International Transactions in Mathematical Sciences and Computers
Peer Reviewed and Refereed Journal
July-Dec 2025, Volume 18 No. 2 pp.223-242
ISSN-(Print) 0974-5068, (Online) 0975-3753

Doi: https://doi.org/10.58517/ITMSC.2025.18209

Cryptanalysis of a Tropical Key Exchange Scheme

*K Ahmed [1], S Pal [2], R Mohan [1].

[1] St. Stephen's College, University of Delhi, Delhi, India

[2] DRDO, Delhi, India

Email: kashif.ahmed@ststephens.edu

Abstract: We present a comprehensive cryptanalysis of a tropical key exchange primitive based on the semidirect product of tropical matrices. While the scheme has previously been subjected to three successful attacks, our study introduces a more efficient and robust approach that significantly improves the computational efficiency and reliability compared to the earlier methods that help achieve an almost 100% success rate. This work also highlights significant vulnerabilities in using tropical matrices as tools for secure exchanges.

Keywords: Cryptanalysis, Tropical key exchange scheme

1. Introduction

Semirings [3, 4] are algebraic structures satisfying all the properties of a ring except for the existence of additive inverse. They may also be seen as a common generalisation of rings and lattices. The set $\boldsymbol{S} = \mathbb{W} \cup \{\infty\}$ equipped with two binary compositions $\oplus$ and $\odot$ defined as

$$a \oplus b = \min(a, b) \text{ and } a \odot b = a + b,$$

*Corresponding Author

is called a tropical semiring [6, 12, 14] or min-plus semiring with additive identity ∞ and multiplicative identity 0. Extending this notion to the set of all matrices of order n results in a tropical semiring of matrices $M_n(S)$ with operations defined as

$$[\boldsymbol{a_{ij}}]\oplus[\boldsymbol{b_{ij}}] = [\boldsymbol{a_{ij}} + \boldsymbol{b_{ij}}],$$

$$[\boldsymbol{a_{ij}}]\odot[\boldsymbol{b_{ij}}] = [\boldsymbol{a_{i1}} \odot \boldsymbol{b_{1j}} \oplus \boldsymbol{a_{i2}} \odot \boldsymbol{b_{2j}} \oplus \ \ldots \oplus \boldsymbol{a_{in}} \odot \boldsymbol{b_{nj}}]$$

and the additive and multiplicative identity elements being

$$O = \begin{pmatrix} \infty & \infty & \cdots & \infty \\ \infty & \infty & \cdots & \infty \\ \vdots & \vdots & \vdots & \vdots \\ \infty & \infty & \cdots & \infty \end{pmatrix} \text{ and } I = \begin{pmatrix} 0 & \infty & \cdots & \infty \\ \infty & 0 & \cdots & \infty \\ \vdots & \vdots & \vdots & \vdots \\ \infty & \infty & \cdots & 0 \end{pmatrix}.$$

Note that in the definition of a tropical semiring, either of $\mathbb{Z}$ or $\mathbb{R}$ can be taken as the underlying set instead of $\mathbb{W}$.

In 2014, Grigoriev and Shpilrain [7] proposed a key exchange protocol based on a tropical semiring. The main idea behind choosing this platform was to prevent linear algebra attacks as solving a system of linear equations can be infeasible in tropical matrix algebra. Their scheme was successfully attacked by Kotov and Ushakov [9] in 2018 by exploiting the patterns displayed by the higher powers of tropical matrices. Grigoriev and Shpilrain [8], in 2019, came up with another set of tropical key exchange schemes. This time they made use of semidirect products in a tropical semigroup with an aim to destroy patterns that higher powers of matrices demonstrated in their previous approach. However, there have been three successful attacks on the protocol -

(1) Rudy and Monico [13] employed a binary search attack on the scheme.

(2) Isaac and Kahrobaei [5] exploited the linear periodicity property of powers of tropical matrices.

(3) Muanalifah and Sergeev [10] presented an attack based upon the solution of the tropical discrete logarithm problem.

Our attack differs from those used in the three techniques above, as we utilize only one of the public matrices and its higher powers instead of computing the direct products proposed in the key exchange scheme. We also demonstrate that our approach is significantly more efficient and guaranteed to work for the prescribed set of parameters.

In the next section, we present the tropical key exchange protocol based on semidirect products, along with its cryptanalysis and a couple of examples. This is followed by a discussion of the advantages of our approach over existing attacks.

1. The Protocol

Grigoriev and Shpilrain proposed two key exchange schemes in [8]; however, it has already been shown in [5] that the second protocol is invalid, as the proposed operation is not associative. Therefore, we propose an attack only on the first primitive. We now provide a brief overview of this key exchange scheme and its underlying structure.

a. Structure Employed

Let $S = (M_k(\mathbb{Z}), \oplus, \odot)$ be a tropical semiring of square matrices of order k. For $A, B \in S$, the adjoint multiplication $\circ$ is defined as

$$A \circ B = A \oplus B \oplus (A \odot B)$$

This operation is both associative and distributive over S and hence, the set $S \times S$ forms a semigroup under the composition $*$ defined as

$$(A, B) * (C, D) = ((A \circ D) \oplus C, B \circ D) = ((A \oplus D \oplus (A \odot D) \oplus C, B \circ D).$$

This structure $(S \times S, *)$ serves as the platform for the proposed key exchange scheme. For brevity, we denote

$H \circ H \circ \ldots \circ H$ (p times) by $H^{\circ p}$ and $(M, H) * (M, H) * \ldots * (M, H)$ (p times) by $(M, H)^p$.

b. Steps involved in the Scheme

(1) Alice and Bob agree on an element $(\boldsymbol{M}, \boldsymbol{H}) \in \boldsymbol{S} \times \boldsymbol{S}$ where $\boldsymbol{M}, \boldsymbol{H} \in \boldsymbol{S}$ are public matrices and choose their private natural numbers $\boldsymbol{m}$ and $\boldsymbol{n}$ respectively.

(2) Alice computes $(\boldsymbol{M}, \boldsymbol{H})^{\boldsymbol{m}} = (\boldsymbol{A}, \boldsymbol{H}^{\circ \boldsymbol{m}})$ using the "square and multiply" method and sends A to Bob keeping $\boldsymbol{H}^{\circ \boldsymbol{m}}$ private.

(3) Bob calculates $(\boldsymbol{M},\boldsymbol{H})^{\boldsymbol{n}} = (\boldsymbol{B},\boldsymbol{H}^{\circ\boldsymbol{n}})$ and transmits $\boldsymbol{B}$ to Alice keeping $\boldsymbol{H}^{\circ\boldsymbol{n}}$private.

(4) Alice computes her private key

$$K_{Alice} = (B \circ H^{\circ m}) \oplus A = B \oplus H^{\circ m} \oplus (B \odot H^{\circ m}) \oplus A.$$

(5) Similarly, Bob's private key is

$$K_{Bob} = (A \circ H^{\circ n}) \oplus B = A \oplus H^{\circ n} \oplus (A \odot H^{\circ n}) \oplus B.$$

Alice's key is the first component of

$$(M,H)^n * (M,H)^m = (B,H^{\circ n}) * (A,H^{\circ m}) = (B \circ H^{\circ m} \oplus A, H^{\circ n} \circ H^{\circ m}),$$

and Bob's key is the first component of

$$(M,H)^m * (M,H)^n = (A,H^{\circ m}) * (B,H^{\circ n}) = (A \circ H^{\circ n} \oplus B, H^{\circ m} \circ H^{\circ n}).$$

Since the left hand sides of both equalities are $(\boldsymbol{M},\boldsymbol{H})^{\boldsymbol{m+n}}$, Alice and Bob obtain the same shared key. The suggested parameters for the protocol are as follows.

- The order k of matrices $\boldsymbol{M},\boldsymbol{H}$ is 30 with their components being integers lying between -1000 and 1000.

- The private numbers $\boldsymbol{m},\boldsymbol{n}$ are of order upto 2^{200}.

2. Cryptanalysis of the Protocol

Baccelli *et al.* in [2] showed that the sequence of powers of tropical matrices is *almost linear periodic*. i.e. after some finite number of terms in the sequence, the elements of the higher powers of matrices start displaying patterns such that each element can be written as the sum of some previous corresponding element and a constant. Nachtigall *et al.* [11] gave a more formal definition of the same.

Definition 3.1. [11] A sequence of matrices $\{\boldsymbol{H^p}, \boldsymbol{p} \in \boldsymbol{N}\}$ is almost linear periodic if there exists a period $\boldsymbol{\rho}$, a linear factor $\boldsymbol{\xi}$, and some defect $\boldsymbol{d}$, such that for all $\boldsymbol{p} > \boldsymbol{d}$ and all indices $\boldsymbol{i}, \boldsymbol{j}$ the following equation holds:

$$\boldsymbol{h_{ij}^{p+\rho}} = \boldsymbol{\xi} + \boldsymbol{h_{ij}^{p}}.$$

Our attack exploits the linear periodicity property of the public tropical matrix $\boldsymbol{H}$. We will now prove a result that simplifies the relationship between A (Alice's transmission) and the public matrices $\boldsymbol{M}, \boldsymbol{H}$. From now on, $\forall\ a \in \mathbb{N},\ H \odot H \odot \ldots \odot H$ (a times) i.e. $H^{\odot a}$ will be denoted by $\boldsymbol{H^a}$ and $\boldsymbol{A}$ represents the first component of $(\boldsymbol{M}, \boldsymbol{H})^{\boldsymbol{a}}$.

Theorem 3.2. *If* $(\boldsymbol{M}, \boldsymbol{H})^{\boldsymbol{a}} = (\boldsymbol{A}, \boldsymbol{H}^{\circ \boldsymbol{a}})$ for any natural number $\boldsymbol{a} > \boldsymbol{1}$, then

$$A = M \oplus ((I \oplus M) \odot (H \oplus H^2 \oplus \ldots \oplus H^{a-1}))$$

where I is the identity matrix of S with respect to $\odot$.

Proof. . The result can be easily proved with the help of the principle of mathematical induction. For $a = 2$,

$$\begin{aligned} (M, H)^2 &= (M, H) * (M, H) \\ &= (M \oplus H \oplus (M \odot H) \oplus M, H \circ H). \\ \implies \quad A &= M \oplus H \oplus (M \odot H) = M \oplus ((I \oplus M) \odot H) \end{aligned}$$

Thus the result is true for $a = 2$. Suppose it holds for $a - 1$. We now show that it holds for a as well.

$$
\begin{aligned}
(M,H)^a &= (M,H)^{a-1} * (M,H) \\
&= (M \oplus ((I \oplus M) \odot (H \oplus H^2 \oplus \ldots \oplus H^{a-2})), H^{\circ(a-1)}) * (M,H) \\
\implies \quad A &= (M \oplus ((I \oplus M) \odot (H \oplus H^2 \oplus \ldots \oplus H^{a-2}))) \circ H \oplus M \\
&= M \oplus ((I \oplus M) \odot (H \oplus H^2 \oplus \ldots \oplus H^{a-2})) \oplus H \\
&\qquad \oplus (M \odot H) \oplus ((I \oplus M) \odot (H^2 \oplus H^3 \oplus \ldots \oplus H^{a-1})) \oplus M \\
&= M \oplus H \oplus (M \odot H) \oplus ((I \oplus M) \odot (H \oplus H^2 \oplus \ldots \oplus H^{a-1})) \\
&= M \oplus (I \oplus M) \odot H \oplus ((I \oplus M) \odot (H \oplus H^2 \oplus \ldots \oplus H^{a-1})) \\
&= M \oplus ((I \oplus M) \odot (H \oplus H^2 \oplus \ldots \oplus H^{a-1})).
\end{aligned}
$$

Hence the result holds for all natural numbers $a > 1$.

We now present a couple of observations made while computing the higher powers of tropical matrices using programs written in C++.

Table 1: *Experiments performed on 100,000 randomly generated tropical matrices of different orders to find out the number of instances when* $H \oplus H^2 \oplus \ldots \oplus H^{1000} = H^{1000}$

Order of H	Percentage of cases when the equality holds
2	74.27%
3	88.61%
4	93.68%
5	96.79%
8	99.71%
10	99.95%
12	99.98%
15	99.99%
30	100%

**The program was coded in C++ language. Refer to Appendix A.*

Observation 1. If H is a tropical square matrix containing randomly chosen integers between -1000 and 1000, the elements of its higher powers tend to become large negative values and decrease linearly with each successive power. We conducted experiments on 100,000 randomly chosen tropical matrices of different orders with elements ranging between -1000 and 1000, calculating their powers upto 1000 (see Appendix A). As shown in Table 1, as the order of matrices increases, the probability of $H \oplus H^2 \oplus \ldots \oplus H^{1000}$ being equal to H^{1000} rises significantly, approaching nearly 100% for matrices of order 30. Since 'a' is sufficiently large, we exploit this property of tropical matrices in our attack by assuming

$$H \oplus H^2 \oplus \ldots \oplus H^{a-1} = H^{a-1}. \quad (1)$$

It is important to note that this equation does not hold universally, as there are special cases where the two sides may be unequal. However, such cases are highly unlikely for matrices of order ≥ 20. We will discuss one such scenario later.

Using the above relationship and Theorem (3.2), one can rewrite the message transmitted by Alice to Bob as

$$A = M \oplus (I \oplus M) \odot H^{m-1} \quad (2)$$

The attack we employ is similar to the one we proposed in [1]. Using this strategy, one can compute Alice's private number m, which then paves way for deriving the shared secret key. Before proceeding with our attack, let us first illustrate, through an example, how higher powers of tropical matrices can be represented as equations involving only a single parameter. Consider a 5×5 tropical matrix A and its tropical powers where the integral entries are randomly chosen from -100 to 100.

$$A = \begin{pmatrix} -48 & -54 & -77 & 82 & 99 \\ 92 & -13 & -66 & -24 & 62 \\ 17 & -66 & 75 & 93 & -95 \\ 11 & -62 & -78 & -40 & 84 \\ -83 & -29 & -17 & 94 & -67 \end{pmatrix}$$

$$A^2 = \begin{pmatrix} -96 & -143 & -125 & -78 & -172 \\ -49 & -132 & -102 & -64 & -161 \\ -178 & -124 & -132 & -90 & -162 \\ -61 & -144 & -128 & -86 & -173 \\ -150 & -137 & -160 & -53 & -134 \end{pmatrix}$$

$$A^3 = \begin{pmatrix} -255 & -201 & -209 & -167 & -239 \\ -244 & -190 & -198 & -156 & -228 \\ -245 & -232 & -255 & -148 & -229 \\ -256 & -202 & -210 & -168 & -240 \\ -217 & -226 & -227 & -161 & -255 \end{pmatrix}$$

$$A^4 = \begin{pmatrix} -322 & -309 & -332 & -225 & -306 \\ -311 & -298 & -321 & -214 & -295 \\ -312 & -321 & -322 & -256 & -350 \\ -323 & -310 & -333 & -226 & -307 \\ -338 & -293 & -294 & -250 & -322 \end{pmatrix}$$

$$A^5 = \begin{pmatrix} -389 & -398 & -399 & -333 & -427 \\ -378 & -387 & -388 & -322 & -416 \\ -433 & -388 & -389 & -345 & -417 \\ -390 & -399 & -400 & -334 & -428 \\ -405 & -392 & -415 & -317 & -389 \end{pmatrix}$$

$$A^6 = \begin{pmatrix} -510 & -465 & -466 & -422 & -494 \\ -499 & -454 & -455 & -411 & -483 \\ -500 & -487 & -510 & -412 & -484 \\ -511 & -466 & -467 & -423 & -495 \\ -472 & -481 & -482 & -416 & -510 \end{pmatrix}$$

$$A^7 = \begin{pmatrix} -577 & -564 & -587 & -489 & -561 \\ -566 & -553 & -576 & -478 & -550 \\ -567 & -576 & -577 & -511 & -605 \\ -578 & -565 & -588 & -490 & -562 \\ -593 & -548 & -549 & -505 & -577 \end{pmatrix}$$

$$A^8 = \begin{pmatrix} -644 & -653 & -654 & -588 & -682 \\ -633 & -642 & -643 & -577 & -671 \\ -688 & -643 & -644 & -600 & -672 \\ -645 & -654 & -655 & -589 & -683 \\ -660 & -647 & -670 & -572 & -644 \end{pmatrix}$$

Note that from $\boldsymbol{A^6}$ onwards, each element of these successive powers differs by a linear factor from the previous corresponding element and this pattern repeats after every third power. Here the set of linear factors is $\{\boldsymbol{-67, -89, -99, -121}\}$ with every $\boldsymbol{(i,j)^{th}}$ element differing from its predecessor by one of the four factors mentioned above. Hence, the period $\boldsymbol{\rho = 3}$ and defect $\boldsymbol{d = 5}$.

We write $t_1 = \left\lfloor \frac{t-d}{\rho} \right\rfloor = \left\lfloor \frac{t-5}{3} \right\rfloor$ where $\lfloor . \rfloor$ denotes the greatest integer function. Let $\left(a_{ij}\right)^t$ denote the $(i,j)^{th}$ th element of the t^{th} tropical power of the matrix A. Then, for $t > 8$,

$$(a_{11})^t = \begin{cases} -389 - 121t_1 - 67t_1 - 67t_1, & if\ (t-5)\ mod\ 3 = 0 \\ -389 - 121t_1 - 67t_1 - 67t_1 - 121, & if\ (t-5)\ mod\ 3 = 1 \\ -389 - 121t_1 - 67t_1 - 67t_1 - 121 - 67, & if\ (t-5)\ mod\ 3 = 2 \end{cases}$$

i.e.

$$(a_{11})^t = \begin{cases} -389 - 255t_1, & if\ (t-5)\ mod\ 3 = 0 \\ -510 - 255t_1, & if\ (t-5)\ mod\ 3 = 1 \\ -577 - 255t_1, & if\ (t-5)\ mod\ 3 = 2 \end{cases}$$

Table 2: *Experiments performed on 100,000 randomly generated tropical matrices of order 30 with elements lying between -1000 to 1000 that gives their respective defects and period.*

Average Defect	Median Defect	Max. Defect	Avg. Period	Median Period	Max. Period
37.37	25	2765	2.86	2	35

**The program was coded in C++ language. Refer to Appendix A.*

Observation 2. Our attack relies on the fact that the defect and period of a tropical matrix of order 30 with entries between -1000 to 1000 can be easily calculated. We performed experiments on 100,000 matrices with same parameters as prescribed in [8] (see Appendix A). The average and median defect turns out to be 37.37 and 25 whereas the average and median period are 2.86 and 2 respectively (Table 2).

a. Steps involved in the attack

We now formally present an attack on the protocol.

(1) Calculate the defect d and period ρ of the public matrix $H = (h_{ij})$ by computing its powers and finding the difference between successive powers. We stop when the matrices of these differences start repeating (a corresponding C++ program for the same is given in Appendix A, Code 2).

(2) For $m > \rho + d$, define $m_1 = \left\lfloor \frac{m-d}{\rho} \right\rfloor$ Let $(dij)_a = (hij)^{a+1} - (hij)^a$. Then the $(i,j)^{th}$ element of H^m is one of the following ρ possibilities:

$$(h_{ij})^m = \begin{cases} (h_{ij})^d + \left((d_{ij})_d + (d_{ij})_{d+1} + \cdots + (d_{ij})_{d+\rho-1}\right).m_1 \\ \qquad when\ (m-d)\ mod\ \rho = 0 \\ (h_{ij})^d + \left((d_{ij})_d + (d_{ij})_{d+1} + \cdots + (d_{ij})_{d+\rho-1}\right).m_1 + (d_{ij})_d \\ \qquad when\ (m-d)\ mod\ \rho = 1 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ (h_{ij})^d + \left((d_{ij})_d + (d_{ij})_{d+1} + \cdots + (d_{ij})_{d+\rho-1}\right).m_1 + (d_{ij})_d + \cdots + (d_{ij})_{d+\rho-2} \\ \qquad when\ (m-d)\ mod\ \rho = \rho - 1. \end{cases}$$

(3) Consider $M \oplus (I \oplus M) \odot H^{m-1} = A$. Note that we need not simplify the entire expression on the left hand side. It is enough to compute any one particular element of $M \oplus (I \oplus M) \odot H^{m-1}$ and then compare it to the corresponding element of A.

(4) As there are ρ number of equations for every component, we solve these equations (for any particular component) one by one and stop as soon as we get an integral solution for m_1 (the remaining equations give non-integral solutions). m_1 gives m which can then be used to find $(B \circ H^{\circ m}) \oplus A$. i.e. Alice's key.

Let us now see the working of our algorithm with the help of a toy example by initially considering a case when $H \oplus H^2 \oplus \ldots \oplus H^a = H^a$. Suppose the chosen public matrices are

$$M = \begin{pmatrix} 35 & -23 & 81 \\ 73 & -71 & 6 \\ -55 & 23 & 12 \end{pmatrix} \text{ and } H = \begin{pmatrix} -73 & -81 & -66 \\ -86 & 44 & -54 \\ 59 & -66 & 73 \end{pmatrix}$$

and the private numbers for Alice and Bob be $m = 43$ and $n = 29$ respectively. Then, the matrix shared by Alice is

$$A = M \oplus ((I \oplus M) \odot (H \oplus H^2 \oplus \ldots \oplus H^{42})) = \begin{pmatrix} -3522 & -3530 & -3515 \\ -3570 & -3578 & -3563 \\ -3562 & -3549 & -3534 \end{pmatrix}$$

and Bob shares

$$B = M \oplus ((I \oplus M) \odot (H \oplus H^2 \oplus \ldots \oplus H^{28})) = \begin{pmatrix} -2353 & -2361 & -2346 \\ -2401 & -2409 & -2394 \\ -2393 & -2380 & -2365 \end{pmatrix}$$

We will use the public parameters M, H and A to find m. To calculate the defect and period of the matrix H, we compute its successive powers and observe that, from fourth power onwards, each component differs from the preceding component by a factor of either -94 or -73 . Thus, $d = 3$ and $\rho = 2$. Note that here

$$I \oplus M = \begin{pmatrix} 0 & -23 & 81 \\ 73 & -71 & 6 \\ -55 & 23 & 0 \end{pmatrix}$$

To calculate m, we need to compute any one element of the expression $M \oplus (I \oplus M) \odot H^{m-1}$ and equate it to the corresponding element of A. Let us compare the first element on both sides. This only requires an expression for the first column of $H^{m'}$ where $m' = m - 1$. For $m' > 5$, we write $m_1 = \left\lfloor \frac{m'-3}{2} \right\rfloor$.

$$H^{m'} = \begin{pmatrix} -240 - 167m_1, & when\ (m'-3)\ mod\ 2 = 0 & \dots\dots\dots\dots\dots & \dots\dots\dots\dots\dots \\ -334 - 167m_1, & when\ (m'-3)\ mod\ 2 = 1 & \dots\dots\dots\dots\dots & \dots\dots\dots\dots\dots \\ -253 - 167m_1, & when\ (m'-3)\ mod\ 2 = 0 & \dots\dots\dots\dots\dots & \dots\dots\dots\dots\dots \\ -326 - 167m_1, & when\ (m'-3)\ mod\ 2 = 1 & \dots\dots\dots\dots\dots & \dots\dots\dots\dots\dots \\ -225 - 167m_1, & when\ (m'-3)\ mod\ 2 = 0 & \dots\dots\dots\dots\dots & \dots\dots\dots\dots\dots \\ -319 - 167m_1, & when\ (m'-3)\ mod\ 2 = 1 & \dots\dots\dots\dots\dots & \dots\dots\dots\dots\dots \end{pmatrix}$$

Equating the first component of both sides of $M \oplus (I \oplus M) \odot H^{m'} = A$ gives

$$\text{either } -276 - 167m_1 = -3522 \text{ or } -349 - 167m_1 = -3522.$$

First equation gives a decimal point solution for $\boldsymbol{m_1}$ and hence it is discarded. The second equation gives $\boldsymbol{m_1 = 19 \Rightarrow m' = 41\ or\ 42}$. Since $\boldsymbol{(m'-3)\ mod\ 2 = 1, m' = 42. i.e. m = 43}$. Using this $\boldsymbol{m}$, the secret key can now be easily calculated.

b. When $H \oplus H^2 \oplus \dots \oplus H^a \neq H^a$

It is worth mentioning that the C++ program we wrote for Table 1 in Appendix A, almost always resulted in random matrices for which $H \oplus H^2 \oplus \dots \oplus H^a = H^a$. However, it is possible that this equality may not hold. In the same C++ program referenced in Table 1, we also computed all cases where atleast one element on each side of Equation 1 coincides. We observed that even for matrices

of small order, where the equality does not hold, most of the elements of $H \oplus H^2 \oplus \ldots \oplus H^a$ still turn out to be equal to the corresponding elements of H^a. In fact, as shown in Table 3, for random tropical matrices of order five and more, with integral entries between -1000 and 1000, atleast one element on each side of the Equation 1 always coincides.

Table 3*: Experiments performed on 100,000 randomly generated tropical matrices of different orders to find out the no. of cases when* $H \oplus \ldots \oplus H^{1000} \neq H^{1000}$ *and the no. of cases when none of the components on the two sides of the eqn. coincide.*

Order of H	% of cases when equality does not hold	% of cases when none of the elements coincide
2	25.76%	13.17%
3	11.39%	11.39%
4	6.32%	0.053%
5	3.21%	0
8	0.29%	0
10	0.05%	0
12	0.02%	0
15	0.01%	0
30	0	0

**The program was coded in C++ language. Refer to Appendix A.*

Let us now see how our attack works in such cases with the help of an example. Suppose the chosen public matrices and private numbers

$$M = \begin{pmatrix} 35 & -23 & 81 \\ 73 & -71 & 6 \\ -55 & 23 & 12 \end{pmatrix}, H = \begin{pmatrix} 5 & -3 & 0 \\ -4 & 5 & 0 \\ 0 & 0 & 0 \end{pmatrix}, m = 43 \text{ and } n = 29.$$

Note that the above parameters are same as those in the previous case except for the matrix H. The matrix shared by Alice is

$$A = M \oplus ((I \oplus M) \odot (H \oplus H^2 \oplus \ldots \oplus H^{42})) = \begin{pmatrix} -167 & -170 & -167 \\ -215 & -218 & -215 \\ -202 & -198 & -198 \end{pmatrix}$$

But, in this case,

$$A = M \oplus (I \oplus M) \odot H^{42} = \begin{pmatrix} -164 & -170 & -167 \\ -212 & -218 & -215 \\ -202 & -195 & -198 \end{pmatrix}$$

which is clearly not equal to $\boldsymbol{A}$. However, it is quite evident that the above matrix differs from $\boldsymbol{A}$ at only a few places (three, to be exact), while the remaining components coincide.

We will now try and attack the protocol. Here, the defect and period for the matrix H are $\boldsymbol{d = 3}$ and $\boldsymbol{\rho = 2}$ respectively. Like in the first example, to compute $\boldsymbol{m}$, we need to compute any one component of the expression $\boldsymbol{M \oplus (I \oplus M) \odot H^{m-1}}$ and equate it to the corresponding element of $\boldsymbol{A}$. However, as we have already seen, not all corresponding elements on the two sides are equal, in particular, the $\boldsymbol{(1,1)^{th}}$, $\boldsymbol{(2,1)^{th}}$ $\boldsymbol{a}$nd $\boldsymbol{(3,2)^{th}}$ elements are unequal.

Let us see what happens if one ends up comparing unequal elements on both sides. Consider the $(2,1)^{th}$ element. For this, one only requires an expression for the first column of $H^{m'}$ where $m' = m - 1$. For $m' > 5$, we write $m_1 = \left\lfloor \frac{m'-3}{2} \right\rfloor$.

$$H^{m'} = \begin{pmatrix} -4 - 7m_1, & when\ (m'-3)\ mod\ 2 = 0 & \dots\dots\dots\dots\dots & \dots\dots\dots\dots\dots \\ -14 - 77m_1, & when\ (m'-3)\ mod\ 2 = 1 & \dots\dots\dots\dots\dots & \dots\dots\dots\dots\dots \\ -11 - 7m_1, & when\ (m'-3)\ mod\ 2 = 0 & \dots\dots\dots\dots\dots & \dots\dots\dots\dots\dots \\ -8 - 7m_1, & when\ (m'-3)\ mod\ 2 = 1 & \dots\dots\dots\dots\dots & \dots\dots\dots\dots\dots \\ -7 - 7m_1, & when\ (m'-3)\ mod\ 2 = 0 & \dots\dots\dots\dots\dots & \dots\dots\dots\dots\dots \\ -11 - 7m_1, & when\ (m'-3)\ mod\ 2 = 1 & \dots\dots\dots\dots\dots & \dots\dots\dots\dots\dots \end{pmatrix}$$

Equating the $(2,1)^{th}$ component of $M \oplus (I \oplus M) \odot H^{m'} = A$ gives,

$$\text{either } -82 - 7m_1 = -212 \text{ or } -85 - 7m_1 = -212.$$

Both equalities give decimal solutions for m_1 which suggests that the two sides are unequal. Therefore, one must now compare another pair of corresponding components. Let us now compare the $(3,1)^{th}$ component. The equations are

$$\text{either } -59 - 7m_1 = -202 \text{ or } -69 - 7m_1 = -202.$$

First equation gives a decimal point solution but the second equation gives $m_1 = 19 \Rightarrow m' = 41\ or\ 42$. Since $(m' - 3)\ mod\ 2 = 1, m' = 42$. i.e. $m = 43$ and we end up with the correct value of m.

Thus, our attack works even when $H \oplus H^2 \oplus \ldots \oplus H^a \neq H^a$, provided that at least one pair of corresponding components on both sides of the equation is the same. When we try to solve the set of equations for an incorrect pair (as we did in the case of the $(2,1)^{th}$ element above), no solution is obtained. One can then consider other pairs until an integral solution is found. As stated earlier, for the experiments we performed, most of the corresponding elements of $H \oplus H^2 \oplus \ldots \oplus H^a$ and H^a were equal, even when the two sides were not identical. We encourage interested readers to further explore the properties of higher powers of such matrices and investigate whether any relationship can be established between the components of $H \oplus H^2 \oplus \ldots \oplus H^a$ and H^a when the two are not identical.

3. Comparison with the other three attacks

We now present some of the advantages of our cryptanalysis over the three other attacks proposed so far.

(1) In our attack, there is no need to compute successive powers of $(\boldsymbol{M}, \boldsymbol{H})$. One only needs to calculate upto $\boldsymbol{d} + \boldsymbol{\rho}$ powers of $\boldsymbol{H}$ (the median value for which turns out to be around 27). This significantly saves time, as the large values of $\boldsymbol{m}, \boldsymbol{n}$ chosen by Alice and Bob hardly affect the time complexity of the algorithm. The median value for the period of the matrix $\boldsymbol{H}$ is about three. Thus, to find the value of $\boldsymbol{m}$, roughly two equations with one variable each need to be solved. Even in extremely rare cases, where the Equation (1) doesn't hold, we were still able to recover Alice's private number $\boldsymbol{m}$.

(2) The binary search attack suggested by Rudy and Monico [13] works for every value of $\boldsymbol{m, n}$ (with $\boldsymbol{m, n}$ being of the order of $\boldsymbol{2^K, K \leq 200}$) but it has the worst time complexity of $O(K^2)$. In other words, the time complexity worsens rapidly with higher values of K. Note that $O(K^2)$ refers to the number of matrix operations. On the other hand, our attack does not involve tropical multiplication of matrices M and H. One only needs to compute the initial higher powers of the matrix $\boldsymbol{H}$ to determine its defect and period, followed by considering any particular element of the Equation (2) and then, in most of the cases, solving around two linear equations in one variable.

(3) Though Isaac and Kahrobaei [5] also focus on finding the defect and period of a matrix but they do so for the sum and product of the matrices $\boldsymbol{M}$ and $\boldsymbol{H}$ that occur as the first component of successive powers of $\boldsymbol{(M, H)}$. Additionally, their assumption is based upon the fact that this sequence of matrices is *almost linearly periodic*. We, on the other hand, exploit the patterns of the higher powers of one of the public matrices $\boldsymbol{H}$ whose *almost linear periodicity* property has already been established in [2].

(4) Muanalifah and Sergeev's attack [10] is based upon the solution of tropical discrete logarithm problem which may require calculating up to $(30 - 1)^2 = 841$ matrix powers (for matrices of order 30). Also, in some cases, their algorithm may not work for the prescribed set of parameters. We have seen that our attack works in all those cases where Equation (1) holds and even otherwise.

4. Conclusion

The key exchange protocol proposed by Grigoriev and Shpilrain is clearly vulnerable to attacks, as has already been shown by different cryptanalyses conducted so far. Although our attack also makes use of the patterns displayed by higher powers of tropical matrices, our approach is distinct and certainly more efficient compared to other proposed attacks. Our approach also has a 100% success rate for the prescribed set of elements. Even in a few isolated cases where Equation (1) fails to hold, we were able to compute the private parameter.

If one carefully observes the various proposed cryptographic protocols that utilize tropical matrices, it is evident that almost all of these primitives leak information about the number of operations performed to obtain the matrix communicated to the other party. The reason is that the components of higher powers of tropical matrices grow large in magnitude with each

operation, and hence, due to this pattern, it becomes straightforward for an adversary to crack the key.

Though, to our knowledge, there has been no cryptographic technique based on tropical algebra that has not been successfully attacked so far, it is still too early to completely discard it. We believe that this contemporary branch of algebra can contribute significantly to cryptographic protocols, particularly when used alongside standard algebraic platforms.

Reference

1. Ahmed, K., Pal, S., Mohan, R.: A review of the tropical approach in cryptography. Cryptologia, 1–25 (2021)
2. Baccelli, F., Cohen, G. A., Olsder, G. A., Quadrat, J. -P.: Synchronization and linearity: An algebra for discrete event systems. Journal of the Operational Research Society 45, 118–119 (1994)
3. Golan, J.: Semirings and their Applications. Springer Science & Business Media, (2013)
4. Gondran, M., Minoux, M.: Graphs, dioids and semirings: new models and algorithms. Springer Science & Business Media, (2008)
5. Isaac, S., Kahrobaei, D.: A closer look at the Tropical Cryptography. International Journal of Computer Mathematics: Computer Systems Theory, 1–6 (2021)
6. Izhakian, Z.: Tropical Arithmetic and Matrix Algebra. Communications in Algebra 37, 1445–1468 (2009)
7. Grigoriev, D., Shpilrain, V.: Tropical Cryptography. Communications in Algebra 42, 2624{2632 (2014)
8. Grigoriev, D., Shpilrain, V.: Tropical Cryptography II. Communications in Algebra 47, 4224{4229 (2019)
9. Kotov, M., Ushakov, A.: Analysis of a Key Exchange Protocol based on Tropical Matrix Algebra. Journal of Mathematical Cryptology 12, 137{141 (2018)
10. Muanalifah, A., Sergeev, S.: On the Tropical Discrete Logarithm Problem and Security of a Protocol based on Tropical Semidirect Product. Communications in Algebra 50(2), 861{879 (2021)
11. Nachtigall, K. et al.: Powers of matrices over an extremal algebra with applications to periodic graphs. Mathematical Methods of Operations Research 46, 87–102 (1997)
12. Pin, J.: Tropical Semirings. In Idempotency, Cambridge Univ. Press, Cambridge, 50–79 (1998)
13. Rudy, D., Monico, C.: Remarks on a Tropical Key Exchange System'. Journal of Mathematical Cryptology 15, 280–283 (2020)
14. Speyer, D., Sturmfels, B.: Tropical Mathematics. Mathematics Magazine 3, 163–173 (2009)

Appendix A. C++ Codes used in the cryptanalysis of the scheme

Code 1: Program to observe i) the number of instances when sum of first 1000 tropical powers of H is same as $H^{\odot 1000}$ and ii) the number of instances when none of the components of the two sides are equal.

```
#include<iostream>
#include<ctime>
#include<cstdlib>
#include<iomanip>
using namespace std;
const int n =30;//order of matrix
const int n1=100000;//number of randomly generated matrices
int sum[n][n],g[n][n],z=0;
int prod(int e[][n],int f[][n], int n,int q);
int main()
{  int c1=0,c2=0,c3=0;
   srand(time(0));
   for(int y=1;y<=n1;y++)
     { int i,j,q=1,ch=1;
       int a[n][n],b[n][n];
       for(i=0;i<n;i++)
         for(j=0;j<n;j++)
           {  a[i][j]=((rand()+time(0))%2000)-1000;
              b[i][j]=a[i][j];
              sum[i][j]=a[i][j];//copy of matrix in sum. this
                 sum finds A+A^2+..A^n
           }
        prod(a,b,n,q);
        while(q!=1000)//finding powers upto this number
          {
            q++;
            prod(a,b,n,q);
          }
        int temp=1;
        for(i=0;i<n;i++)
          {
            for(j=0;j<n;j++)
              { if (sum[i][j]!=g[i][j])
                temp=2;
              }
          }
   if(temp==1)
     {
       c1++;z++;
     }
   else
     {
       c2++;cout<<"Matrix number: "<<z++;
       cout<<"\nLast Matrix is:\n";
       for(i=0;i<n;i++)
         {
           for(j=0;j<n;j++)
             cout<<g[i][j]<<" ";
             cout<<endl;
         }
        cout<<"\nsum of powers is:\n";
```

```
        for(i=0;i<n;i++)
          {
            for(j=0;j<n;j++)
              cout<<sum[i][j]<<" ";
           cout<<endl;
          }
        cout<<"\n\n";
        for(i=0;i<n;i++)
          {
            for(j=0;j<n;j++)
              if(sum[i][j]==g[i][j])
              {  c3++;
                 j=n;i=n;//to exit two for loops immediately
                      }
          }
      }
    }
   cout<<"same = "<<c1<<" ("<<c1/float(n1)*100<<"%), unequal = "
      <<c2<<" ("<<c2/float(n1)*100<<"%)."<<endl;
   cout<<c3<<" ("<<c3/float(n1)*100<<"%) matrices had atleast
      one element common and "<<c2-c3<<" ("<<(c2-c3)/float(n1)
      *100<<"%) of matrices were completely distinct.\n\n";
   return 0;
}
int prod(int e[][n],int f[][n],int n,int q)
{  int i,j;
   int  d[n],k,l,mini;
   for(i=0;i<n;i++)
     {
       for(j=0;j<n;j++)
         {
           for(k=0;k<n;k++)
             {
               d[k]=e[i][k]+f[k][j];
             }

           for(k=0;k<n;k++)
             mini=d[0];
           for(l=1;l<n;l++)
             {
               if (mini>d[l])
                mini=d[l];
             }
           g[i][j]=mini;
         }
     }
   for(i=0;i<n;i++)
     {
       for(j=0;j<n;j++)
         {
           f[i][j]=g[i][j];
         }

     }
  for(i=0;i<n;i++)
    for(j=0;j<n;j++)
      sum[i][j]=min(sum[i][j],f[i][j]);//find sum of powers
}
```

Code 2: Program to find the average period and defect of tropical matrices of order up to 30 with elements lying between -1000 to 1000.

```
#include<iostream>
#include<ctime>
#include<cstdlib>
#include<iomanip>
#include <algorithm> //for sort
using namespace std;
const int n = 30;//order of matrix
const int n1 = 100000; //number of random matrices
int defect[n1],period[n1];
int dif[20000][n][n],g[n][n],c=1,z,largeD=0,largeP=0;
float avgD=0,avgP=0;
int prod(int e[][n], int n,int q);
int main()
{ srand(time(0));
  for(z=1;z<=n1;z++)
    {
      c=1;
      int i,j,q=0;
      int a[n][n];
      for(i=0;i<n;i++)
        for(j=0;j<n;j++)
            {
                a[i][j]=((rand()+time(0))%2000)-1000;
                g[i][j]=a[i][j];
            }
      prod(a,n,q);
       while(q!=n1 && c==1)
         {
           q++;
           prod(a,n,q);
         }
      }
  cout<<"The defects and periods of "<<n1<<" random matrices of
     order "<<n<<" are:\n\n";
  for(int i=0;i<n1;i++)
    cout<<defect[i]<<" ";
  cout<<"\n\n";
    for(int i=0;i<n1;i++)
    cout<<period[i]<<" ";
  cout<<"\n\n\n";
  float sumd=0,sump=0;
  for(int i=0;i<n1;i++)
    {
      sumd=sumd+defect[i];
      sump=sump+period[i];
    }
  sort(defect, defect+n1); //to sort an array
  sort(period,period+n1);
  cout<<"Average Defect: "<<sumd/n1<<", Median Defect: "<<(
     defect[n1/2]+defect[n1/2+1])/2.0<<", Largest Defect: "<<
     defect[n1-1] <<endl<<endl;
  cout<<"Average Period: "<<sump/n1<<", Median Period: "<<(
     period[n1/2]+period[n1/2+1])/2.0<<", Largest Period: "<<
     period[n1-1]<<endl;
  return 0;
```

```
}
 int prod(int e[][n],int n,int q)
{ int i,j,f[n][n];
  for(i=0;i<n;i++)
    {
       for(j=0;j<n;j++)
        {
          f[i][j]=g[i][j];//copy of powers
        }
    }
  int  d[n],k,l,mini;
  for(i=0;i<n;i++)
    {
       for(j=0;j<n;j++)
        {
           for(k=0;k<n;k++)
            {
              d[k]=e[i][k]+f[k][j];
            }

           for(k=0;k<n;k++)
             mini=d[0];
           for(l=1;l<n;l++)
            {
              if(mini>d[l])
                mini=d[l];
            }
           g[i][j]=mini;
        }
    }
  for(i=0;i<n;i++)
    for(j=0;j<n;j++)
      dif[q][i][j]=g[i][j]-f[i][j];
  for(i=0;i<q;i++)
    {
      int t=1;
      for(j=0;j<n;j++)
        {
          for(k=0;k<n;k++)
            {
              if(dif[q][j][k]!=dif[i][j][k])
                {
                  t=0;break;

                }
            }
        }
      if(t==1)
        {
          cout<<"Matrix "<<z<<":"<<" Defect: "<<q+1<<" and
              Period: "<<q-i<<".\n\n";
          c=0;
          period[z-1]=q-i;defect[z-1]=q+1;
        }
    }
}
```