Analysis of REDOG: the Pad Thai attack

Alex Pellegrini, Marc Vorstermans

Eindhoven University of Technology, the Netherlands
alex.pellegrini@live.com, m.h.l.vorstermans@tue.nl

Abstract. This short paper presents the Pad Thai message recovery attack on REDOG. We analyze the two parts of the ciphertext separately and build several linear systems of equations, one of which is error-free and can be solved for the original message. The running time of our attack suggests that REDOG's parameters fall short of the claimed security.

1 REDOG Specification

This section introduces the specification of REDOG as per [1].
The system parameters are positive integers $(n, k, \ell, q, m, r, \lambda, t_1, t_2)$, with $\ell < n$ and $t_1 + \lambda t_2 \leq r \leq \lfloor (n-k)/2 \rfloor$, as well as a hash function $\mathsf{hash} : \mathbb{F}_{2^m}^{2n-k} \to \mathbb{F}_{2^m}^{\ell}$.

KeyGen:

1. Select $H = (H_1 \mid H_2)$, $H_2 \in \mathsf{GL}_{n-k}(\mathbb{F}_{2^m})$, a parity check matrix of a $[2n-k, n]$ Gabidulin code, with syndrome decoder Φ correcting r errors.
2. Select a complete rank matrix $M \in \mathbb{F}_{2^m}^{\ell \times n}$ and isometry $P \in \mathbb{F}_{2^m}^{n \times n}$ (with respect to the rank metric).
3. Select a λ-dimensional $\mathbb{F}_2$-subspace $\Lambda \subset \mathbb{F}_{2^m}$ containing 1 and select a random circulant matrix $S^{-1} \in \mathsf{GL}_{n-k}(\mathbb{F}_{2^m})$ having entries only in Λ;
4. Compute $F = MP^{-1}H_1^T \left(H_2^T\right)^{-1} S$ and publish the public key $\mathsf{pk} = (M, F)$. Store the secret key $\mathsf{sk} = (P, H, S, \Phi)$.

Encrypt $(\mathbf{m} \in \mathbb{F}_{2^m}^{\ell}, \mathsf{pk})$

1. Generate uniformly random $\mathbf{e} = (\mathbf{e}_1, \mathbf{e}_2) \in \mathbb{F}_{2^m}^{2n-k}$ with $\mathbf{e}_1 \in \mathbb{F}_{2^m}^{n}$ having $\mathsf{wt}_R(\mathbf{e}_1) = t_1$ and $\mathbf{e}_2 \in \mathbb{F}_{2^m}^{n-k}$ having $\mathsf{wt}_R(\mathbf{e}_2) = t_2$;
2. Compute $\mathbf{m} = \mathsf{msg} + \mathsf{hash}(\mathbf{e})$.
3. Compute $\mathbf{c}_1 = \mathbf{m}M + \mathbf{e}_1$ and $\mathbf{c}_2 = \mathbf{m}F + \mathbf{e}_2$ and send $(\mathbf{c}_1, \mathbf{c}_2)$.

Decrypt $((\mathbf{c}_1, \mathbf{c}_2), \mathsf{sk})$

1. Compute $\mathbf{c}' = \mathbf{c}_1 P^{-1} H_1^T - \mathbf{c}_2 S^{-1} H_2^T = \mathbf{e}' H^T$ where the vector $\mathbf{e}' := (\mathbf{e}_1 P^{-1}, -\mathbf{e}_2 S^{-1})$.
2. Decode $\mathbf{c}'$ using Φ to obtain $\mathbf{e}'$, recover $\mathbf{e} = (\mathbf{e}_1, \mathbf{e}_2)$ using P and S.
3. Solve $\mathbf{m}M = \mathbf{c}_1 - \mathbf{e}_1$. Output $\mathsf{msg} = \mathbf{m} - \mathsf{hash}(\mathbf{e})$.

1.1 Suggested parameters

We list the suggested parameters of REDOG for 128,192 and 256 bits of security submitted to KPQC.

Security parameter	$(n, k, \ell, q, m, r, \lambda, t_1, t_2)$
128	$(30, 6, 25, 2, 59, 12, 3, 6, 2)$
192	$(44, 8, 37, 2, 83, 18, 3, 12, 2)$
256	$(58, 10, 49, 2, 109, 24, 3, 15, 3)$

Table 1. Suggested parameters.

2 The Pad Thai attack

In this section, we describe an attack on REDOG which succeeds in recovering the messages corresponding to REDOG's ciphertexts.

2.1 Overview

We break down the description of the Pad Thai attack in two steps. The result is a linear system of equations which can be solved uniquely for $\mathbf{m}$ and subsequently for $\mathbf{e}_1$ and $\mathbf{e}_2$. Finally, we compute $\mathsf{msg} = \mathbf{m} - \mathsf{hash}(\mathbf{e}_1 \mid \mathbf{e}_2)$.

First step. The goal of the first step is to construct a linear system of equations starting from the relation $\mathbf{c}_2 = \mathbf{m}F + \mathbf{e}_2$. The idea is to combine columns of F and the corresponding entries of $\mathbf{c}$ in order to obtain a system of equations $\mathbf{c}_2' = \mathbf{m}F' + \mathbf{e}_2'$ where $\mathbf{e}_2'$ has only t_2 nonzero entries whose positions are known.

Assume that we know the mentioned system. Observe that, for all security levels of REDOG, we have $t_2 = 2$ or $t_2 = 3$, which means that $n - k - t_2$ entries in $\mathbf{c}'$ are error-free. Let $i_1, \ldots, i_{t_2} \in \{1, \ldots, n-k\}$ be such that $e_{2,i_j} \neq 0$ for $j \in \{1, \ldots, t_2\}$. Take $F'' \in \mathbb{F}_{2^m}^{\ell \times (n-k-t_2)}$ as the submatrix of F' consisting of the columns F_i' for $i \neq i_1, \ldots, i_{t_2}$. Similarly, compute $\mathbf{c}_2'' \in \mathbb{F}_{2^m}^{n-k-t_2}$ by taking the entries c_i' where $i \neq i_1, \ldots, i_{t_2}$. Then, the message $\mathbf{m}$ satisfies

$$\mathbf{c}_2'' = \mathbf{m}F'', \tag{1}$$

which is an underdetermined system as we have ℓ unknowns of $\mathbf{m}$ and $n-k-t_2$ equations, where $n-k-t_2 < \ell$ for every security level. In reality, $\ell - n + k = t_2 + 1$ for every security level, which means that we are $t_2 + 1$ equations short.

Second step. In order to uniquely compute $\mathbf{m}$ we need to pad the system in (1) with $t_2 + 1$ extra error-free equations by combining some of the equations from

$\mathbf{c}_1 = \mathbf{m}M + \mathbf{e}_1$. Let $c'_{1,i} = \mathbf{m}M'_i$ for $i = 1, \ldots, t_2+1$ be such error-free equations. We can add these equations to (1) and obtain a new system

$$(\mathbf{c}''_2 \mid c_{1,1} \mid \cdots \mid c_{i,t_2+1}) = \mathbf{m}\left(F'' \mid M'_1 \mid \cdots \mid M'_{t_2+1}\right). \tag{2}$$

In REDOG's specification, M is chosen uniformly at random among the full-rank matrices in $\mathbb{F}_{2^m}^{\ell \times n}$. Moreover, F is assumed to be another random matrix by [1, Problem 2] so we can safely assume that $\left(F'' \mid M'_1 \mid \cdots \mid M'_{t_2+1}\right) \in \mathbb{F}_{2^m}^{\ell \times \ell}$ is a random matrix, thus having full rank with high probability. We can now compute $\mathbf{m}$ by inverting the system (2) and recover msg.

2.2 First step

We describe a method to produce a system of equations $\mathbf{c}' = \mathbf{m}F' + \mathbf{e}'_2$ where the Hamming weight $\mathsf{wt}_H(\mathbf{e}'_2) = t_2$ and the positions of non-zero entries of $\mathbf{e}'_2$ are known. This can be done because of the following observation.

Remark 2.1. Let F_i denote the i-th column of F, then $c_{2,i} = \mathbf{m}F_i + e_{2,i}$. Assume that $e_{2,i} = e_{2,j}$ for some i, j. Then

$$c_{2,i} + c_{2,j} = \mathbf{m}(F_i + F_j) + e_{2,i} + e_{2,j} = \mathbf{m}(F_i + F_j).$$

Let $\alpha_1, \ldots, \alpha_{t_2} \in \mathbb{F}_{2^m}^*$ be such that $\langle \mathbf{e}_2 \rangle_{\mathbb{F}_2} = \langle \alpha_1, \ldots, \alpha_{t_2} \rangle_{\mathbb{F}_2}$. So each entry $e_{2,i}$ of $\mathbf{e}$ can assume a value in a $\mathbb{F}_2$-vector subspace of $\mathbb{F}_{2^m}$ containing 2^{t_2} elements. This suggests that REDOG chooses $\mathbf{e}_2$ among $2^{(n-k)t_2}$ possibilities (actually, less than $2^{(n-k)t_2}$ as the rank weight constraint $\mathsf{wt}_R(\mathbf{e}_2) = t_2$ must also hold). Label the unknown values of $\langle \mathbf{e}_2 \rangle$ as $\{0, \alpha_1, \alpha_2, \ldots, \alpha_{2^{t_2}-1}\}$, where

$$\alpha_j = \sum_{h=1}^{t_2} z_{j,h}\alpha_h$$

for some $z_{h,j} \in \mathbb{F}_2$ for every $j = t_2 + 1, \ldots, 2^{t_2} - 1$.

Definition 2.2. *Let $t \in \mathbb{N}$ be a positive integer. We define the set of arrangements of t elements over n positions as the set of ordered tuples in $\left(2^{\{1,\ldots,n\}}\right)^t$ defined as*

$$A_{t,n} := \left\{ \mathbf{a} \in \left(2^{\{1,\ldots,n\}}\right)^t \mid \bigcup_{i=1}^{t} a_i = \{1, \ldots, n\}, a_i \cap a_j = \emptyset \ \forall i \neq j \right\}.$$

Proposition 2.3. *Let $\alpha = \{\alpha_1, \ldots, \alpha_{t_2}\} \subset \mathbb{F}_{2^m}^*$ be a set of $\mathbb{F}_2$-linearly independent elements. There exists a one-to-one correspondence between the set $E_{\alpha,n-k} := \{\mathbf{e} \in \mathbb{F}_{2^m}^{n-k} \mid \langle \mathbf{e}_2 \rangle_{\mathbb{F}_2} \subseteq \langle \alpha_1, \ldots, \alpha_{t_2} \rangle_{\mathbb{F}_2}\}$ and $A_{2^{|\alpha|},n-k}$.*

Proof. For $\mathbf{e} \in E_{\alpha,n-k}$, denote by

$$\mathbf{e}^0 := \{i \in \{1,\dots,n-k\} \mid e_i = 0\}$$

and by

$$\mathbf{e}^{\alpha_j} := \{i \in \{1,\dots,n-k\} \mid e_i = \alpha_j\}$$

the positions where $\mathbf{e}_2$ is 0 and α_j for all $j = 1,\dots,2^{t_2}-1$, respectively. We prove that the map

$$\begin{aligned}\varphi_{\alpha,n-k} : E_{\alpha,n-k} &\to A_{2^{|\alpha|},n-k} \\ \mathbf{e} &\mapsto (\mathbf{e}^0, \mathbf{e}^{\alpha_1}, \dots, \mathbf{e}^{\alpha_{2^{t_2}-1}})\end{aligned}$$

is a bijection by showing that it is both injective and surjective. Let $\mathbf{e}, \mathbf{f} \in E_{\alpha,n-k}$ be such that $\mathbf{e} \neq \mathbf{f}$. Then there exists $i \in \{1,\dots,n-k\}$ such that $e_i \neq f_i$. Write $e_i = \alpha_{j_1}$ and $f_i = \alpha_{j_2}$ for some $j_1, j_2 \in \{1,\dots,2^{t_2}-1\}$ with $j_1 \neq j_2$, then $\mathbf{e}^{\alpha_{j_1}} \neq \mathbf{f}^{\alpha_{j_1}}$. It follows that $\varphi_{\alpha,n-k}(\mathbf{e}) \neq \varphi_{\alpha,n-k}(\mathbf{f})$.

On the other hand, let $\mathbf{a} \in A_{2^{|\alpha|},n-k}$ and let $\mathbf{e} \in \mathbb{F}_{2^m}^{n-k}$ be such that $e_j = 0$ for every $j \in a_1$ and $e_j = \alpha_{i-1}$ for every $j \in a_{i-1}$ and every $i = 2,\dots,2^{t_2}$. Then clearly $\varphi_{\alpha,n-k}(\mathbf{e}) = \mathbf{a}$. □

Definition 2.4. *Let* $\mathbf{e} \in \mathbb{F}_{2^m}^{n-k}$ *and* $\alpha = \{\alpha_1,\dots,\alpha_{t_2}\} \subset \mathbb{F}_{2^m}^*$ *such that* $\langle \mathbf{e} \rangle_{\mathbb{F}_2} = \langle \alpha_1,\dots,\alpha_{t_2} \rangle_{\mathbb{F}_2}$. *Then we call* $\varphi_{\alpha,n-k}(\mathbf{e})$ *the arrangement of* $\mathbf{e}$.

Here is an algorithm that rearranges a system of equations according to a given arrangement.

Algorithm 2.5 RearrangeSystem

Input: An arrangement $\mathbf{a} \in A_{2^{t_2},n-k}$, a REDOG's partial ciphertext $\mathbf{c}_2 \in \mathbb{F}_{2^m}^{n-k}$ corresponding to a message $\mathbf{m} \in \mathbb{F}_{2^m}^{\ell}$ under the partial public key $F \in \mathbb{F}_{2^m}^{\ell \times (n-k)}$.
Output: A vector $\mathbf{c}_2'' \in \mathbb{F}_{2^m}^{n-k-t_2}$ and a matrix $F'' \in \mathbb{F}_{2^m}^{\ell \times (n-k-t_2)}$.

1. Fix elements $x_i \in a_i$ for every $i = 2,\dots,t_2+1$;
2. Construct $F'' \in \mathbb{F}_{2^m}^{\ell \times (n-k-t_2)}$ and $\mathbf{c}_2''$ by computing the following columns and values:
   - $F_j'' = F_j$ and $c_{2,i}'' = c_{2,i}$ for every $j \in a_1$;
   - $F_j'' = F_j + F_{x_i}$ and $c_{2,j}'' = c_{2,j} + c_{2,x_i}$ for all $j \in a_i \setminus \{x_i\}$ and $i = 2,\dots,t_2+1$;
   - $F_j'' = F_j + \sum_{h=1}^{t_2} z_{j,h} F_j$ and $c_{2,j}'' = c_{2,j} + \sum_{h=1}^{t_2} z_{j,h} c_{2,j}$ for all $j \in a_i$ and $i = t_2+2,\dots,2^{t_2}$
3. Return F'' and $\mathbf{c}_2''$.

We state the following proposition; the proof is a simple inspection of Algorithm 2.2, and thus omit it.

Proposition 2.6. *Let* $\mathbf{c}_2 = \mathbf{m}F + \mathbf{e}_2$ *be a REDOG ciphertext and* $\varphi_{\alpha,n-k}(\mathbf{e}_2)$ *be the arrangement of* $\mathbf{e}_2$*. Then Algorithm 2.2 returns* $\mathbf{c}_2'' \in \mathbb{F}_2^{n-k-t_2}$ *and* $F'' \in \mathbb{F}_{2^m}^{\ell\times(n-k-t_2)}$ *such that*

$$\mathbf{c}_2'' = \mathbf{m}F''.$$

Proposition 2.6 together with Algorithm 2.2 provides a method that transforms the equation system $\mathbf{c}_2 = \mathbf{m}F + \mathbf{e}_2$ into a smaller system $\mathbf{c}_2'' = \mathbf{m}F''$ that does not involve any noise.

Remark 2.7. Note that Proposition 2.6 assumes knowledge of the arrangement of $\mathbf{e}_2$. We want to stress that, since $\alpha_1, \ldots, \alpha_{t_2}$ are unknown, knowing the arrangement of $\mathbf{e}_2$ does not necessarily mean knowing $\mathbf{e}_2$. This assumption will be satisfied as we iterate over all possible arrangements of $\mathbf{e}_2$.

2.3 Second step

In this subsection, we investigate how to pad the equation system $\mathbf{c}_2'' = \mathbf{m}F''$ with t_2+1 additional equations to uniquely determine $\mathbf{m}$. The idea is to construct these extra equations, combining equations from $\mathbf{c}_1 = \mathbf{m}M + \mathbf{e}_1$. Observe that since $\mathsf{wt}_R(\mathbf{e}_1) = t_1$, then any set $\{e_{1,i_1}, \ldots, e_{1,i_{t_1+1}}\}$ of $t_1 + 1$ entries of $\mathbf{e}_1$ is linearly dependent, i.e. there exist $z_1, \ldots, z_{t_1+1} \in \mathbb{F}_2$ not all zero such that

$$\sum_{j=1}^{t_1+1} z_j e_{1,i_j} = 0.$$

This suggests that given a set of $t_1 + 1$ equations of $\mathbf{c}_1 = \mathbf{m}F + \mathbf{e}_1$ one can search the space of $\mathbb{F}_2$-linear combinations for $t_1 + 1$ non-zero combinations of the equations, which cancels the error factor.

Remark 2.8. Observe that we need to make sure that the columns M_{i_j} for $j = 1, \ldots, t_1 + 1$ are linearly independent, as otherwise we might run into

$$\sum_{j=1}^{t_1+1} z_j M_{i_j} = 0$$

which is a useless equation. However, the probability for this to happen is negligible for each parameter set.

Since we need $t_2 + 1$ extra equations to pad the system, we need to find $t_2 + 1$ equations simultaneously with this method.

2.4 The full attack

For each system rearrangement that we perform in the first step, we need to test all paddings in the second step. Testing the solution of each system we construct implies computing a candidate message $\mathbf{m}' \in \mathbb{F}_{2^m}^{\ell}$ and candidate errors $\mathbf{e}_1' \in \mathbb{F}_{2^m}^{n}$

and $\mathbf{e}'_2 \in \mathbb{F}_{2^m}^{n-k}$ and checking whether the rank weights of $\mathbf{e}'_1$ and $\mathbf{e}'_2$ match t_1 and t_2, respectively. Therefore, combining the two steps described in this section, we obtain the following algorithm.

Algorithm 2.9 PadThaiAttack

Input: A REDOG's ciphertext $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2) \in \mathbb{F}_{2^m}^{2n-k}$ corresponding to a message $\mathbf{m} \in \mathbb{F}_{2^m}^{\ell}$ under the public key $\mathsf{pk} = (M \mid F) \in \mathbb{F}_{2^m}^{\ell \times (2n-k)}$.
Output: The message $\mathbf{m}$.

For each arrangement $\mathbf{a} \in A_{2^{t_2},n-k}$ do:

1. Let $F'', \mathbf{c}'' = \text{RearrangeSystem}(\mathbf{a})$;
2. Pick random sets $J_1, \ldots, J_{t_2+1} \subset \{1, \ldots, n\}$ with $|J_i| = t_1 + 1$;
3. Let M_{J_i} be the matrix consisting of columns of M indexed by J_i;
4. if $\mathsf{rk}(M_{J_i}) \leq t_1 + 1$ for some $i \in \{1, \ldots, t_2 + 1\}$ then go to step 2.
5. For every $(\mathbf{v}_1, \ldots, \mathbf{v}_{t_2+1}) \in \left(\mathbb{F}_2^{t_1+1}\right)^{t_2+1}$ do:
   (a) Compute $M'_i = M_{J_i}\mathbf{v}^\top$ for each $i = 1, \ldots, t_2 + 1$;
   (b) Let $\mathbf{c}_{1,J_i}$ be the vector consisting of the entries of $\mathbf{c}_1$ indexed by J_i;
   (c) Compute $c'_{1,i} = \mathbf{c}_{1,J_i}\mathbf{v}_i^\top$ for each $i = 1, \ldots, t_2 + 1$;
   (d) Let $G := (F'' \mid M'_1, \ldots, M'_{t_2+1})$ and $\mathbf{y} =$ and $\mathbf{y} := (\mathbf{c}'' \mid c'_{1,2}, \ldots, c'_{1,t_2+1}))$;
   (e) Compute $\mathbf{m}' = \mathbf{y}G^{-1}$;
   (f) Compute $\mathbf{e}'_1 = \mathbf{c}_1 - \mathbf{m}'M$ and $\mathbf{e}'_2 = \mathbf{c}_2 - \mathbf{m}'F$;
   (g) If $\mathsf{wt}_R(\mathbf{e}'_1) = t_1$ and $\mathsf{wt}_R(\mathbf{e}'_2) = t_2$ then return $\mathbf{m}' - \mathsf{hash}((\mathbf{e}'_1 \mid \mathbf{e}'_2))$.

In the next section, we give the complexity analysis of our attack and point out some areas of improvement.

3 Analysis of the Pad Thai attack

In this section we describe the complexity of our attack on REDOG described in Algorithm 2.4. Let us start with the following easy lemma.

Lemma 3.1. *The cardinality of* $A_{2^{t_2},n-k}$ *is* $2^{t_2(n-k)}$.

Proof. By Proposition 2.3 there is a bijection between $A_{2^{t_2},n-k}$ and $E_{\alpha,n-k}$ for a fixed set $\alpha = \{\alpha_1, \ldots, \alpha_{t_2}\} \subset \mathbb{F}_{2^m}^*$. The number of elements in $E_{\alpha,n-k}$ is clearly $2^{t_2(n-k)}$. □

A first assessment of the complexity of the Pad Thai attack is given in the following proposition.

Proposition 3.2. *The algorithm 2.4 recovers the message* $\mathbf{m}$ *corresponding to a REDOG ciphertext* $\mathbf{c}$ *under public key* $\mathsf{pk} = (M \mid F)$ *in*

$$\mathcal{O}(2^{(t_1+1)(t_2+1)+t_2(n-k)}\ell^\omega m) \tag{3}$$

field operations, where ω *is the matrix multiplication exponent.*

Proof. The algorithm consists of two nested cycles. The first cycle iterates over all arrangements $A_{2^{t_2},n-k}$, whose cardinality is reported in Lemma 3.1.
The second cycle iterates over all the elements of $\left(\mathbb{F}_2^{t_1+1}\right)^{t_2+1}$, which are $2^{(t_2+1)(t_1+1)}$.
The complexity of each nested cycle is dominated by that of inverting the matrix $G \in \mathbb{F}_{2^m}^{\ell\times\ell}$ whose cost is in $\mathcal{O}(\ell^\omega m)$ where the m factor reflects the fact that G is defined over the extension field $\mathbb{F}_{2^m}$. Combining these factors, we obtain the claimed complexity. □

The following table reports the updated security provided by REDOG based on our attack.

Security parameter	$(n, k, \ell, q, m, r, \lambda, t_1, t_2)$	Pad Thai attack
128	$(30, 6, 25, 2, 59, 12, 3, 6, 2)$	**87.92**
192	$(44, 8, 37, 2, 83, 18, 3, 12, 2)$	**132**
256	$(58, 10, 49, 2, 109, 24, 3, 15, 3)$	**230.53**

Table 2. $\log_2$ of the complexity of the Pad Thai attack for each security level of REDOG according to equation (3).

Table 3 suggests that the combination of parameters of REDOG security level 256 has a smaller loss of security under the Pad Thai attack compared to security levels 128 and 192.

3.1 Attack improvements

In this subsection, we point out an interesting behavior of Algorithm 2.2 of the first step of our attack for parameter sets with $t_2 = 2$, i.e. for security levels 128 and 192. We observe that Algorithm 2.2 rearranges the system $\mathbf{c}_2 = \mathbf{m}F + \mathbf{e}_2$ depending only on the zero positions of $\mathbf{e}_2$. Here is an example.

Example 3.3. Let $\mathbf{e}, \mathbf{f} \in \mathbb{F}_{2^m}^{n-k}$ be the vectors

$$\mathbf{e} = (0, 0, \alpha, \alpha, \beta, \alpha + \beta)$$

and

$$\mathbf{f} = (0, 0, \beta, \beta, \alpha + \beta, \alpha).$$

These two vectors have the same arrangement under different labeling of their entries. Now, let $x_1 = 3$ and $x_2 = 5$. Inspecting Algorithm 2.2, we can see that the output $\mathbf{e}''$ and $\mathbf{f}''$ coincide.

This means that we can run the first cycle on a subset of the arrangements $A_{2^{t_2},n-k}$ as Algorithm 2.2 does not distinguish between errors having the same arrangement. A similar improvement exists for security 256 that we have not yet investigated.

Another improvement comes from noting that we are considering arrangements that correspond to error vectors $\mathbf{e}_2$ that have rank weight $\mathsf{wt}_R(\mathbf{e}_2) \leq t_2$.

Considering only arrangements for error vectors with rank weight exactly t_2 slightly reduces the number of arrangements that we need to iterate on in the first step.

References

[1] Jon-Lark Kim et al. *REDOG*. Submission to KpqC Round 2. 2023.