z_{1kij} &\in \{0, 1\}, \quad \forall i, j, k \in [n], \\ \sum_k z_{1kij} &= 1 \quad \forall i, j \in [n], \end{aligned} \quad (12)$$

$$\begin{aligned} a_{ik} + x_{kj} &\leq t_{ij} \quad \forall i, j, k \in [n], \\ a_{ik} + x_{kj} + (1 - z_{2kij})M &\geq t_{ij} \quad \forall i, j, k \in [n], \\ z_{2kij} &\in \{0, 1\}, \quad \forall i, j, k \in [n], \\ \sum_k z_{2kij} &= 1 \quad \forall i, j \in [n], \end{aligned} \quad (13)$$

$$\begin{aligned} y_{ik} + b_{kj} &\leq r_{ij} \quad \forall i, j, k \in [n], \\ y_{ik} + b_{kj} + (1 - z_{3kij})M &\geq r_{ij} \quad \forall i, j, k \in [n], \\ z_{3kij} &\in \{0, 1\} \quad \forall i, j, k \in [n], \\ \sum_k z_{3kij} &= 1 \quad \forall i, j \in [n], \end{aligned} \quad (14)$$

$$\begin{aligned} b_{ik} + y_{kj} &\leq r_{ij} \quad \forall i, j, k \in [n], \\ b_{ik} + y_{kj} + (1 - z_{4kij})M &\geq r_{ij} \quad \forall i, j, k \in [n], \\ z_{4kij} &\in \{0, 1\}, i, j, k \in [n], \\ \sum_k z_{4kij} &= 1 \quad \forall i, j \in [n], \end{aligned} \quad (15)$$

$$\begin{aligned} x_{ik} + w_{kl} + y_{lj} &\leq u_{ij} \quad \forall i, j, k, l \in [n], \\ x_{ik} + w_{kl} + y_{lj} + (1 - z_{5kl ij})M &\geq u_{ij} \quad \forall i, j, k, l \in [n], \\ z_{5kl ij} &\in \{0, 1\}, \\ \sum_{k,l} z_{5kl ij} &= 1 \quad \forall i, j \in [n], \end{aligned} \quad (16)$$

where a_{ij}, b_{ij}, w_{ij} are, respectively, the entries of the public matrices A, B, W , and x_{ij}, y_{ij} are the variables of the system.

2: Solve the MILP, and construct $X = (x_{ij})$ and $Y = (y_{ij})$.

3: **return** (X, Y) .

Algorithm 8 MILP Shpilrain attack on max–min Stickel protocol

Input: Public matrices A, B, W , transmitted message U

Output: Matrices X, Y .

1: Represent (11) (over the max–min semiring) by the following system

$$\begin{aligned}
 x_{ik} - (1 - z_{1kij}^{(1)})M &\leq t_{ij} \quad \forall i, j, k \in [n], \\
 a_{kj} - (1 - z_{1kij}^{(2)})M &\leq t_{ij} \quad \forall i, j, k \in [n], \\
 z_{1kij}^{(1)} + z_{1kij}^{(2)} &= 1 \quad \forall i, j, k \in [n], \\
 x_{ik} + (1 - z_{1kij}^{(3)})M &\geq t_{ij} \quad \forall i, j, k \in [n], \\
 a_{kj} + (1 - z_{1kij}^{(3)})M &\geq t_{ij} \quad \forall i, j, k \in [n], \\
 z_{1kij}^{(1)}, z_{1kij}^{(2)}, z_{1kij}^{(3)} &\in \{0, 1\} \quad \forall i, j, k \in [n] \\
 \sum_k z_{1kij}^{(3)} &= 1 \quad \forall i, j \in [n],
 \end{aligned} \tag{17}$$

$$\begin{aligned}
 a_{ik} - (1 - z_{2kij}^{(1)})M &\leq t_{ij} \quad \forall i, j, k \in [n], \\
 x_{kj} - (1 - z_{2kij}^{(2)})M &\leq t_{ij} \quad \forall i, j, k \in [n], \\
 z_{2kij}^{(1)} + z_{2kij}^{(2)} &= 1 \quad \forall i, j, k \in [n], \\
 a_{ik} + (1 - z_{2kij}^{(3)})M &\geq t_{ij} \quad \forall i, j, k \in [n], \\
 x_{kj} + (1 - z_{2kij}^{(3)})M &\geq t_{ij} \quad \forall i, j, k \in [n], \\
 z_{2kij}^{(1)}, z_{2kij}^{(2)}, z_{2kij}^{(3)} &\in \{0, 1\} \quad \forall i, j, k \in [n] \\
 \sum_k z_{2kij}^{(3)} &= 1 \quad \forall i, j \in [n],
 \end{aligned} \tag{18}$$

$$\begin{aligned}
 y_{ik} - (1 - z_{3kij}^{(1)})M &\leq r_{ij} \quad \forall i, j, k \in [n], \\
 b_{kj} - (1 - z_{3kij}^{(2)})M &\leq r_{ij} \quad \forall i, j, k \in [n], \\
 z_{3kij}^{(1)} + z_{3kij}^{(2)} &= 1 \quad \forall i, j, k \in [n], \\
 y_{ik} + (1 - z_{3kij}^{(3)})M &\geq r_{ij} \quad \forall i, j, k \in [n], \\
 b_{kj} + (1 - z_{3kij}^{(3)})M &\geq r_{ij} \quad \forall i, j, k \in [n], \\
 z_{3kij}^{(1)}, z_{3kij}^{(2)}, z_{3kij}^{(3)} &\in \{0, 1\} \quad \forall i, j, k \in [n] \\
 \sum_k z_{3kij}^{(3)} &= 1 \quad \forall i, j \in [n],
 \end{aligned} \tag{19}$$

$$\begin{aligned}
 b_{ik} - (1 - z_{4kij}^{(1)})M &\leq r_{ij} \quad \forall i, j, k \in [n], \\
 y_{kj} - (1 - z_{4kij}^{(2)})M &\leq r_{ij} \quad \forall i, j, k \in [n], \\
 z_{4kij}^{(1)} + z_{4kij}^{(2)} &= 1 \quad \forall i, j, k \in [n], \\
 b_{ik} + (1 - z_{4kij}^{(3)})M &\geq r_{ij} \quad \forall i, j, k \in [n], \\
 y_{kj} + (1 - z_{4kij}^{(3)})M &\geq r_{ij} \quad \forall i, j, k \in [n], \\
 z_{4kij}^{(1)}, z_{4kij}^{(2)}, z_{4kij}^{(3)} &\in \{0, 1\} \quad \forall i, j, k \in [n] \\
 \sum_k z_{4kij}^{(3)} &= 1 \quad \forall i, j \in [n],
 \end{aligned} \tag{20}$$

$$\begin{aligned}
 x_{ik} - (1 - z_{5kl ij}^{(1)})M &\leq u_{ij} \quad \forall i, j, k, l \in [n], \\
 w_{kl} - (1 - z_{5kl ij}^{(2)})M &\leq u_{ij} \quad \forall i, j, k, l \in [n], \\
 y_{lj} - (1 - z_{5kl ij}^{(3)})M &\leq u_{ij} \quad \forall i, j, k, l \in [n], \\
 z_{5kl ij}^{(1)} + z_{5kl ij}^{(2)} + z_{5kl ij}^{(3)} &= 1 \quad \forall i, j, k, l \in [n], \\
 x_{ik} + (1 - z_{5kl ij}^{(4)})M &\geq u_{ij} \quad \forall i, j, k, l \in [n], \\
 w_{kl} + (1 - z_{5kl ij}^{(4)})M &\geq u_{ij} \quad \forall i, j, k, l \in [n], \\
 y_{lj} + (1 - z_{5kl ij}^{(4)})M &\geq u_{ij} \quad \forall i, j, k, l \in [n], \\
 z_{5kl ij}^{(1)}, z_{5kl ij}^{(2)}, z_{5kl ij}^{(3)}, z_{5kl ij}^{(4)} &\in \{0, 1\}, \\
 \sum_{k,l} z_{5kl ij}^{(4)} &= 1 \quad \forall i, j \in [n],
 \end{aligned} \tag{21}$$

where a_{ij}, b_{ij}, w_{ij} are, respectively, the entries of the public matrices A, B, W , and x_{ij}, y_{ij} are the variables of the system.

2: Solve the MILP, and construct $X = (x_{ij})$ and $Y = (y_{ij})$.

3: return (X, Y) .

Note that a distinct advantage of these attacks is that they are independent of the polynomial degree used in the protocol. Therefore, Alice and Bob cannot improve the protocol's resistance against these attacks by increasing the polynomial degree, a way that is very effective against Kotov–Ushakov attack and its max–min analog (Algorithms 1 and 2). In other words, the limitations for MILP Shpilrain attacks fully depend on the MILP techniques being used, but it is inevitable that the memory usage blows up as the matrix dimensions increase, due to the high number of equations and hence variables involved in the linear program. Figure 7 shows the time taken by Algorithm 7 for different polynomial degrees.

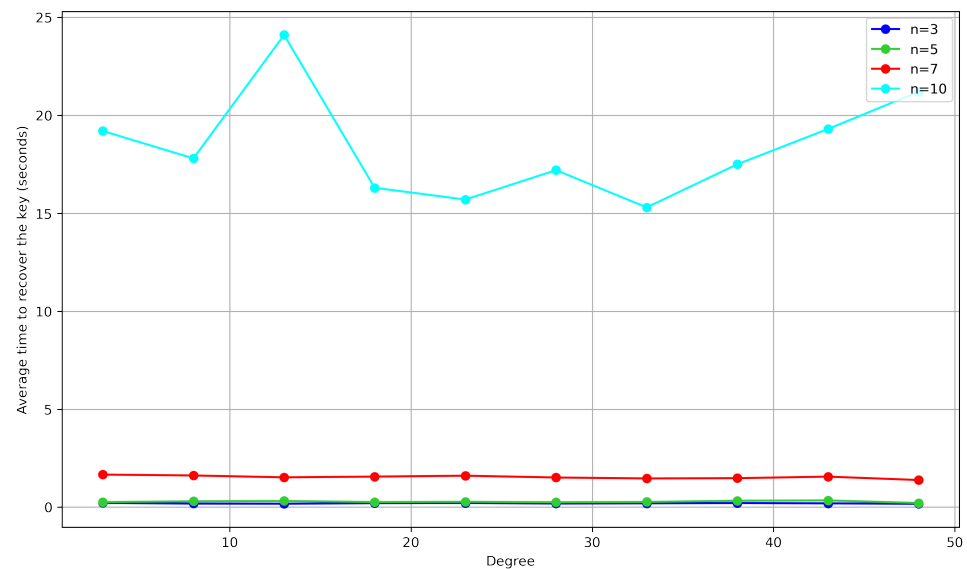


Figure 7. Attacking tropical version of Protocol 1 using Algorithm 7.

As illustrated in Figure 7, this attack is much faster than Algorithm 1 and maintains consistent computational efficiency across varying polynomial degrees. It is worth noting that for larger matrix dimensions, such as $n = 10$ or higher, the Gurobi solver may encounter challenges in directly solving the system in some trials. Fine-tuning of the solver parameters is required to solve the system in such cases. The time taken by Algorithm 8 for different polynomial degrees is shown in Figure 8. Note that due to the higher number of equations and variables in the max–min case compared with the tropical case, the memory required for encoding the linear program for a dimension higher than 8 would exceed the available memory threshold.

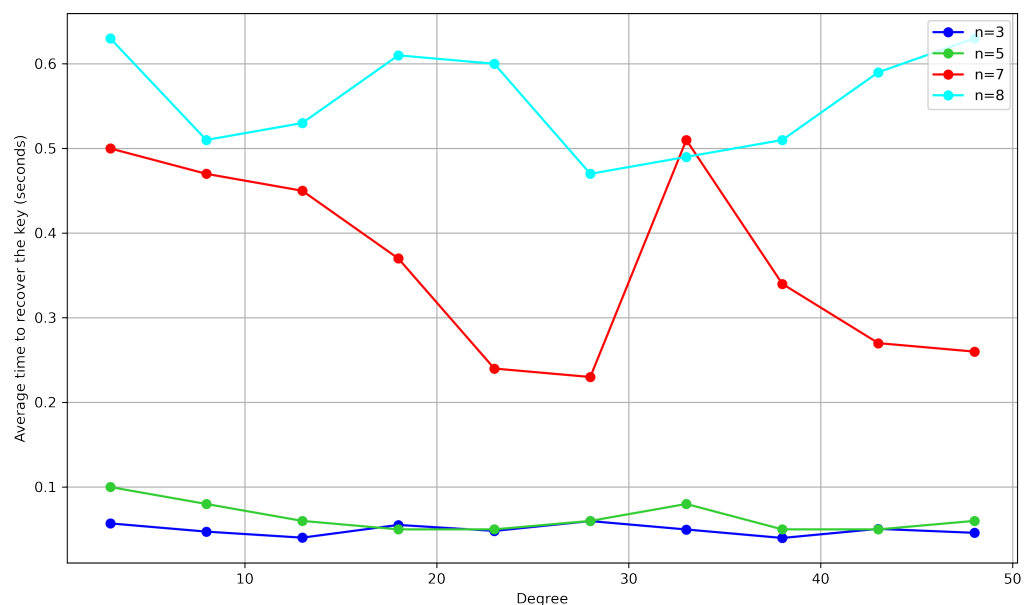
We now summarize the performance of the suggested attacks in Table 3. Here, note that our conclusion on the computational efficiency is based on the numerical experiments (see Figures 1, 3 and 5–8). The required assumptions and some other notes are summarized in Table 4.

Table 3. Comparative performance of the algorithms.

Algorithm	Semiring	Computational Efficiency	Memory Use	Empirical Success
1 Kotov–Ushakov (Algorithm 1)	Tropical	Inefficient	Low	100%
2 Kotov–Ushakov (Algorithm 2)	Max–min	Inefficient	Low	100%
3 Simulated Annealing (Algorithm 3)	Tropical	Efficient (most cases)	Low	100%
4 Simulated Annealing (Algorithm 4)	Max–min	Efficient (most cases)	Low	<100%
5 MILP Kotov–Ushakov (Algorithm 5)	Tropical	Inefficient	Low	100%
6 MILP Kotov–Ushakov (Algorithm 6)	Max–min	Inefficient	Low	100%
7 MILP Shpilrain (Algorithm 7)	Tropical	Efficient	High	100%
8 MILP Shpilrain (Algorithm 8)	Max–min	Efficient	High	100%

Table 4. Constraints, assumption, and notes.

Algorithm	Constraints, Assumptions, and Notes
Kotov–Ushakov (Algorithms 1 and 2)	<ul style="list-style-type: none"> Public matrices A, B, W and transmitted U are assumed known. Require D to be larger than the actual maximum degree of polynomials used by Alice and Bob.
Simulated Annealing (Algorithms 3 and 4)	<ul style="list-style-type: none"> Public matrices A, B, W and transmitted U are assumed known. Require D to be larger than the actual maximum degree of polynomials used by Alice and Bob. Sufficiently large initial temperature parameter. Larger range of entries and coefficients increases the observed execution time. Stopping criteria: the objective function reaches 0 (tropical); the objective function does not change after a specified number of loops (max–min).
MILP Kotov–Ushakov (Algorithms 5 and 6)	<ul style="list-style-type: none"> Public matrices A, B, W and transmitted U are assumed known. Require D larger than the actual maximum degree of polynomials used by Alice and Bob. Large value of the parameter M must be chosen as it affects correctness and numerical stability. Use a MILP solver.
MILP Shpilrain (Algorithms 7 and 8)	<ul style="list-style-type: none"> Public matrices A, B, W and transmitted U are assumed known. Large value of the parameter M must be chosen as it affects correctness and numerical stability. Independent of the actual maximum degree of polynomials used by Alice and Bob. Use a MILP solver. Require substantial memory.

**Figure 8.** Attacking max–min version of Protocol 1 using Algorithm 8.

4. Attacking Stickel’s Protocol over Digital Semiring

A recent implementation of Stickel protocol (Protocol 1) was introduced by [8], which employs a newly defined semiring referred to by the authors as the “digital semiring”. The authors claim that this new implementation of Stickel protocol resists the known attacks

such as the Kotov–Ushakov attack. Let us discuss how the methods outlined in this paper as well as those in [5] can be applied in this new situation.

The digital semiring of [8], which we here denote by $\mathbb{N}(\vee, \wedge)$, is defined over the set of natural numbers \mathbb{N} with adjoined $+\infty$, and is based on an unconventional order relation defined by

$$a \preceq b \Leftrightarrow \begin{cases} (a) \leq (b), & \text{if } (a) \neq (b), \\ a \leq b, & \text{if } (a) = (b), \end{cases} \quad (22)$$

where (a) denotes the sum of digits of $a \in \mathbb{N}$. It is understood that the sum of digits of $+\infty$ is $+\infty$, so this is the greatest element of the semiring. Based on this order relation, we then define the new addition $a \oplus b$ as the greatest element (also denoted as $a \vee b$) among a, b with respect to this order relation, and $a \otimes b$ as the smallest element (also denoted as $a \wedge b$) among a, b with respect to this order relation.

For the practical purposes of software implementation, Alice and Bob are always limited by a big enough number M , and therefore they would actually be using a semiring of the form $\mathbb{N}_M(\vee, \wedge)$ similarly defined using (22) over the natural numbers not exceeding M . However, it then can be shown that this semiring $\mathbb{N}_M(\vee, \wedge)$ is isomorphic to the semiring $\mathbb{N}_M(\max, \min)$, which is the set of natural numbers not exceeding M for which the operations are defined by $a \oplus b = \max(a, b)$ and $a \otimes b = \min(a, b)$. Indeed, the isomorphism is given by the mapping $f: \mathbb{N}_M(\vee, \wedge) \mapsto \mathbb{N}_M(\max, \min)$, for which

$$f(a) = \begin{cases} 0, & \text{if } a = 0, \\ \sum_{i=1}^{(a)-1} |[i]_{\leq M}| + |[a]_{\leq a}|, & \text{otherwise.} \end{cases} \quad (23)$$

where $[i]_{\leq a}$, for natural a, i such that $0 \leq i, a \leq M$, denotes the set of natural numbers whose sum of digits is equal to i and which do not exceed a , and $|[i]_{\leq a}|$ denotes the number of elements in this set.

Consequently, the attacks on the max–min semiring implementation of Stickel protocol discussed in this paper are equally applicable to the digital semiring implementation, due to the known limitations of Alice and Bob and the isomorphism given by (23). This also includes the guaranteed attack described in [5] (the max–min version of Kotov–Ushakov attack). Thus, the attacker only needs to take one additional step to exploit this isomorphism. A possible approach for such exploitation is to group the elements of the digital semiring by their digit sums, arranging the groups and the numbers within each group in ascending order. Each element in the digital semiring is then mapped to a corresponding element in the max–min semiring with the natural order from smallest to largest. The resulting algorithm has complexity at most $O(M \log_{10} M)$ since we have to go through each number and compute the sum of its digits (which has complexity not exceeding $O(\log_{10} M)$).

Figure 9 illustrates the computational time needed to execute it for different maximum values M .

As shown in Figure 9, the computational time required for this isomorphism mapping is relatively minor, but it obviously increases as Alice and Bob agree on higher ranges. However, it can be argued that they cannot extend these ranges indefinitely due to the risk of potential numerical instability. Thus, while attacking the Stickel protocol over the digital semiring involves this additional computational overhead, it is a one-time setup and does not affect the computational time during individual attack sessions since it should only be pre-computed once. Therefore, to keep the paper more concise, we have not included numerical experiments for attacking the Stickel protocol over the digital semiring, as these would be identical to the experiments on attacking the Stickel protocol over the max–min semiring described in the previous section and in [5]. We also note that a different attack

on the Stickel protocol over digital semiring has been recently published in [24], which develops a branch and bound approach and exploits the structure of the circulant matrices involved in the protocol.

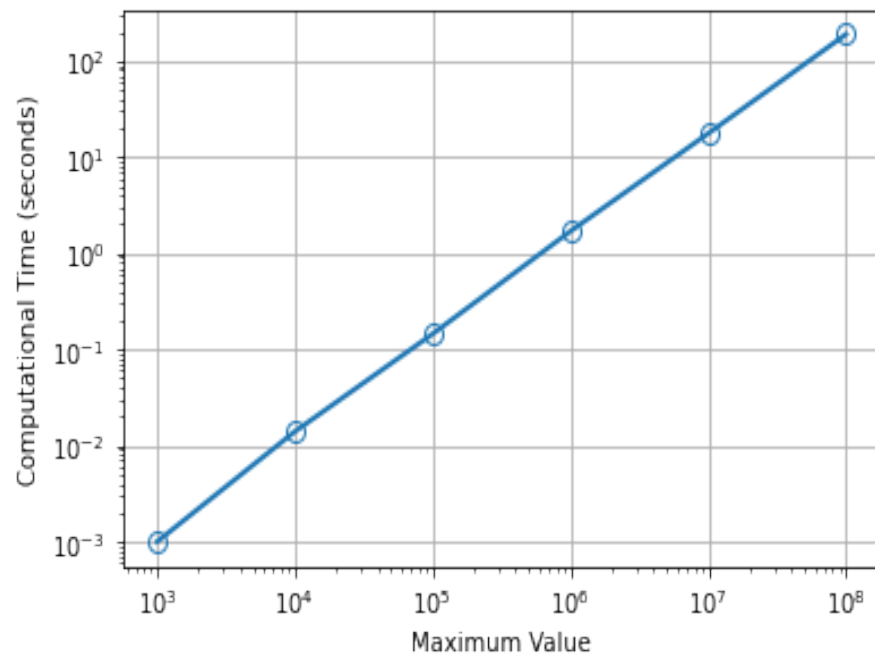


Figure 9. Digital semiring pre-computation.

5. Forging the Tropical Signatures

A digital signature protocol based on the hardness of tropical polynomial factorization was proposed in [9]. Several heuristics to attack this protocol have been proposed in [25,26]. These heuristics primarily focus on generating a valid forged signature from a previously legitimate signature. To counter these attacks, along with other trivial forgeries, a revised version of the protocol has also been introduced. In this section, we present new attacks that directly target the public key, which also apply to the revised version, as the public key is unchanged. In what follows, we present the protocol and how it can be attacked. To read this section, the basic knowledge of semiring algebra (see Definitions 1–4) will be required from the reader, but only the tropical semiring \mathbb{R}_{\max} will be used (see Table 1 for a concise summary).

Protocol 2 (The tropical digital signatures [9]).

Private Key: Two tropical polynomials X, Y , with integer coefficients from $[0, r]$ and the sum of their degrees is $2d$.

Public Key: r and d , and the multiplication of the two secret polynomials $M = X \otimes Y$.

Signing:

1. Compute the hash of the message, and use it to form the tropical polynomial H using a known deterministic procedure.
2. Select random private polynomials U, V such that $\deg(U) = \deg(Y)$ and $\deg(V) = \deg(X)$, with coefficients in $[0, r]$, and let $N = U \otimes V$.
3. The signature is the tuple $(H, H \otimes X \otimes U, H \otimes Y \otimes V, N)$.

Verification:

1. Compute H as in the first step of signing, and verify it.
2. Verify that $\deg(H \otimes X \otimes U) = \deg(H \otimes Y \otimes V) = 3d$ and $\deg(N) = 2d$.

3. Verify that neither $H \otimes X \otimes U$ nor $H \otimes Y \otimes V$ is a tropical constant multiple of $H \otimes M$ or $H \otimes N$.
4. Verify that coefficients of $H \otimes X \otimes U$ and $H \otimes Y \otimes V$ are within $[0, 3r]$ and those of N are within $[0, 2r]$.
5. Compute $W = (H \otimes X \otimes U) \otimes (H \otimes Y \otimes V)$, and accept the signature if and only if $W = H \otimes H \otimes M \otimes N$.

The security of this protocol relies on the hardness of tropical polynomial factorization, which was shown to be NP-hard [27]. This problem can be formulated as follows:

Problem 3 (Tropical Polynomial Factorization). *Given a tropical polynomial $M = X \otimes Y$, find X and Y .*

At first glance, it might seem straightforward to factor M using the tropical fundamental theorem of algebra [28], which states that any tropical polynomial can be easily factored into exactly linear polynomials. Let us explore this theorem formally.

Theorem 1 (Tropical fundamental theorem of algebra [28]). *Any tropical polynomial of degree n*

$$M(t) = \bigoplus_{i=0}^n (m_i \otimes t^{\otimes i})$$

can be efficiently factored into linear factors. Specifically, there exists a constant c and roots r_1, r_2, \dots, r_n such that

$$M(t) = c \otimes \left(\bigotimes_{i=1}^n (t \oplus r_i) \right),$$

The roots r_i are the points where the piecewise-linear function $M(t)$ changes slope. This factorization provides a canonical form of $M(t)$ as a function.

Note that the factorization from this theorem is a functional factorization, meaning $M(t)$ holds for all t as a function. However, it does not necessarily preserve the original coefficient sequence (m_0, m_1, \dots, m_n) of M . That is, the string of coefficients obtained from this factorization is a canonical (most reduced) form of the tropical polynomial. However, this canonical form, while equivalent to the original polynomial as a function, does not necessarily preserve the initial polynomial's sequence of coefficients.

In contrast, a sequence-based factorization requires finding X and Y such that their polynomial multiplication matches the original coefficients of M , where the coefficients m_k of M are as follows:

$$m_k = \bigoplus_{(i,j): i+j=k} (x_i \otimes y_j) = \max_{i+j=k} (x_i + y_j), \quad k = 0, 1, \dots, n.$$

Therefore, the security of Problem 3 relies on factoring M as a sequence (i.e., string of numbers), a problem shown to be NP-hard. Factoring M as a function does not generally preserve the original sequence, which most likely causes the original sequence recovery to fail. That is, a function-based factorization yield factors that satisfy the same maximum operations but do not necessarily reconstruct the original sequence of coefficients. In contrast, a sequence-based factorization requires that multiplying the factors exactly reproduces the original coefficients of M . As such, it is required for the attacks on Problem 3 to target a “sequence-based” factoring of M , where the multiplication of the factors exactly recovers the original coefficients of M .

Note that there are possibly many factorizations of M , meaning the original factors X and Y are not generally unique. Therefore, for the attacker's purpose of producing a valid forged signature in Protocol 2, it is sufficient to find any factors that pass the verification process. This non-uniqueness in factorization can be exploited as a basis for some heuristic attacks. Thus, in the proposed attacks that follow, the attacker's objective is to find X' and Y' such that $X' \otimes Y' = M$, with the additional constraints that their degrees sum to $2d$ and their coefficients are from $[0, r]$, so they can pass the verification process. Successfully finding X' and Y' enables the attacker to impersonate the signer and hence produce a valid signature to any arbitrary message. Specifically, with H being the polynomial formed from an arbitrary hashed message, and choosing U and V with $\deg(U) = \deg(Y')$ and $\deg(V) = \deg(X')$, with coefficients in $[0, r]$, the forged signature $(H, H \otimes X' \otimes U, H \otimes Y' \otimes V, N = U \otimes V)$ is verified correctly, as all of the above verification steps clearly hold, and it is highly unlikely that the second and third polynomials of this tuple will be shifted versions of the public polynomials $H \otimes M$ or $H \otimes N$, respectively. We now propose two attacks utilizing this approach.

- **Kotov–Ushakov-based attack**

Note that M essentially represents a convolution of the two sequences X and Y , with max-plus operations. This allows the problem to be formulated as a one-sided linear system using matrices, by treating each product of the secret coefficients as a variable. However, the length of the original sequences is unknown. Consequently, the attack must iterate over possible lengths for X' until a suitable solution to the one-sided linear system is found.

Formally, we know that each coefficient m_k of $M = X \otimes Y$ can be represented as

$$m_k = \bigoplus_{(i,j): i+j=k} x_i \otimes y_j,$$

where m_k , x_i , and y_j denote the coefficients of the polynomials M , X , and Y , respectively. Then, with x_i and y_j being the unknowns, this system can be equivalently written as the linear system $A \otimes z = b$, where A is a binary matrix that indicates which variables are present in the k -th equation, z is the vector of unknowns with each element $z_{ij} = x_i \otimes y_j$, and b is the vector containing the known coefficients of M . The following example shows an illustration of this representation.

Example 1 (One-sided linear system representation of polynomial multiplication). *For a polynomial M of degree 4, and polynomials X and Y each of degree 2, the polynomial multiplication $M = X \otimes Y$ can be represented as the following linear system:*

$$\begin{bmatrix} 0 & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty \\ -\infty & 0 & -\infty & 0 & -\infty & -\infty & -\infty & -\infty & -\infty \\ -\infty & -\infty & 0 & -\infty & 0 & -\infty & 0 & -\infty & -\infty \\ -\infty & -\infty & -\infty & -\infty & -\infty & 0 & -\infty & 0 & -\infty \\ -\infty & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty & -\infty & 0 \end{bmatrix} \otimes \begin{bmatrix} x_0 \otimes y_0 \\ x_0 \otimes y_1 \\ x_0 \otimes y_2 \\ x_1 \otimes y_0 \\ x_1 \otimes y_1 \\ x_1 \otimes y_2 \\ x_2 \otimes y_0 \\ x_2 \otimes y_1 \\ x_2 \otimes y_2 \end{bmatrix} = \begin{bmatrix} m_0 \\ m_1 \\ m_2 \\ m_3 \\ m_4 \end{bmatrix}$$

Thus, the attacker's goal is to find a solution to this linear system. That is, a solution r_{ij} that satisfies $r_{ij} = x_i \otimes y_j$ for all $i \in \{0, 1, \dots, d_x\}$ and $j \in \{0, 1, \dots, d_y\}$, for some x_i and y_j . Additional constraints must be imposed on x_i and y_j to ensure that the forged signature is verified correctly. These constraints are $x_i, y_j \in [0, r]$ and $d_x + d_y = 2d$, where

r and d are public parameters of the protocol. Note that this system is not guaranteed to have a solution unless d_x equals the original degree of the polynomial X , but this degree is secret. Consequently, the attacker must test multiple values of d_x until a solution is found. However, it is possible that a solution can be found even when d_x differs from the original degree of X due to the possible non-unique factorization of M . The attack is formally described below.

Figure 10 presents the performance of this attack when $t = 3$, showing the success rate and computational time over 10 trails for multiple values of d . Note that, for all numerical experiments, the degree of X in the protocol instance is chosen as specified by the authors, i.e., randomly selected from the interval $[\frac{3}{4}d, \frac{5}{4}d]$. The degree of Y is then determined accordingly, as the sum of the degrees of X and Y must equal $2d$.

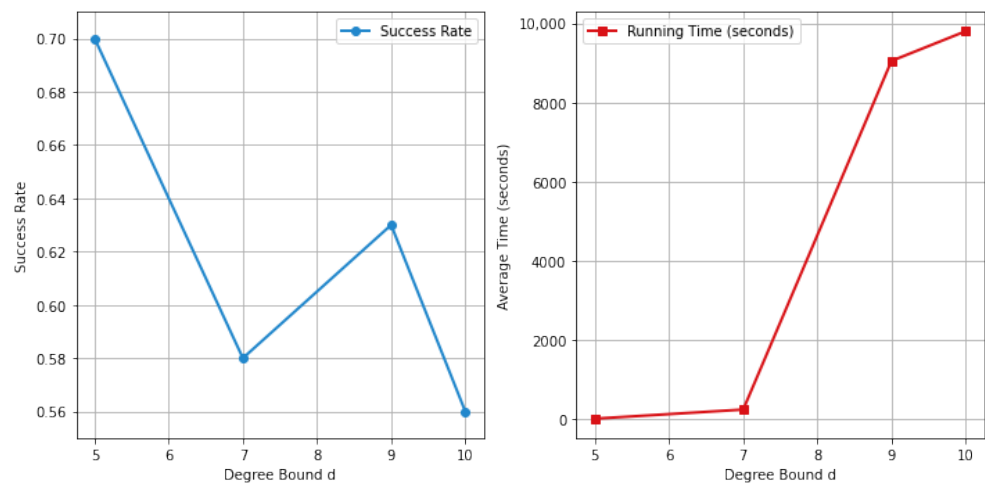


Figure 10. Success rate and computational time of Algorithm 9.

Algorithm 9 Kotov–Ushakov-based attack on Protocol 2

Input: Public key polynomial M , signature parameters t, r , degree bound $2d$.

Output: Recovered factors X', Y' .

- 1: **for** $dx = 1$ **to** t **do**
- 2: Set $dy = 2d - dx$.
- 3: Construct binary matrix A and vector b for the linear system as in Example 1.
- 4: Compute the greatest solution $c_{ij} = \min_i(b_i - A_{ij})$ and the sets $S_{ij} = \arg \min_i(b_i - A_{ij})$ for all $i \in \{0, \dots, d_x\}$ and $j \in \{0, \dots, d_y\}$.
- 5: Among all minimal covers of $\{0, 1, \dots, 2d\}$ by S_{ij} , that is, all minimal subsets $\mathcal{C} \subseteq \{0, 1, \dots, 2d\}$ such that

$$\bigcup_{(i,j) \in \mathcal{C}} S_{ij} = \{0, 1, \dots, 2d\},$$

find a cover for which the system

$$\begin{aligned} x_i + y_j &= c_{ij}, & \text{if } (i, j) \in \mathcal{C}, \\ x_i + y_j &\leq c_{ij}, & \text{otherwise,} \\ x_i, y_j &\in [0, r]. \end{aligned}$$

is solvable.

- 6: If a solution is found, break the loop. If no solution is found, proceed to the next d_x until a solution is found.
 - 7: Construct the polynomials X' and Y' using the derived x_i and y_j , respectively.
 - 8: **return** (X', Y') .
-

While the attack achieves a considerable success rate, its efficiency is limited, even for short polynomial lengths, due to the large number of enumerated minimal covers. Therefore, it is impractical for the recommended protocol parameters ($d = 150$).

- **Mixed-integer linear programming (MILP) attack**

The attacker similarly aims to find X' and Y' that recovers the original M . In this attack, similar to the approach used in the attacks discussed in Sections 3.2 and 3.3, the attacker transforms the disjunctive constraints in the formula for each m_k into a set of linear constraints by introducing Boolean variables z_{kij} . This reformulation allows the problem to be solved as a mixed-integer linear program.

More precisely, since each coefficient m_k of M satisfies

$$m_k = \max_{(i,j): i+j=k} (x_i + y_j),$$

it can be equivalently expressed through the following subsystem of inequalities:

$$x_i + y_j \leq m_k, \quad \forall i, j,$$

$$x_i + y_j + (1 - z_{kij})T \geq m_k, \quad \forall i, j,$$

$$\sum_{i,j} z_{kij} = 1, \quad z_{kij} \in \{0, 1\}, \quad \forall i, j.$$

Here, T is a sufficiently large constant. This approach can be used to propose the following attack.

Figure 11 shows the performance of this attack with $t = 3$, where it achieves a success rate comparable to the previous attack but with significantly greater efficiency, even for the recommended protocol parameters ($d = 150$).

In practical terms, this success rate means that the attacker can successfully factor the public key in approximately half of all randomly generated instances. Consequently, if the protocol were deployed, the attacker could potentially impersonate half of the users and sign messages using their signatures. Recall that this success rate is explained by the existence of alternative factors X' and Y' different from the original pair, which still satisfy the verification process and can be efficiently found via the MILP formulation.

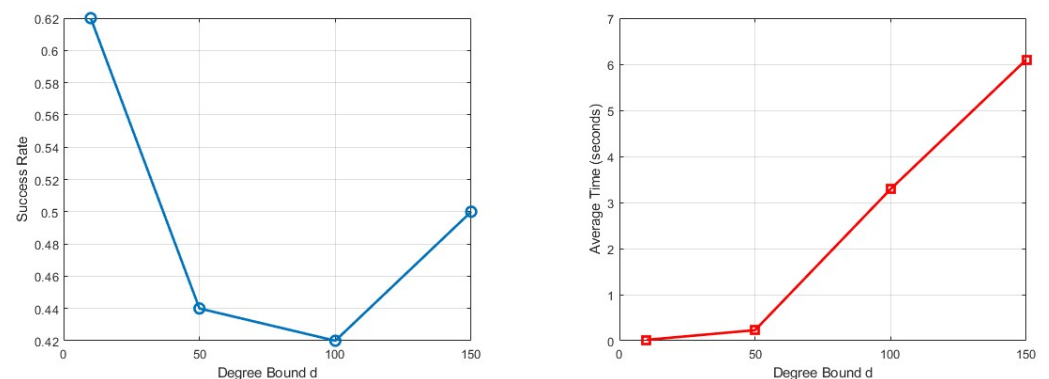


Figure 11. Success rate and computational time of Algorithm 10.

Algorithm 10 MILP-based attack on Protocol 2

Input: Public key coefficients m_k , signature parameters t, r , degree bound $2d$, big constant T .

Output: Recovered factors X', Y' .

- 1: **for** $d_x = 1$ **to** t **do**
- 2: Set $d_y = 2d - d_x$.
- 3: Solve the following system for all $k \in \{0, 1, \dots, 2d\}$, and for all $i \in \{0, 1, \dots, d_x\}$ and $j \in \{0, 1, \dots, d_y\}$ such that $i + j = k$ using a MILP solver.

$$\begin{cases} x_i + y_j \leq m_k, \\ x_i + y_j + (1 - z_{kij})T \geq m_k, \\ \sum_{(i,j): i+j=k} z_{kij} = 1, \quad z_{kij} \in \{0, 1\}. \end{cases}$$

- 4: Construct the polynomials X' and Y' using the derived x_i and y_j , respectively.
- 5: **return** (X', Y') .

6. Conclusions

In this paper, we proposed three new attacks against the tropical and max–min implementations of Stickel protocol. Our aim was to avoid the problem of minimal covers enumeration and the associated worst case exponential complexity encountered in the Kotov–Ushakov attacks. While we previously proposed an attack against these protocols [5,14] that avoided enumerating all minimal solutions by carefully selecting a single minimal solution, this method, although very successful for the tropical case, occasionally fails. Consequently, it is plausible that Alice and Bob could design the protocol’s public matrices to resist this attack, and this method still shows increasing complexity with the polynomial degree used, though not exponentially. Thus, the goal of the techniques implemented in Algorithms 3–8 was to achieve a success rate above 95% with the lowest possible execution time and reduced dependence on the polynomial degree, which is commonly the variable parameter controlled by Alice and Bob.

The first proposed attack (Algorithms 3 and 4) aims to find a solution x that minimizes an objective function of the shape $\sum_i ((A \otimes x)_i - b_i)^2$ instead of finding all minimal solutions of a system $A \otimes x = b$ as in the typical Kotov–Ushakov attack. This attack employs the simulated annealing algorithm, a global optimization technique, to find such solution. It achieved a perfect success rate 100% against the tropical Stickel protocol and a high success rate (above 90%) against the max–min Stickel protocol, both with very fast execution times. Additionally, the execution time showed only a minor increase as the polynomial degree increased. However, unlike the Kotov–Ushakov attack, this approach is sensitive to the size of public matrix entries and polynomial coefficients used in the protocol. While it remains usually effective even for large values, we are more likely to encounter some trials that take significantly longer than average to solve. Also, we cannot definitely say that simulated annealing outperforms other attacks in the max–min case since it is not achieving a perfect success rate in our experiments (or rather, we have to “sacrifice” the success rate in order for the attack to be complete within a reasonable timeframe).

The second proposed attack (Algorithms 5 and 6) aims to solve the system $A \otimes x = b$ by transforming it into a mixed-integer linear system and then solving it using MILP solver. Unfortunately, this attack demonstrated slower execution times compared to the typical Kotov–Ushakov attack, and it remains heavily dependent on the polynomial degree used in the targeted protocols. Consequently, similar to the typical Kotov–Ushakov attack, Alice and Bob can resist this attack by increasing the polynomial degree.

The third proposed attack (Algorithms 7 and 8), which we call Shpilrain’s attack, aims to solve equations (11) by formulating them as a mixed-integer linear program. Interestingly, this attack is completely independent from the used polynomial degree in the protocol, which makes it effective even if Alice and Bob use very high polynomial degrees. The attack has also demonstrated remarkably fast execution times, taking roughly 21 s for the tropical case with dimension 10 and polynomial degree 50. A significant limitation of this attack is its high memory requirement due to the need of encoding a large number of equations, namely on the order of n^4 . Consequently, Alice and Bob could potentially defend against it by employing large matrix dimensions. However, it is worth noting that the typical Kotov–Ushakov attack would likely encounter similar challenges in such scenarios, specifically those related to the high number of minimal covers.

Let us also observe that Shpilrain’s attack also applies to the modifications of Stickel protocol based on Jones matrices and Linde-de la Puente matrices suggested in [15]. Namely, the protocol based on Jones matrices is only replacing the tropical polynomials of A and B with tropical quasi-polynomials of the same matrices, so we can still find X and Y directly from (11) (and its MILP reformulation). As for the Linde-de la Puente matrices, equations $X \otimes A = A \otimes X$ and $Y \otimes B = B \otimes Y$ have to be replaced with linear inequalities and equations that define Linde-de la Puente matrices. We are not including the numerical results here but the situation is similar to what is reported in Figure 7.

Finally, it is notable that the findings presented in this paper likely indicate that the max-min and hence also “digital” implementations of the Stickel protocol overall tend to be more resistant to the attacks described in this paper and [5] than the tropical implementation. This conclusion arises because two of the three proposed attacks in this paper, alongside the single cover heuristic [14], demonstrate much greater effectiveness against the tropical case. Furthermore, the typical Kotov–Ushakov attack is more efficient against the tropical Stickel protocol compared to its analogue against the max–min Stickel protocol. Better implementation of Shpilrain’s attack and alternative ideas which would allow for solving Problems 1 and 2 with higher dimensional matrices are still to be considered. Also, the reasons behind the relatively good performance of simulated annealing in the tropical case and “satisfactory” performance in the max–min case are not clear to us and can be a topic of further research, as well as the conditions under which the simulated annealing based attacks are guaranteed to solve a problem within a reasonable timeframe.

Author Contributions: Conceptualization, S.A. and S.S.; methodology, S.A. and S.S.; software, S.A.; validation, S.A. and S.S.; formal analysis, S.A. and S.S.; investigation, S.A. and S.S.; writing — original draft preparation, S.A. and S.S.; writing — review and editing, S.A. and S.S.; visualization, S.A.; supervision, S.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data is contained within the article.

Acknowledgments: We are grateful to our anonymous referees for many interesting questions which they raised and their comments, which helped us improve the quality of presentation.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Grigoriev, D.; Shpilrain, V. Tropical cryptography. *Commun. Algebra* **2013**, *42*, 2624–2632. [[CrossRef](#)]
2. Shpilrain, V. Cryptanalysis of Stickel’s key exchange scheme. In *Computer Science—Theory and Applications*; Hirsch, E.A., Razborov, A.A., Semenov, A., Slissenko, S., Eds.; Springer: Berlin/Heidelberg, Germany, 2008; LNTCS; Volume 5010, pp. 283–288.
3. Stickel, E. A new method for exchanging secret keys. In *Proceedings of the Third International Conference on Information Technology and Applications (ICITA’05)*, Washington, DC, USA, 4–7 July 2005; Volume 2, pp. 426–430.
4. Klir, G.J.; Yuan, B. *Fuzzy Sets and Fuzzy Logic. Theory and Applications*; Prentice Hall: Englewood Cliffs, NJ, USA, 1995.

5. Alhussaini, S.; Sergeev, S. On implementation of Stickel's key exchange protocol over max-min and max-T semirings. *J. Math. Cryptol.* **2024**, *18*, 20240014. [CrossRef]
6. Durcheva, M. Cryptography based on (idempotent) semirings: Abandoning tropicality? *Encyclopedia* **2025**, *5*, 26. [CrossRef]
7. Kotov, M.; Ushakov, A. Analysis of a key exchange protocol based on tropical matrix algebra. *J. Math. Cryptol.* **2018**, *12*, 137–141. [CrossRef]
8. Huang, H.; Jiang, X.; Peng, C.; Pan, G. A new semiring and its cryptographic applications. *Aims Math.* **2024**, *9*, 20677–20691. [CrossRef]
9. Chen, J.; Grigoriev, D.; Shpilrain, V. Tropical cryptography III: Digital signatures. *J. Math. Cryptol.* **2024**, *18*, 20240005. [CrossRef]
10. Golan, J.S. *Semirings and their Applications*; Springer: Berlin/Heidelberg, Germany, 2000.
11. Sánchez, Á.O.; Portela, D.C.; López-Ramos, J.A. On the solutions of linear systems over additively idempotent semirings. *Mathematics* **2024**, *12*, 2904. [CrossRef]
12. Alhussaini, S.; Sergeev, S. On the security of the initial tropical Stickel protocol and its modification based on Linde-de la Puente matrices. In *Applicable Algebra in Engineering, Communication and Computing*; Springer Nature: Berlin/Heidelberg, Germany, 2025. [CrossRef]
13. Huang, H.; Li, C. Tropical cryptography based on multiple exponentiation problem of matrices. *Secur. Commun. Netw.* **2022**, *2022*, 1024161. [CrossRef]
14. Alhussaini, S.; Collett, C.; Sergeev, S. Generalized Kotov-Ushakov attack on tropical Stickel protocol based on modified tropical circulant matrices. *Kybernetika* **2024**, *60*, 603–623. [CrossRef]
15. Muanalifah, A.; Sergeev, S. Modifying the tropical version of Stickel's key exchange protocol. *Appl. Math.* **2020**, *65*, 727–753. [CrossRef]
16. Gavalec, M. Solvability and unique solvability of max–min fuzzy equations. *Fuzzy Sets Syst.* **2001**, *124*, 385–393. [CrossRef]
17. Zahariev, Z. Solving Max-Min Fuzzy Linear Systems of Equations. Algorithm and Software. *Annual of "Informatics" section. Union of Scientists in Bulgaria*, 6:1–16. 2013. Available online: http://e-university.tu-sofia.bg/e-publ/files/12485_SUB-Informatics-2013-6-001-016.pdf (accessed on 12 September 2025).
18. Peeva, K.; Kyosev, Y. Fuzzy Relational Calculus—Theory, Applications and Software (with CD-ROM). In *Advances in Fuzzy Systems—Applications and Theory*; World Scientific Publishing Company: Singapore, 2004; Volume 22.
19. Elbassioni, K.M. A note on systems with max–min and max-product constraints. *Fuzzy Sets Syst.* **2008**, *159*, 2272–2277. [CrossRef]
20. Michalewicz, Z.; Fogel, D. *How to Solve It: Modern Heuristics*; Springer: Berlin/Heidelberg, Germany, 2000.
21. Tsuzuki, M.d.G.; Martins, T.d.C. *Simulated Annealing: Strategies, Potential Uses and Advantages*; Mathematics Research Developments Series; Nova Science Publishers, Incorporated: Hauppauge, NY, USA, 2014.
22. Schutter, B.D.; Heemels, W.P.M.H.; Bemporad, A. On the equivalence of linear complementarity problems. *Oper. Res. Lett.* **2002**, *30*, 211–222. [CrossRef]
23. Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual*; Gurobi Optimization, LLC: Beaverton, OR, USA, 2023.
24. Ponmaheshkumar, A.; Kotov, M.; Perumal, R. Cryptanalysis of a key exchange protocol based on a digital semiring. In *Communications in Algebra*; Taylor and Francis: Abingdon-on-Thames, UK, 2025. [CrossRef]
25. Panny, L. Forging tropical signatures. In *Applied Cryptography and Network Security Workshops*; Andreoni, M., Ed.; Springer Nature: Cham, Switzerland, 2024; pp. 3–7.
26. Brown, D.R.L.; Monico, C. More forging (and patching) of tropical signatures. *Cryptology ePrint Archive*; Paper 2023/1837, 2023. Available online: <https://eprint.iacr.org/2023/1837> (accessed on 12 September 2025).
27. Kim, K.H.; Roush, F.W. Factorization of polynomials in one variable over the tropical semiring. *arXiv* **2005**, arXiv:math/0501167. [CrossRef]
28. Butkovič, P. *Max-Linear Systems: Theory and Algorithms*; Springer: London, UK, 2010.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.