ANALYSIS OF A KEY EXCHANGE PROTOCOL BASED ON TROPICAL MATRIX ALGEBRA

MATVEI KOTOV AND ALEXANDER USHAKOV

Abstract. In this paper we consider a two party key-exchange protocol proposed in [4, Section 2] which uses tropical matrix algebra as a platform. Our analysis shows that the scheme is not secure.
Keywords. Tropical algebra, cryptography, key-exchange, min-plus systems.
2010 Mathematics Subject Classification. 94A60, 68W30.

1. Introduction

In this paper we analyze a key-exchange protocol based on tropical matrix algebra proposed in [4, Section 2]. Ideas similar to [4] were used before in the "classic" case, i.e., for algebras with familiar addition and multiplication. However in classic case these schemes were shown to be vulnerable to various linear algebra attacks. The idea to use an algebra with another addition and multiplication came as an attempt to avoid those attacks, as there are no known algorithms for solving systems of linear equations in tropical sense (it is an active field of research currently).

1.1. **Tropical algebra.** Consider the extended set of real numbers $\mathbb{R}\cup\{\infty\}$ and binary operations $\oplus, \otimes$ defined by:

$$x \oplus y = \min(x, y),$$
$$x \otimes y = x + y.$$

A set $S \subseteq \mathbb{R} \cup \{\infty\}$ closed under $+$, containing 0 and ∞ is called a *tropical semi-ring*. It is straightforward to check that $(S, \oplus, \otimes)$ satisfies all axioms of a commutative ring with unity 0 except for existence of additive inverses. In this paper $S = \mathbb{Z} \cup \{\infty\}$.

The set of all $n \times n$ matrices $M_n(S)$ with entries from S can be equipped with operations $\oplus$ and $\otimes$ as well, as defined below:

$$(a_{ij}) \oplus (b_{ij}) = (a_{ij} \oplus b_{ij})$$
$$(a_{ij}) \otimes (b_{ij}) = (a_{i1} \otimes b_{1j} \oplus \ldots \oplus a_{in} \otimes b_{nj}).$$

The obtained algebra $R = (M_n(S), \oplus, \otimes)$ is called a *tropical matrix algebra*.

For more information on tropical algebras see [1]. For more on non-commutative algebraic structures used in cryptography see [6].

Date: September 2, 2015.
The second author has been partially supported by NSF grant DMS-1318716.

1.2. **The protocol.** Here we describe a two party key-exchange protocol proposed in [4, Section 2]. Let $A, B \in R$ be matrices satisfying $A \otimes B \neq B \otimes A$, called *public base matrices*.

(1) Alice generates random polynomials $p_1(x), p_2(x) \in \mathbb{Z}[x]$ and sends $U = p_1(A) \otimes p_2(B)$ to Bob.
(2) Bob generates random polynomials $q_1(x), q_2(x) \in \mathbb{Z}[x]$ and sends $V = q_1(A) \otimes q_2(B)$ to Alice.
(3) Alice computes $K_A = p_1(A) \otimes V \otimes p_2(B)$.
(4) Bob computes $K_B = q_1(A) \otimes U \otimes q_2(B)$.

It is easy to check that Alice and Bob finally compute the same matrix $K = K_A = K_B$, called the *shared key*.

The following key generation method is suggested in [4, Section 2.5].

- The size of matrices $n = 10$.
- The entries of matrices A and B are integers, selected randomly in $[-10^{10}, 10^{10}]$.
- The degrees of polynomials $p_1(x), p_2(x), q_1(x), q_2(x)$ are selected uniformly randomly in the range $[1, 10]$.
- The coefficients of $p_1(x), p_2(x), q_1(x), q_2(x)$ are selected uniformly randomly in $[-1000, 1000]$.

1.3. **Computational assumption.** For a passive eavesdropper to break the protocol means to be able to compute the value of K based on the values of A, B, U, V. For that it clearly suffices to find a pair of matrices X, Y satisfying the following conditions:

$$\begin{cases} X \otimes A = A \otimes X, \\ Y \otimes B = B \otimes Y, \\ X \otimes Y = U, \end{cases} \tag{1}$$

or to solve a similar system for Bob's public key. Indeed, if X, Y satisfy the conditions above, then the product $X \otimes V \otimes Y$ is equal to K. In the case of matrix algebra over $(\mathbb{Z}, +, \cdot)$ one would reduce the system above to a system of linear equations (as in [8, 7]). The same approach does not seem to work with tropical algebra as explained in [4].

2. Simple heuristic for original key generation method

In this section we argue that one can find a solution of the system (1) by a very simple heuristic when the public information is generated as proposed in [4].

Consider a particular 3×3 matrix A with entries chosen uniformly randomly in the range $[-100, 100]$ and its tropical powers:

$$A = \begin{pmatrix} -42 & 13 & -96 \\ -28 & 16 & 65 \\ -85 & 31 & -75 \end{pmatrix},$$

$$A^{\otimes 2} = \begin{pmatrix} -181 & -65 & -171 \\ -70 & -15 & -124 \\ -160 & -72 & -181 \end{pmatrix},$$

$$A^{\otimes 3} = \begin{pmatrix} -256 & -168 & -277 \\ -209 & -93 & -199 \\ -266 & -150 & -256 \end{pmatrix},$$

$$A^{\otimes 4} = \begin{pmatrix} -362 & -246 & -352 \\ -284 & -196 & -305 \\ -341 & -253 & -362 \end{pmatrix},$$

$$A^{\otimes 5} = \begin{pmatrix} -437 & -349 & -458 \\ -390 & -274 & -380 \\ -447 & -331 & -437 \end{pmatrix}.$$

As we can see the entries in $A^{\otimes i}$ very soon become negative and decrease linearly with i. In a linear combination of $A^{\otimes i}$'s with sufficiently small coefficients x_i, say $-10 \leq x_i \leq 10$, smaller monomials are irrelevant compared with the leading monomial, i.e., for any choice of the coefficients $-10 \leq x_i \leq 10$ we have:

$$\bigoplus_{i=0}^{5} x_i \otimes A^{\otimes i} = x_5 \otimes A^{\otimes 5}. \tag{2}$$

This is precisely the case for key generation proposed in [4]. Half of the entries in A and B have negative value. The entries in the tropical-powers $A^{\otimes i}$ and $B^{\otimes i}$ become all negative very fast (when $i \geq 2$) and decrease faster than the range for coefficients which makes lower monomials in the polynomials p_1, p_2 irrelevant. Based on that observation one can expect that the system (1) has a solution of the form:

$$X = c \otimes A^{\otimes i} \text{ and } Y = B^{\otimes j}$$

for some $i, j \in [1, D]$ and $c \in \mathbb{Z}$, where D is the upper bound for degrees of polynomials in the protocol.

This gives us the following simple heuristic attack. For each $i, j \in [1, D]$ we compute the matrix $T^{ij} = U - A^{\otimes i} \otimes B^{\otimes j}$. If $T^{ij} = (c)_{kl}$ for some constant c, then $X = c \otimes A^{\otimes i}$, $Y = B^{\otimes j}$ is a required solution to (1), and the algorithm successfully stops. If there are not such i and j, then the algorithm fails.

Table 1 shows success rates of the described algorithm for different key-generation strategies. If the keys are generated as originally proposed, then the success rate is 97%. Success rate decreases to 71% if we allow a larger range for coefficients in polynomials p_1, p_2. It becomes negligible if we allow only non-negative entries in A and B. Below we suggest an attack which works well in this case.

Range for coeff. of polynomials	$[-10^3, 10^3]$	$[-10^{10}, 10^{10}]$	$[0, 10^{10}]$
Range for entries of matrices	$[-10^{10}, 10^{10}]$	$[-10^{10}, 10^{10}]$	$[0, 10^{10}]$
Avg. time	0.5 sec	0.6 sec	1.2 sec
Success rate	97%	71%	2%

TABLE 1. Experimental results of the simple heuristical algorithm

3. General attack

In this section we discuss a general attack on the protocol. The success rate of this attack does not depend on parameters of key generation and equals 100%. We find matrices X and Y of the form:

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

with unknown coefficients $x_i, y_j \in \mathbb{Z}$. Clearly for such matrices the first and the second equality in (1) are satisfied. Notice that:

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

Therefore to break the protocol we need to find $x_0, \dots, x_D, y_0, \dots, y_D \in \mathbb{Z}$ such that:

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

Using the definition of $\oplus$ and $\otimes$, we get a system of equations

$$\min_{i,j} \left(x_i + y_j + T^{ij}_{kl} \right) = 0, \text{ for each } k, l \in [1, n], \tag{3}$$

where $T^{ij} = A^{\otimes i} \otimes B^{\otimes j} - U$. Solving the system (3) is the main goal of this section.

Denote by m_{ij} the least entry in the matrix T^{ij} and by P_{ij} the set of entries where the minimum is achieved:

$$m_{ij} = \min_{k,l} T^{ij}_{kl}, \quad P_{ij} = \{(k, l) \mid T^{ij}_{kl} = m_{ij}\}.$$

Notice that any solution x_i, y_j, $0 \le i, j \le D$, of (3) satisfies the following conditions:

- $x_i + y_j \ge -m_{ij}$;
- for every k, l there exist i, j such that $(k, l) \in P_{ij}$ and $x_i + y_j = -m_{ij}$.

Therefore, to solve (3) we need to find a cover $C \subseteq \{P_{00}, P_{01}, \dots, P_{DD}\}$ of the set $[1, n] \times [1, n]$ and values x_i, y_j, $0 \le i, j \le D$, satisfying:

$$\begin{cases} x_i + y_j = -m_{ij} & \text{if } P_{ij} \in C, \\ x_i + y_i \ge -m_{ij} & \text{otherwise.} \end{cases} \tag{4}$$

Range for coef. of polynomials	$[-10^3, 10^3]$	$[-10^{10}, 10^{10}]$	$[0, 10^{10}]$
Range for entries of matrices	$[-10^{10}, 10^{10}]$	$[-10^{10}, 10^{10}]$	$[0, 10^{10}]$
Maximal number of covers	102	296	20736
Avg. number of covers	4.2	16.52	705.5
Median of number of covers	1	1	7.5
Avg. number of tested covers	1	1.03	1.57
Median of number of tested covers	1	1	1
Avg. time	1.9 sec	2.0 sec	2.5 sec
Success rate	100%	100%	100%

TABLE 2. Experimental results of the general attack

Hence, in order to solve (3) we need to enumerate all minimal covers for $[1, n] \times [1, n]$ and then find those that define consistent systems of equalities and inequalities (4).

Recall that finding a minimal set cover problem is one of Karp's 21 problems shown to be NP-complete in 1972. Nevertheless, for all randomly generated instances of the protocol it was relatively easy to enumerate all possible minimal covers.

To enumerate the covers we use a simple recursive procedure and a couple of heuristics. As one can see in Table 2, often the number of covers is small, and rarely it is greater than 10000. In every experiment we were able to enumerate all the minimal covers in a few seconds.

Also, we noticed that often an appropriate cover (a cover defining a consistent system (4)) is smaller than most other covers produced by our algorithm. Furthermore, for covers of the same size an appropriate cover often has smaller value of

$$|\{i \mid \exists j \ P_{ij} \in C\}| \cdot |\{j \mid \exists i \ P_{ij} \in C\}|$$

than others. Therefore, to speed up computations we sort the covers using these criteria. This strategy works very well, as shown in Table 2, on average we have to test only one or two covers to find a solution.

Finally, to determine if a system (4) is consistent for a particular cover C we use the simplex method, see [2].

The described attacks were implemented in GAP [3]. The implementation can be found in [5]. Our tests were run on Intel Core i3 1.50GHz computer with 4GB of RAM, Ubuntu 14.04, GAP 4.7. We generated 100 random instances for every set of parameters presented in the tables.

4. Conclusion

The protocol described in [4, Section 2] is not secure when used with the proposed parameter values. It is not clear how to modify key generation to provide a sufficient level of security. We encourage an interested reader to use our code [5] and perform his/her computational experiments over the tropical algebra.

References

[1] P. Butkovič. *Max-linear Systems: Theory and Algorithms*. Springer Monographs in Mathematics. Springer London, 2010.
[2] T. Cormen, C. Leiserson, R. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, 3 edition, 2009.
[3] The GAP Group. *GAP – Groups, Algorithms, and Programming, Version 4.7.7*, 2015.
[4] D. Grigoriev and V. Shpilrain. Tropical cryptography. *Comm. Algebra*, 43:2624–2632, 2014.
[5] M. Kotov and A. Ushakov. Implementation of attacks on a key excange protocol based on tropical matrix algebra. Available at `https://github.com/mkotov/tropical`.
[6] A. G. Miasnikov, V. Shpilrain, and A. Ushakov. *Non-Commutative Cryptography and Complexity of Group-Theoretic Problems*. Mathematical Surveys and Monographs. AMS, 2011.
[7] C. Mullan. Cryptanalysing variants of Stickel's key agreement scheme. preprint, 2010.
[8] V. Shpilrain. Cryptanalysis of Stickel's key exchange scheme. In *Computer Science in Russia – CSR 2008*, volume 5010 of *Lecture Notes Comp. Sc.*, pages 283–288. Springer, 2008.

Department of Mathematics, Stevens Institute of Technology, Hoboken, NJ, USA
E-mail address: `mkotov,aushakov@stevens.edu`