A Distributed Coded Computation Scheme Based on a Gabidulin Code and the Performance Evaluation

Koki Kazama
Waseda University
Email: kwaazsaemdaa@fuji.waseda.jp

Toshiyasu Matsushima
Waseda University
Email: toshimat@waseda.jp

***Abstract*—We focus on a distributed coded computation scheme for matrix multiplication. In this system, the product matrix is encoded and decoded through the overall system to correct errors in computation. We propose a new distributed coded computation scheme, for one example, a scheme based on a Gabidulin code, and evaluate the computation time complexity and the error-correcting capability of the overall system.**

I. Introduction

We focus on a distributed computation scheme in which errors are corrected in computing multiplication of two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ on a finite field $\mathbb{F}_q$. In the system of a distributed computation scheme, the main computer (master) partitions the matrix and distributes them to multiple computers (workers), and (2) workers perform parallel computing. This system has an advantage of decreasing the computation time complexity, while this has a disadvantage of increasing possibility of occurring errors in computation. A distributed coded computation scheme (DCC) is a distributed computation scheme using error-correcting codes to correct errors in computation. On performance evaluation on DCCs, (a) its computation time complexity (CTC) and (b) its error-correcting capability of the overall system are important criteria while they are a tradeoff.

First, we propose a new distributed coded computation scheme called *Group-type Distributed Coded Computation scheme* (GDCC) and, as one example, a *GDCC based on a Gabidulin code* (GDCCG). A Gabidulin code encodes a matrix over $\mathbb{F}_q$ and correct an error matrix $\boldsymbol{E}$ if $\mathrm{rank}(\boldsymbol{E}) \leq t$, where t is a constant defined later. The key idea is that this distributed computation system performs a process over $\mathbb{F}_q$, which is equivalent to encoding a vector over an extension field $\mathbb{F}_{q^m}$ corresponding $\boldsymbol{AB}$ to a codeword over $\mathbb{F}_{q^m}$. Thus the encoder in the GDCCG system encodes all columns of$\boldsymbol{AB}$ together while the encoders in previous DCC systems encode each column of $\boldsymbol{AB}$. This enables to correct error matrices whose columns are depending on each other and enables to decrease the overall CTC of the system simultaneously. The GDCCG is a little different from the scheme in [1] because more workers are used and they are equally partitioned into multiple groups and the parallel computation of matrix products is performed within each group. Moreover, a more general scheme (GDCC) is proposed and the evaluation of (a) in this paper is more detailed in this paper.

Second, we evaluate (a) the CTC and (b) the error-correcting capability of the GDCC and show the advantages as follows. In the evaluation on (a), we evaluate the CTC of the GDCCG and the stand-alone system (SA), in which the master computes $\boldsymbol{AB}$ solely in the system, and show the condition of parameters (the number of workers and groups) in which the GDCC is supreme to the SA. We cannot generally evaluate (a) on GDCC because the CTC of the decoding algorithm is depending on the code. Thus we evaluate (a) only on GDCCG. We define the CTC of a DCC system as the number of four arithmetic operations over $\mathbb{F}_q$ in parallel computing of each worker and decoding of the master. To evaluate the number of operations of the encoder and the decoder using a Gabidulin code over $\mathbb{F}_{q^m}$ in the GDCCG system, we first show how many times it is necessary when the operation of $\mathbb{F}_{q^m}$ is decomposed into the operation of $\mathbb{F}_q$, and then we count them. In the evaluation on (b), we show error matrices which the GDCC system can correct. Specifically, we show that the GDCCG system can correct an error matrix if $\mathrm{rank}(\boldsymbol{E}) \leq t$. This indicates that the decoder of the GDCC system correct errors for all workers if it satisfies some conditions while the previous systems cannot correct.

The rest of this paper is organized as follows. First, as a basis for the proposition, we clarify the correspondence in principle between previous distributed coded computations (DCCs) and error-correcting codes (ECCs). Specifically, an encoding of error-correcting codes corresponds to an encoding of $\boldsymbol{AB}$ into the output result $\Pi(\boldsymbol{AB})$ of the system. GDCCG encodes all columns of $\boldsymbol{AB}$ together. Next, we explain how to construct GDCC. Finally, we evaluate (b) the error-correcting capability of the group distributed coding scheme, and (a) and (b) of GDCCG. In particular, for the evaluation on (a), we derive the parameter conditions under which the CTC of the GDCCG system is less than that of the SA system. For the evaluation on (b), we show that GDCCG can correct an error matrix $\boldsymbol{E}$, while previous schemes cannot correct it.

In this paper, q is a power of 2. Positive integers $n, k_{\mathrm{A}}, k_{\mathrm{B}}, l$ satisfy $2 \leq k_{\mathrm{A}} < n, 2 \leq k_{\mathrm{B}}$ and $2 \leq l$. m is $\max\{k_{\mathrm{B}}, n\}$. For two integers m and n which satisfies $m \leq n$, $[m, n]$ denotes $[m, n] := \{ m, m+1, \ldots, n \}$. $[n]$ denotes $[1, n]$. All vectors are column vectors except specifically noted. $\boldsymbol{E}^\top$ is the transpotent of a matrix $\boldsymbol{A}$. $\mathbb{F}_q$ is a finite field with q elements. $\mathbb{F}_q^{n \times b}$ denotes the set of all $n \times b$ matrices over $\mathbb{F}_q$, and $\mathbb{F}_q^n := \mathbb{F}_q^{n \times 1}$. $\boldsymbol{e}_{\cdot j} \in \mathbb{F}_q^n$ denotes j-th column of a matrix $\boldsymbol{E} \in \mathbb{F}_q^{n \times b}$ $\boldsymbol{e}_{i\cdot} \in \mathbb{F}_q^b$ denotes the transpotent of i-th row vector. Thus the matrix $\boldsymbol{E}$ is $(\boldsymbol{e}_{\cdot 1}, \ldots, \boldsymbol{e}_{\cdot b}) = (\boldsymbol{e}_{1\cdot}, \ldots, \boldsymbol{e}_{n\cdot})^\top$. For any

set $A \subset [n]$, we define a matrix $\boldsymbol{G}^{\top}_{A.} \in \mathbb{F}_q^{|A| \times k_{\mathrm{A}}}$ as a matrix constructed from all $i(\in A)$-th row $\boldsymbol{g}_{i.}^{\top} \in \mathbb{F}_q^{1 \times k_{\mathrm{A}}}$ of the matrix $\boldsymbol{G}$.

Definition 1.1 ($\boldsymbol{v}$, $\boldsymbol{f}^s$): $v_1, \ldots, v_m \subset \mathbb{F}_{q^m}$ are linearly independent over $\mathbb{F}_q$ and $v_i = v_1^{q^{i-1}}$ for any $i \in [m]$, i.e. $\{v_1, \ldots, v_m\}$ is a normal basis over $\mathbb{F}_{q^m}$. We define $\boldsymbol{v}$ as a vector $(v_1, \ldots, v_m)$. For any $s \in \mathbb{N}$, a linear homomorphism $\boldsymbol{f}^s \colon \mathbb{F}_{q^m}^s \to \mathbb{F}_q^{s \times m}$ over $\mathbb{F}_q$ outputs a matrix $\boldsymbol{X} \in \mathbb{F}_q^{s \times m}$ which satisfies $\boldsymbol{x} = \boldsymbol{X}\boldsymbol{v}$ from the input $\boldsymbol{x} \in \mathbb{F}_{q^m}^s$. When $s = 1$, we wright f^1 as $\boldsymbol{f}^s$.

? denotes the symbol representing an erasure or decoding failure. We formally define the sum and difference of a and ? as ? for any $a \in \mathbb{F}_q$.

II. Relation between ECCs and Previous DCCs

As a basis for the proposition, we explain the correspondence in principle between previous DCCs and ECCs in [1]. Specifically, an encoding of error-correcting codes corresponds to an encoding of $\boldsymbol{AB}$ into the output result $\Pi(\boldsymbol{AB})$ of the system. GDCCG encodes all columns of $\boldsymbol{AB}$ together.

This system encodes $\boldsymbol{AB}$ to $\Pi(\boldsymbol{AB})$ using a function $\Pi \colon \mathbb{F}_q^{k_{\mathrm{A}} \times k_{\mathrm{B}}} \to \mathcal{C}(\subset \mathbb{F}_q^{n \times b})$, and decode a matrix $\boldsymbol{Y} = \Pi(\boldsymbol{AB}) + \boldsymbol{E}$, where $\boldsymbol{E} \coloneqq (\boldsymbol{e}_{1.}, \ldots, \boldsymbol{e}_{n.})^{\top}$. $\Pi(\boldsymbol{AB})$ is a matrix whose i-th row is $\boldsymbol{g}_{i.}^{\top}\boldsymbol{AB}$ for any $i \in [n]$. $\boldsymbol{AB}$, $\Pi(\boldsymbol{AB})$, $\boldsymbol{E}$, $\boldsymbol{Y}$, $\boldsymbol{G}$, Π, ψ and $\mathcal{C}$ correspond with information, codeword, error ,recieved word, generator matrix, encoder, decoder, code in ECCs, respectively. We call them product matrix, *codeword (matrix)*, *error (matrix)*, *recieved matrix*, *generator matrix*, *encoder*, *decoder*, *code*, respectively.

We propose a scheme which $\Pi(\boldsymbol{AB})$ is a codeword $\boldsymbol{f}^n(\boldsymbol{GA}(\boldsymbol{B}, \boldsymbol{0}_{l \times (m-k_{\mathrm{B}})})\boldsymbol{v})$ of a code $\mathcal{C}(\subset \mathbb{F}_q^{n \times m})$. We explain more details after. In particular, the scheme whose matrix-encoder Π is a Gabidulin encoder can correct low rank error matrices $\boldsymbol{E}$, which cannot be corrected by previous schemes. The workers in this system are partitioned into multiple groups for shortening the time complexy of the system.

In this section, we propose a computation scheme called *GDCC*. Moreover, we propose a scheme called *GDCC with a Gabidulin code* (GDCCG) as an example of a GDCC.

First, we define symbols. We define $\tilde{\mathcal{C}} \subset \mathbb{F}_{q^m}^n$ as an (n, k_{A}) linear code over $\mathbb{F}_{q^m}$. This code has a generator matrix $\boldsymbol{G} \in \mathbb{F}_{q^m}^{n \times k_{\mathrm{A}}}$ of a canonical systematic encoder. $\mathcal{C} \coloneqq \boldsymbol{f}^n(\tilde{\mathcal{C}})(\subset \mathbb{F}_q^{n \times m})$ is a linear code over $\mathbb{F}_q$. The decoder of $\mathcal{C}$ is $\psi \colon \mathbb{F}_q^{n \times m} \to \mathcal{C} \cup \{?\}$. We define a set $\mathcal{E}$ $(\subset \mathbb{F}_q^{n \times m})$ as the set of all error matrices which can be corrected rightly, i.e. for any matrix $\boldsymbol{E} \in \mathcal{E}$ and any codeword $\boldsymbol{C} \in \mathcal{C}$, it holds that $\psi(\boldsymbol{C} + \boldsymbol{E}) = \boldsymbol{C}$.

Lemma 2.1: Any codeword of a linear code $\mathcal{C} = \boldsymbol{f}^n(\tilde{\mathcal{C}})$ over $\mathbb{F}_q$ is $\boldsymbol{f}^n(\boldsymbol{GMv})$ for some matrix $\boldsymbol{M} \in \mathbb{F}_q^{k_{\mathrm{A}} \times m}$.

Proof : A codeword an (n, k_{A}) linear code $\tilde{\mathcal{C}} \subset \mathbb{F}_{q^m}^n$ over $\mathbb{F}_{q^m}$ is $\boldsymbol{GMv}$ for some matrix $\boldsymbol{M} \in \mathbb{F}_q^{k_{\mathrm{A}} \times m}$. □

We propose a GDCC $((\pi_{ij} | i \in [n_{\mathrm{A}}], j \in [n_{\mathrm{B}}]), \boldsymbol{G}, \psi)$ when $q, k_{\mathrm{A}}, k_{\mathrm{B}}, l, n, n_{\mathrm{A}}$ and n_{B} are given. In this scheme, we use the master and $n_{\mathrm{A}}n_{\mathrm{B}}$ workers which are partitioned into multiple groups, where $n_{\mathrm{A}} \in \mathbb{N}$ devides n and $n_{\mathrm{B}} \in \mathbb{N}$ devides m. All workers are equally partitioned into n_{A} groups. The j-th worker of the i-th group is called *a worker* $(i, j) \in [n_{\mathrm{A}}] \times [n_{\mathrm{B}}]$ (Figure 1). We assume that errors occurs only when workers computes something. The master has the matrix $\boldsymbol{G} \in \mathbb{F}_{q^m}^{n \times k_{\mathrm{A}}}$ mentioned above and a function ψ. A function π_{ij} are stored in each worker $(i, j) \in [n_{\mathrm{A}}] \times [n_{\mathrm{B}}]$.

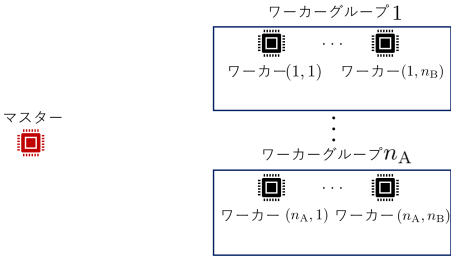

Fig. 1. the master and the workers

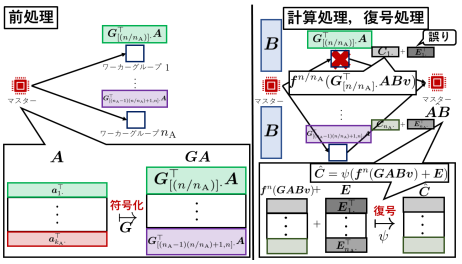

Fig. 2. the flow of the DCC

In GDCC, when the matrix $\boldsymbol{A}$ is input to the master, then the master and all workers perform *preprocess*. For any time when the matrix $\boldsymbol{B}$ is input to the master, all workers perform *Computing Process*, and then the master performs *Decoding Process*. The result of Decoding Process of the master is $\hat{\boldsymbol{AB}} \in \mathbb{F}_q^{k_{\mathrm{A}} \times k_{\mathrm{B}}}$, which is an estimated results of $\boldsymbol{AB}$. See the details in below (Figure 2).

[$\langle$*Preprocess*$\rangle$] (Figure 3) The master encodes the matrix $\boldsymbol{A}$ to

$$\boldsymbol{GA} = \begin{pmatrix} \boldsymbol{G}^{\top}_{[1, n/n_{\mathrm{A}}].}\boldsymbol{A} \\ \vdots \\ \boldsymbol{G}^{\top}_{[(n_{\mathrm{A}}-1)(n/n_{\mathrm{A}})+1, n].}\boldsymbol{A} \end{pmatrix} \in \mathbb{F}_{q^m}^{n \times l}. \tag{1}$$

The master store $\boldsymbol{G}^{\top}_{[(i-1)(n/n_{\mathrm{A}})+1, i(n/n_{\mathrm{A}})].}\boldsymbol{A} \in \mathbb{F}_{q^m}^{(n/n_{\mathrm{A}}) \times l}$ in all workers of each group $i \in [n_{\mathrm{A}}]$. Then, the workers in the i-th group store the j'-th symbol of a matrix $f^1(\boldsymbol{g}_{i'.}^{\top}\boldsymbol{a}_{\cdot l'}v_{m'}) \in$

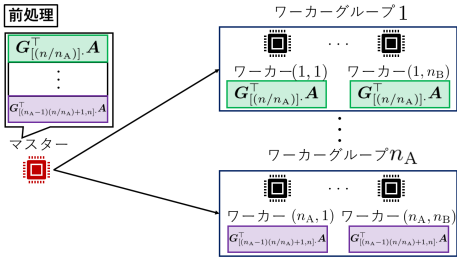

Fig. 3. Preprocess

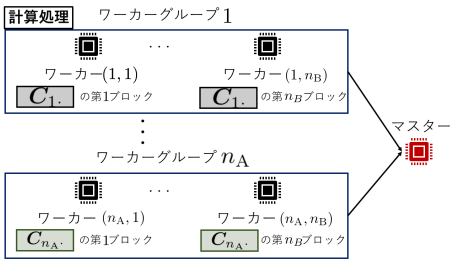

Fig. 4. Computing Process

$\mathbb{F}_q^{1\times m}$ for all $l' \in [l]$, $m' \in [m]$, $i' \in [(i-1)(n/n_{\mathrm{A}})+1, i(n/n_{\mathrm{A}})]$ and $j' \in [(j-1)(m/n_{\mathrm{B}})+1, j(m/n_{\mathrm{B}})]$. We define this as $\tilde{a}_{i'l'm'j'} \in \mathbb{F}_q$. $\boldsymbol{a}_{\cdot l'} \in \mathbb{F}_q^{k_{\mathrm{A}}}$is the l'-th column of the matrix $\boldsymbol{A}$. $\tilde{a}_{i'l'm'j}$ can be computed from $\boldsymbol{g}_{i'\cdot}^{\top}\boldsymbol{A} = (\boldsymbol{g}_{i'\cdot}^{\top}\boldsymbol{a}_{\cdot 1}, \ldots, \boldsymbol{g}_{i'\cdot}^{\top}\boldsymbol{a}_{\cdot l})$.

[⟨*Computing Process*⟩] (Figure 4) The master sends the matrix $\boldsymbol{B}$ to all workers. Hereafter, we define $\boldsymbol{B}' = \boldsymbol{B}$ if $n \le k_{\mathrm{B}}$, and $\boldsymbol{B}' = (\boldsymbol{B}, \boldsymbol{0})$ if $n > k_{\mathrm{B}}$, where $\boldsymbol{0}$ is an $l\times(m-k_{\mathrm{B}})$ zero matrix over $\mathbb{F}_q$. Each worker (i,j) computes the $(j-1)(m/n_{\mathrm{B}})+1, \ldots, j(m/n_{\mathrm{B}})$-th columns (we think them as *j-th block*) of an $(m/n_{\mathrm{B}})\times m$ matrix

$$\boldsymbol{C}_{i\cdot} \coloneqq \boldsymbol{f}^{m/n_{\mathrm{B}}}\left(\boldsymbol{G}_{[(i-1)(m/n_{\mathrm{B}})+1, i(m/n_{\mathrm{B}})]\cdot}\boldsymbol{A}\boldsymbol{B}'\boldsymbol{v}\right). \quad (2)$$

This computation can be done by computing $\sum_{l'\in[l]}\sum_{m'\in[m]}\tilde{a}_{i'l'm'j'}b_{l'm'}$ from $\boldsymbol{B}$ for any $(i',j') \in [(i-1)(m/n_{\mathrm{B}})+1, i(m/n_{\mathrm{B}})]\times[(j-1)(m/n_{\mathrm{B}})+1, j(m/n_{\mathrm{B}})]$. See proposition 2.1.

For any $(i,j) \in [n_{\mathrm{A}}]\times[n_{\mathrm{B}}]$, we define the correct computing result of any worker (i,j) as $\pi_{ij}(\boldsymbol{g}_{i\cdot}^{\top}\boldsymbol{A}, \boldsymbol{B})$. we define *product function* as the function $\pi_{ij}\colon \mathbb{F}_{q^m}^{(n/n_{\mathrm{A}})\times l} \times \mathbb{F}_q^{l\times k_{\mathrm{B}}} \to \mathbb{F}_q$ which computes the $(j-1)(m/n_{\mathrm{B}})+1, \ldots, j(m/n_{\mathrm{B}})$-th columns of the matrix in the equation (2) from $\boldsymbol{G}_{[(i-1)(n/n_{\mathrm{A}})+1, i(n/n_{\mathrm{A}})]\cdot}^{\top}\boldsymbol{A} \in \mathbb{F}_{q^m}^{(n/n_{\mathrm{A}})\times l}$ and $\boldsymbol{B}$. We assume that the error $e_{i'j'} \in \mathbb{F}_q$ occurs in computing the j'-th symbol of $f^1\left(\boldsymbol{g}_{i'\cdot}^{\top}\boldsymbol{A}\boldsymbol{B}'\boldsymbol{v}\right) \in \mathbb{F}_q^{1\times m}$. The master recieves a matrix $\boldsymbol{Y} \coloneqq \boldsymbol{f}^n(\boldsymbol{G}\boldsymbol{A}\boldsymbol{B}'\boldsymbol{v}) + \boldsymbol{E}$, where $\boldsymbol{E}$ is a matrix whose (i',j')-th entry is $e_{i'j'}$ for any $(i',j') \in [n]\times[m]$. This matrix is constructed from all output results of all workers.

[⟨*Decoding Process*⟩] The master gets $\psi(\boldsymbol{Y})$ from the matrix $\boldsymbol{Y}$ with a function $\psi : \mathbb{F}_q^{n\times m} \to \mathcal{C}\cup\{?\}$, where a symbol $? \notin \mathcal{C}$ represents a fact that the master cannot get an estimated matrix $\hat{\boldsymbol{A}\boldsymbol{B}}$. Since the generator matrix is a generator matrix of a canonical systematic encoder, if $\psi(\boldsymbol{Y}) \in \mathcal{C}$, the master gets $\hat{\boldsymbol{A}\boldsymbol{B}}$ from $\psi(\boldsymbol{Y})$.

Assumption 2.1: We do not include CTC of the preprocess in the evaluation on (a) since $\boldsymbol{A}$ is input only once. Moreover, we do not include any communication time among workers and the master.

Proposition 2.1: For any $(i',j') \in [(i-1)(m/n_{\mathrm{B}})+1, i(m/n_{\mathrm{B}})]\times[(j-1)(m/n_{\mathrm{B}})+1, j(m/n_{\mathrm{B}})]$, the (i',j')-th entry of a matrix $\boldsymbol{f}^n(\boldsymbol{G}\boldsymbol{A}\boldsymbol{B}\boldsymbol{v})$ is $\sum_{l'\in[l]}\sum_{m'\in[k_{\mathrm{B}}]}\tilde{a}_{i'l'm'j'}b_{l'm'}$.

Proof : It is the j'-th entry of

$$\begin{aligned}
&f^1\left(\boldsymbol{g}_{i'\cdot}^{\top}\boldsymbol{A}\boldsymbol{B}'\boldsymbol{v}\right)\\
&= f^1\left(\begin{pmatrix}\boldsymbol{g}_{i'\cdot}^{\top}\boldsymbol{a}_{\cdot 1} & \ldots & \boldsymbol{g}_{i'\cdot}^{\top}\boldsymbol{a}_{\cdot l}\end{pmatrix}\begin{pmatrix}\sum_{m'\in[m]} b'_{1m'}v_{m'}\\ \vdots \\ \sum_{m'\in[m]} b'_{lm'}v_{m'}\end{pmatrix}\right)\\
&= f^1\left(\begin{pmatrix}\boldsymbol{g}_{i'\cdot}^{\top}\boldsymbol{a}_{\cdot 1} & \ldots & \boldsymbol{g}_{i'\cdot}^{\top}\boldsymbol{a}_{\cdot l}\end{pmatrix}\begin{pmatrix}\sum_{m'\in[k_{\mathrm{B}}]} b_{1m'}v_{m'}\\ \vdots \\ \sum_{m'\in[k_{\mathrm{B}}]} b_{lm'}v_{m'}\end{pmatrix}\right)\\
&= f^1\left(\sum_{l'\in[l]}\boldsymbol{g}_{i'\cdot}^{\top}\boldsymbol{a}_{\cdot l'}\left(\sum_{m'\in[k_{\mathrm{B}}]} b_{l'm'}v_{m'}\right)\right)\\
&= f^1\left(\sum_{l'\in[l]}\sum_{m'\in[k_{\mathrm{B}}]} b_{l'm'}\left(\boldsymbol{g}_{i'\cdot}^{\top}\boldsymbol{a}_{\cdot l'}\right)v_{m'}\right)\\
&= f^1\left(\sum_{j\in[m]}\left(\sum_{l'\in[l]}\sum_{m'\in[k_{\mathrm{B}}]}\tilde{a}_{i'l'm'j}b_{l'm'}\right)v_j\right)\\
&= \left(\sum_{l'\in[l]}\sum_{m'\in[k_{\mathrm{B}}]}\tilde{a}_{i'l'm'1}b_{l'm'}, \ldots, \sum_{l'\in[l]}\sum_{m'\in[k_{\mathrm{B}}]}\tilde{a}_{i'l'm'm}b_{l'm'}\right).\square
\end{aligned}$$

Corollary 2.1: Any worker $(i,j) \in [n_{\mathrm{A}}]\times[n_{\mathrm{B}}]$ performs $(lk_{\mathrm{B}}-1)(mn/n_{\mathrm{A}}n_{\mathrm{B}})$ additions over $\mathbb{F}_q$ and $lk_{\mathrm{B}}(mn/n_{\mathrm{A}}n_{\mathrm{B}})$ multiplications over $\mathbb{F}_q$ in Computing Process.

Proof : Proposition 2.1 shows that any worker (i,j) computes the j'-th symbol $\sum_{l'\in[l]}\sum_{m'\in[k_{\mathrm{B}}]}\tilde{a}_{i'l'm'j'}b_{l'm'}$ of $f^1(\boldsymbol{g}_{i'\cdot}^{\top}\boldsymbol{A}\boldsymbol{B}'\boldsymbol{v})$ for any $(i',j') \in [(i-1)(m/n_{\mathrm{B}})+1 : i(m/n_{\mathrm{B}})]\times[(j-1)(m/n_{\mathrm{B}})+1, j(m/n_{\mathrm{B}})]$ in Computing Process. Moreover, each $\tilde{a}_{i'l'm'j'}$ is already computed in Preprocess. Thus the worker (i,j) performs $lk_{\mathrm{B}}-1$ additions over $\mathbb{F}_q$ and lk_{B} multiplications over $\mathbb{F}_q$ to compute $\sum_{l'\in[l]}\sum_{m'\in[k_{\mathrm{B}}]}\tilde{a}_{i'l'm'j'}b_{l'm'}$ from the input $\boldsymbol{B}$. The worker (i,j) performs this computation for all $(i',j') \in [(i-1)(m/n_{\mathrm{B}})+1, i(m/n_{\mathrm{B}})]\times[(j-1)(m/n_{\mathrm{B}})+1, j(m/n_{\mathrm{B}})]$. $\square$

We propose a GDCC based on a Gabidulin code as an example of GDCCs.

Definition 2.1 (Gabidulin code [2]): Let $m, k\ (\in \mathbb{N})$ satisfy $m \ge n \ge k$. Let $h_1, \ldots, h_n \in \mathbb{F}_{q^m}$ be independent over $\mathbb{F}_q$. Let $\boldsymbol{H} \in \mathbb{F}_{q^m}^{n\times(n-k)}$ is a matrix whose (i,j)-th entry is $h_i^{q^{j-1}}$ for any $(i,j) \in [n]\times[k]$. An (n,k) linear code $\tilde{\mathcal{C}}_{\mathrm{G}} \subset \mathbb{F}_{q^m}^n$ over $\mathbb{F}_{q^m}$ which has a parity check matrix $\boldsymbol{H}$ is called an (n,k) *Gabidulin code* over $\mathbb{F}_{q^m}$.

Definition 2.2 (GDCCG): Let $\boldsymbol{G} \in \mathbb{F}_{q^m}^{n\times k_{\mathrm{A}}}$ is a canonical systematic generator matrix of a (n,k_{A}) Gabidulin code over $\mathbb{F}_{q^m}$. ψ is a bounded rank distance decoder of this code. This GDCC is called *GDCCG*.

III. Performance Evaluations on the Proposed Schemes

We evaluate the of (a) CTC of the GDCCG system and (b) error-correcting capablity of the GDCC system and the GDCCG system. We evaluate (b) on GDCC and GDCCG and show that GDCCG corrects an error matrix which previous

schemes cannot correct. Moreover, we compare (a) on the stand-alone scheme (SA) and that of GDCCG and give the parameter condition when GDCCG is supereme to the SA.

A. Evaluations on Error-Correcting Capabilities

We evaluate (b) on GDCC and GDCCG.

Theorem 3.1 (evaluation on (b) on GDCC): The GDCC system computes correctly if the error matrix $\boldsymbol{E}$ is in $\mathcal{E}$.

Proof : GDCC outputs $\boldsymbol{f}^n(\boldsymbol{GAB'v}) \in \mathbb{F}_q^{n\times m}$ from $\boldsymbol{A}$ and $\boldsymbol{B}$ when no error occurs in the computation. If $\boldsymbol{E} \in \mathcal{E}$, then $\psi(\boldsymbol{f}^n(\boldsymbol{GAB'v}) + \boldsymbol{E}) = \boldsymbol{f}^n(\boldsymbol{GAB'v})$ since $\boldsymbol{f}^n(\boldsymbol{GAB'v}) \in \mathcal{C}$ from Lemma 2.1. Thus this scheme corrects all error matrices in $\mathcal{E}$. □

Theorem 3.2 (evaluation on (b) on GDCCG): The GDCCG system computes correctly if the error matrix $\boldsymbol{E}$ satisfies $\mathrm{rank}\boldsymbol{E} \leq t$.

Proof : An (n, k_{A}) Gabidulin code over $\mathbb{F}_{q^m}$ can correct an error matrix $\boldsymbol{E}$ if $\mathrm{rank}(\boldsymbol{E}) \leq t$. □

B. Evaluations on Computation Time Complexities

1) Definitions and Assumptions on Computation Time Complexities: We assume some assumptions in this paper to evaluate CTC of a decoding algorithm [3] of a Gabidulin code.

Definition 3.1: We define *computation time complexity* (CTC) of a process as the number of four arithmetic operations over $\mathbb{F}_q$ which the system performs in the process from input to output. An addition, a substraction, a multiplication and an inversion are equally treated as one operation in the evaluation on CTC. *CTC of the system* is the overall CTC from input $\boldsymbol{B}$ to output $\hat{\boldsymbol{AB}}$.

We use corresponding between $\boldsymbol{a} \in \mathbb{F}_{q^m}^n$ and $\boldsymbol{f}^n(\boldsymbol{a})$ in the proposed scheme. We assume Assumption 3.1, also assumed in [4] [3] [5].

Assumption 3.1: We do not include CTC of computing $\boldsymbol{f}^n(\boldsymbol{a}) \in \mathbb{F}_q^{n\times m}$ from $\boldsymbol{a} \in \mathbb{F}_{q^m}^n$ in the evaluation on (a). We do not include that of computing $(\boldsymbol{f}^n)^{-1}(\boldsymbol{A}) \in \mathbb{F}_{q^m}^n$ from $\boldsymbol{A} \in \mathbb{F}_q^{n\times m}$.

Proposition 3.1: For any prime power q and positive integer m and $a \in \mathbb{F}_{q^m}$, if $f^1(a) = (a_1, \ldots, a_m) \in \mathbb{F}_q^{1\times m}$, then $f^1(a^{q^i}) = (a_{m-i+1}, \ldots, a_m, a_1, a_2, \ldots, a_{m-i})$.

Definition 3.2 (cyclic shift [4]): For any q, m and a, we define i-th*cyclic shift up* as $\boldsymbol{a}^{\uparrow i} := f(a^{q^i})$ and *cyclic shift down* $\boldsymbol{a}^{\downarrow i} := f(a^{q^i})$.

We assume Assumption 3.2, also assumed in [4] [3] [5].

Assumption 3.2: We do not evaluate time complexities of cyclic shifts up or down.

To decompose four arithmetic operations over $\mathbb{F}_{q^m}$ to those over $\mathbb{F}_q$, we assume some conditions on q and m.

Definition 3.3 (Multiplication Table [4]): For any $i \in [m]$, we define $T_{i1}, \ldots, T_{im} \in \mathbb{F}_q$ as the elements uniquely determined by $v_1 v_i = \sum_{j\in[m]} T_{ij} v_j$. $\boldsymbol{T} := (T_{ij})_{(i,j)\in[m]^2} \in \mathbb{F}_q^{m\times m}$ is called *multiplication table*. We define $C(\boldsymbol{T}) \in \mathbb{Z}$ as the number of nonzero entries of $\boldsymbol{T}$. $C(\boldsymbol{T})$ is called the *complexity of the normal basis* $\{v_1, \ldots, v_m\}$.

Lemma 3.1 ([6]): An addition over $\mathbb{F}_{q^m}$ costs$m$ additions over $\mathbb{F}_q$. Moreover, an multiplication over $\mathbb{F}_{q^m}$ costs $m(C(\boldsymbol{T})+1) - m^2 - 1$ additions over $\mathbb{F}_q$ and $m(C(\boldsymbol{T})+m)$ multiplications over $\mathbb{F}_q$

Definition 3.4 (Optimal Normal Basis [7]): For any normal basis $\{v_1, \ldots, v_m\}$ and multicative table $\boldsymbol{T}$, $C(\boldsymbol{T}) \geq 2m-1$. The normal basis $\{v_1, \ldots, v_m\}$ is called an *optimal* normal basis if it achieves this lower bound.

We set a normal basis $\{v_1, \ldots, v_m\} \in \mathbb{F}_{q^m}$ over $\mathbb{F}_q$ an *optimal* normal basis.

Lemma 3.2 ([7]): An optimal normal basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$ exists iff q and m satisfies the following. $\log_2 q$ and m are prime with each other. $2m+1$ is a prime number. A multicative group $(\mathbb{Z}/(2m+1)\mathbb{Z})^\star = (\mathbb{Z}/(2m+1)\mathbb{Z}) \setminus \{0\}$ is generated from 2 and -1.

Assumption 3.3 (Assumption for evaluation on (a)): We assume the condition of Lemma 3.2.

Proposition 3.2: q and m satisfies Assumption 3.3. An addition, which is also a substraction, over $\mathbb{F}_{q^m}$ costs m additions over $\mathbb{F}_q$. A multiplication over $\mathbb{F}_{q^m}$ costs m^2-1 additions and m^2 multiplication over $\mathbb{F}_q$. An inversion over $\mathbb{F}_{q^m}$ costs $(m^2-1)(2\lfloor\log_2(m\log_2 q - 1)\rfloor + 2)$ additions and $m^2(2\lfloor\log_2(m\log_2 q - 1)\rfloor + 2)$ multiplications over $\mathbb{F}_q$.

In this paper, $h_1, \ldots, h_n$ in Definition 2.1 satisfy $h_i = v_i$ for any $i \in [n]$.

Corollary 3.1: q and m satisfies the condition of Lemma 3.2. We define $t = \lfloor (n-k_{\mathrm{A}})/2 \rfloor$ and $d = n - k_{\mathrm{A}} + 1$. Then D is an upper bound of the complexity of the decoding algorithm [3] of an (n, k_{A})Gabidulin code over $\mathbb{F}_{q^m}$. D is

$$
\begin{aligned}
&2mnt + m^2 + m - 1 + \frac{2}{3}(m(m-1)(2m-1) - t(t-1)(2t-1)) \\
&- (t-2)(m-t) - (t-1)(m^2 - m - t^2 + t) \\
&+ (dn + d^2 - t^2 + 2dt + mt - n - 4d - t - 1)m \\
&+ (dn + 3d^2 + 3t^2 - 4dt + mt - n - 9d + 9t + 5)(2m^2 - 1) \\
&+ 2t(2m^2 - 1)(2\lfloor\log_2(m\log_2 q - 1)\rfloor). \qquad (3)
\end{aligned}
$$

2) Evaluations and A Comparison to the Stand-Alone System: We evaluate (a) on the GDCCG system in case $\mathrm{rank}\boldsymbol{E} \leq t$ using D defined in Eq. (3). The CTC of the GDCC system is the sum of the number of four arithmetic operations over $\mathbb{F}_q$ of each worker and that of Decoding Process of the master.

Theorem 3.3 (evaluation on (a) on GDCCG): Under Assumption 3.3, the sum of CTC of Computation Process of each worker and that of Decoding Process of the master in the GDCCG system is at most

$$(2k_{\mathrm{B}}l - 1)(mn/n_{\mathrm{A}}n_{\mathrm{B}}) + D. \qquad (4)$$

Proof : We showed from Corollary 2.1 that CTC of Computation Process of each worker $(i, j) \in [n_{\mathrm{A}}] \times [n_{\mathrm{B}}]$ is at most $(2k_{\mathrm{B}}l - 1)(mn/n_{\mathrm{A}}n_{\mathrm{B}})$. The master computes $\boldsymbol{f}^n(\boldsymbol{GAB'v})$ from $\boldsymbol{f}^n(\boldsymbol{GAB'v}) + \boldsymbol{E}$ with the bounded rank-distance decoder in Decoding Process. We do not include time to compute $\boldsymbol{AB}$ from $\boldsymbol{f}^n(\boldsymbol{GAB'v})$ since $\boldsymbol{G}$ is a generator matrix of a canonical systematic encoder. Thus the CTC is at most D. □

We compare the CTC on the stand-alone scheme.

Definition 3.5: We define *stand-alone scheme* (SA) as the scheme in which the master computes $\boldsymbol{AB}$ sololy from the input $\boldsymbol{A}, \boldsymbol{B}$.

Proposition 3.3: CTC of the SA system is $k_{\mathrm{A}}k_{\mathrm{B}}(2l-1)$.

Corollary 3.2 (Comparison (a) on the proposed scheme with that of the SA system): Under Assumption 3.3, if $n, n_{\mathrm{A}}, n_{\mathrm{B}}$ satisfies the below, the value of Eq.(4) is less than the CTC of the SA system.

$$l > \frac{1}{2k_{\mathrm{B}}(k_{\mathrm{A}} - (mn/n_{\mathrm{A}}n_{\mathrm{B}}))}(k_{\mathrm{A}}k_{\mathrm{B}} - (mn/n_{\mathrm{A}}n_{\mathrm{B}}) + D).$$

Example 3.1: Set $n = n_{\mathrm{A}}, m = n_{\mathrm{B}}$, $q = 2$, $k_{\mathrm{A}} = 100$, $k_{\mathrm{B}} = 293$ and $l = 100000$. The value of Eq.(4) is less than the CTC of the SA system when $n(> k_{\mathrm{A}} = 100)$ satisfies $101 \leq n \leq 172$. Table 4.1 of [7] shows that $(q, m) = (2, 293)$ satisfy Assumption 3.3.

IV. Conclusion and Future Works

We proposed a new distributed coded computation scheme called GDCC and, as one example, GDCCG. Moreover, we evaluated two trade-off important criterion of DCCs, (a) its computation time complexity and (b) its error-correcting capability of the overall system. Specially, we showed the parameter condition in which GDCCG is supreme to SA scheme in the evaluation on (a). Moreover, we showed error matrices which GDCCG can correct while previous schemes cannot correct in the evaluation on (b). In future works, we propose new DCCs taking into account of communication load and the number of workers.

References

[1] K. Kazama, A. Kamatsuka, T. Yoshida, and T Matsushima. A coded computation method based on a gabidulin code and the evaluation of its error-correcting capability (in japanese). *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences A*, Vol. J104-A, No. 6, pp. 156–159, 2021.

[2] E. M. Gabidulin. Theory of codes with maximum rank distance. *Problems of Information Transmission (English translation of Problemy Peredachi Informatsii)*, Vol. 21, No. 1, pp. 1–12, 1985.

[3] M. Gadouleau and Zhiyuan Yan. Complexity of decoding gabidulin codes. In *2008 42nd Annual Conference on Information Sciences and Systems*, pp. 1081–1085, 2008.

[4] S. Gao, D. Panario, V. Shoup, et al. Algorithms for exponentiation in finite fields. *Journal of Symbolic Computation*, Vol. 29, No. 6, pp. 879–889, 2000.

[5] R. B. Venturelli and D. Silva. An evaluation of erasure decoding algorithms for gabidulin codes. In *2014 International Telecommunications Symposium (ITS)*, pp. 1–5, 2014.

[6] D. Silva and F. R. Kschischang. Fast encoding and decoding of gabidulin codes. In *2009 IEEE International Symposium on Information Theory (ISIT)*, pp. 2858–2862, 2009.

[7] S. Gao. *Normal bases over finite fields.* University of Waterloo Waterloo, 1993.

Appendix

Gabidulin 符号の復号アルゴリズム [2] の概要と，その有限体上四則演算の回数を，[3] の考え方に基づいて説明する．受信語 $\boldsymbol{y} \in \mathbb{F}_{q^m}^n$ を入力として，$\hat{\boldsymbol{c}} \in \mathbb{F}_{q^m}^n$ を出力とするアルゴリズムは以下の Step1,...,7 から構成される．以下では t を限界ランク距離 $\lfloor (n-k_{\mathrm{A}})/2 \rfloor$，$d$ を最小ランク距離 $n - k_{\mathrm{A}} + 1 (\geq 2)$ と定義する．[3] との差異は後述するが，特に Step 3 が大きく異なる．

1) シンドローム $(s_1, \ldots, s_{d-1})^\top := \boldsymbol{H}^\top \boldsymbol{y} \in \mathbb{F}_{q^m}^{d-1}$ を計算する．
   このとき，$\mathbb{F}_{q^m}$ 上加算は $(n-1)(d-1)$ 回，$\mathbb{F}_{q^m}$ 上乗算が $n(d-1)$ 回必要である．
2) 線形化多項式 $S(x) := \sum_{i \in [n-k]} s_i x^{q^{i-1}}$ に対し，拡張ユークリッド互除法を用いて，$\deg_q F(x) < t$, $F(x) = \Lambda(x) * S(x) \mod z^{q^{d-1}}$ を満たす線形化多項式 $F(x), \Lambda(x)$ を計算する．
   計算時間の上界については，$\mathbb{F}_{q^m}$ 上加算が $(d-1)(d-2) - \frac{1}{2}t(t-1) + 2t(d-t-2)$ 回，$\mathbb{F}_{q^m}$ 上乗算が $(d-1)(d-2) - \frac{1}{2}t(t-5) + 2(d-t-1)(d-t-2)$ 回，$\mathbb{F}_{q^m}$ 上逆元を求める演算[1]が t 回必要である．詳細は後述する．
3) $\Lambda(x)$ の t 個の根 $E_1, \ldots, E_t \in \mathbb{F}_{q^m}$ と値を計算する[2]．
   計算時間については，$\mathbb{F}_q$ 上加算，減算，乗算，逆元を求める演算の合計が $\frac{2}{3}(m(m-1)(2m-1) - t(t-1)(2t-1)) - (t-2)(m-t) - (t-1)(m(m-1) - t(t-1)) + m^2 + m - 1$ 回，$\mathbb{F}_{q^m}$ 上加算が tm 回，$\mathbb{F}_{q^m}$ 上乗算が $(t+1)m$ 回である．詳細は後述する．
4) 各 $w \in [r]$ に対して $E_1 x_1^w + \cdots + E_t x_t^w = s_w$ を満たす $x_1, \ldots, x_t \in \mathbb{F}_{q^m}$ を計算する．
   計算時間については，$\mathbb{F}_{q^m}$ 上加算が $\frac{3}{2}t(t-1)$ 回，$\mathbb{F}_{q^m}$ 上乗算が $(t+1)\left(\frac{3}{2}t - 1\right)$ 回，$\mathbb{F}_{q^m}$ 上逆元を求める演算が t 回である．導出は [3] と同様なので省略する．
5) 各 $w \in [0, t-1]$ に対して $x_w = Y_{w1}h_1 + \cdots + Y_{wn}h_n$ を満たす $\boldsymbol{Y} \in \mathbb{F}_q^{t \times n}$ を計算する．
   計算時間は，$\{h_1, \ldots, h_n\}$ が正規基底 $\{v_1, \ldots, v_n\}$ に等しいとすると，演算回数 0 である．
6) $\boldsymbol{e} = \boldsymbol{Y}(E_1, \ldots, E_t)^\top \in \mathbb{F}_{q^m}^n$ を計算する．
   計算時間については，$\boldsymbol{Y}$ は $\mathbb{F}_q$ 上 $n \times t$ 行列であり，$(E_1, \ldots, E_t)^\top \in \mathbb{F}_{q^m}^t$ も $\mathbb{F}_q$ 上 $t \times m$ 行列に対応させられることを考えると，$\mathbb{F}_q$ 上加算，減算，乗算，逆元を求める演算の合計が $mn(2t-1)$ 回である．
7) $\boldsymbol{x} = \boldsymbol{y} - \boldsymbol{e} \in \mathbb{F}_{q^m}^n$ を計算する．
   計算時間は，$\mathbb{F}_q$ 上減算が mn 回である．

本論文と [3] では，以下のような $\mathbb{F}_q$ 上四則演算回数の差異が現れる．Step 2 において，線形化多項式の積の各係数を計算するときの演算回数に違いが現れることにより，本論文で算出した合計演算回数と，[3] で算出した回数が異なる．また，Step 6 においては，本論文では，mnr 回の $\mathbb{F}_q$ 上乗算と「$mn(t-1)$ 回」の $\mathbb{F}_q$ 上加算により計算できると算出したが，[3] では mnt 回の $\mathbb{F}_q$ 上乗算と「mnt 回」の $\mathbb{F}_q$ 上加算により計算できると算出した．さらに，Step 3 の計算回数の導出においては，確率的アルゴリズムと確定的アルゴリズムを扱う点で計算量評価が全く異なる．

A. *Step 2* の詳細

Step 2 は以下の 1,2 によって行われる．以下の 1,2 を以降では Step2-1,2-2 と称する．入力はシンドローム線形化多項式 $S(x) \in \mathbb{F}_{q^m}[x]$，出力は $F(x), \Lambda(x) \in \mathbb{F}_{q^m}[x]$ である．

1) $F_0(x) = x^{q^{d-1}}$，$F_1(x) = S(x)$ とおく．$\mathbb{F}_{q^m}$ 上線形化多項式 $G_1(x), G_2(x), \ldots$ および $F_0(x), F_1(x), \ldots$ を再

[1] a から a^{-1} を求める演算を指す．除算 ab^{-1} は b^{-1} を求めてから乗算 ab^{-1} を求める演算である．

[2] t 個未満しか解がない場合もあるが，ここでは簡単のため t 個にする

帰的に以下のように定義する．各 $i = 0, 1, \ldots$ に対し，

$$\begin{cases} F_i(x) = G_{i+1}(x) * F_{i+1}(x) + F_{i+2}(x) \\ \deg_q F_{i+2}(x) < \deg_q F_{i+1}(x) < \deg_q F_i(x). \end{cases} \quad (5)$$

以降，$\deg_q F_{i+1}(x) < \frac{d-1}{2} \leq \deg_q F_i(x)$ を満たす正整数 i を r とおく．

2) $A_{-1}(x) = 0$，$F_0(x) = x$ とおく．$\mathbb{F}_{q^m}$ 上線形化多項式 $A_1(x), A_2(x), \ldots$ を再帰的に以下のように定義する．各 $i = 1, 2, \ldots, r$ に対し，

$$A_i(x) = G_i(x) * A_{i-1}(x) + A_{i-2}(x) \quad (6)$$

と定義し，さらに，$\Lambda(x)$ を $A_r(x)$ をその最高次係数で割ったモニック線形化多項式とおく．

この Step2-1,2-2 の計算時間について説明する．

Proposition A.1: 変数が x の $\mathbb{F}_{q^m}$ 上線形化多項式 $F(x) = \sum_{i\in[0,u]} f_i x^{q^i}$，$G(x) = \sum_{i\in[0,v]} g_i x^{q^i}$，$f_u g_v \neq 0$ の積は，$\mathbb{F}_{q^m}$ 上加算 uv 回，$\mathbb{F}_{q^m}$ 上乗算 $(u+1)(v+1)$ 回を行うことで計算できる．

[3] と本論文ではこの回数の算出が異なっている．[3] では，加算 $uv - u - v - 1$ 回，乗算 uv 回として算出している．

Proof : 積の回数を算出する．$f_j g_k^{q^k}$，$i \in [0, u], k \in [0, v]$ の計算には，積が $(u+1)(v+1)$ 回必要である．ただし，仮定 3.2 から，g_k が与えられたもとでの $g_k^{q^k}$ の計算時間は無視するものとした．和の回数は，積の回数の総和から積線形多項式の q-次数を引けばよい．□

Proposition A.2: $\mathbb{F}_{q^m}$ 上の線形化多項式 $F(x)$，$G(x)$ の q-次数をそれぞれ u, v とおく．ここで，$u \leq v$ とする．$F(x)$ を $G(x)$ で割った時の商線形化多項式と剰余線形化多項式の組，すなわち $F(x) = Q(x) * G(x) + R(x)$ を満たす組 $(Q(x), R(x))$ を求めるには，最大で $\mathbb{F}_{q^m}$ 上の加算が $v(u - v + 1)$ 回，乗算が $(v+1)(u-v+1)$ 回，逆元を求める演算が 1 回必要である．

今回は証明は省略．この結果は [3] と同じである．

Proposition A.3: 上記のアルゴリズムにおいて，$r \leq t$ および $\deg_q \Lambda(x) \leq t$ が成立する．

Proof : 簡単のため，$\deg_q F_i(x)$ を d_i と表す．各 $i = 0, 1, \ldots, r$ に対して，$d_{i+1} < d_i$ および $d_i < d - 1 - i$ が成立する．よって，

$$(d_{r+1} <) \frac{d-1}{2} \leq d_r \leq d - 1 - r. \quad (7)$$

このことから，$r \leq t$ が示される．

式 (5) から，各 $i \in [r]$ に対して $\deg_q G_i(x) = d_{i-1} - d_i$ が成立することが示される．一方，式 (6) から，各 $i \in [r]$ に対して $\deg_q A_i(x) = d_{i-1} - d_i + \deg_q A_{i-1}(x)$ が成立することが示される．よって，$\deg_q A_i(x)$ が $d_0 - d_i$，すなわち $d - 1 - d_i$ であることがわかる．よって，

$$\deg_q \Lambda(x) = \deg_q A_r(x) = d - 1 - d_r \leq d - 1 - \frac{d-1}{2} = \frac{d-1}{2}. \quad (8)$$

このことから，$\deg_q \Lambda(x) \leq t$ が示される．□

Step 2-1 は，F_{q^m} 上の逆元を求める演算が t 回，乗算が高々 $2t(d-1) - \frac{1}{2}t(t-1)$ 回，加算が高々 $t(d-1)$ 回必要あれば求められる．以下で理由を説明する．命題 A.2 から，各 $i = 0, 1, \ldots, r-1$ において，$F_i(x), F_{i+1}(x)$ から $G_{i+1}(x), F_{i+2}(x)$ を求めるには，F_{q^m} 上の逆元を求める演算が 1 回，乗算が $(d_{i+1}+1)(d_i - d_{i+1} + 1)$ 回，加算が $d_{i+1}(d_i - d_{i+1} + 1)$ 回必要である．これを $i = 0, \ldots, r-1$ まで行うことにより，$F_0(x)$ から $F_{r+1}(x)$ まで，および $G_1(x)$ から $G_r(x)$ までを求めるには，F_{q^m} 上の逆元を求める演算が $r(\leq t)$ 回，乗算が高々 $(d-2)(d-1) - \frac{1}{2}t(t-5)$ 回，加算が高々 $(d-2)(d-1) - \frac{1}{2}t(t-1)$ 回あれば良いことがわかる．乗算の回数の上界については，以下の通り求められる．

$$\sum_{i=0}^{r-1}(d_{i+1}+1)(d_i - d_{i+1} + 1) \quad (9)$$

$$= r + \sum_{i=0}^{r-1} d_{i+1}(d_i - d_{i+1}) + \sum_{i=0}^{r-1} d_i \quad (10)$$

$$\leq r + d_1 \sum_{i=0}^{r-1}(d_i - d_{i+1}) + \sum_{i=0}^{r-1}(d - 1 - i) \quad (11)$$

$$\leq r + (d-2)(d-1-d_r) + rd - \frac{1}{2}r(r+1) \quad (12)$$

$$= (d-2)(d-1) - \frac{1}{2}r(r-5) + (d-2)(r - d_r) \quad (13)$$

$$\leq (d-2)(d-1) - \frac{1}{2}r(r-5) + (d-2)(r - \frac{d-1}{2}) \quad (14)$$

$$\leq (d-2)(d-1) - \frac{1}{2}t(t-5) + (d-2)(t - \frac{d-1}{2}) \quad (15)$$

$$\leq (d-2)(d-1) - \frac{1}{2}t(t-5). \quad (16)$$

加算については次のように算出できる．

$$\sum_{i=0}^{r-1} d_{i+1}(d_i - d_{i+1} + 1) \quad (17)$$

$$= -(d_0 - d_r + r) + \sum_{i=0}^{r-1}(d_{i+1}+1)(d_i - d_{i+1} + 1) \quad (18)$$

$$\leq -2r + (d-2)(d-1) - \frac{1}{2}t(t-5). \quad (19)$$

Step 2-2 は，F_{q^m} 上乗算が高々 $\frac{(d-1)(d+2t-5)}{4}$ 回，加算が高々 $\frac{(d-3)(d+2t-5)}{4}$ 回あれば求められる．以下で理由を説明する．命題 A.1 から，各 $i = 1, \ldots, r$ において，$A_{i-2}(x), A_{i-1}(x), G_i(x)$ から $A_i(x)$ を求めるには，F_{q^m} 上乗算が

$$(1 + \deg_q G_i(x))(1 + \deg_q A_{i-1}(x)) = (d_{i-1} - d_i + 1)(d - d_{i-1}) \quad (20)$$

回，加算が

$$\deg_q G_i(x) \deg_q A_{i-1}(x) + (1 + \deg_q A_{i-2}(x)) \quad (21)$$

$$= (d_{i-1} - d_i)(d - 1 - d_{i-1}) + (d - d_{i-2}) \quad (22)$$

回必要である．ところで，$i = 1$ においては，$A_i(x) = G_1(x)$ なので，計算は不要である．そこで，この計算を $i = 2, \ldots, r$ で行うことを考える．$A_r(x)$ までを求めるには，F_{q^m} 上乗算が高々 $2(d-t-1)(d-t-2)$ 回，加算が高々 $2t(d-t-2)$

回あれば良いことがわかる．$(t \geq)r \geq 2$ の場合，乗算の回数の上界については，以下の通り求められる．

$$\sum_{i=2}^{r}(d_{i-1}-d_i+1)(d-d_{i-1}) \leq \sum_{i=2}^{r}(d_{i-1}-d_i+1)(d-d_{r-1}) \tag{23}$$

$$= (d_1-d_r+r-1)(d-d_{r-1}) \tag{24}$$

$$\leq (d-2-\frac{d-1}{2}+r-1)(d-d_{r-1}) \tag{25}$$

$$\leq \frac{d+2t-5}{2}\frac{d-1}{2} \tag{26}$$

$$\leq (2d-2t-4)\frac{d-1}{2} \tag{27}$$

$$\leq 2(d-t-2)(d-t-1). \tag{28}$$

式 (28) の値は，$r=1$ のときの演算回数 (=0) の上界でもある．加算の回数の上界も同様にして求められる．$(t \geq)r \geq 2$ とする．このとき，$d \geq 5$ である．

$$\sum_{i=2}^{r}(d_{i-1}-d_i)(d-1-d_{i-1})+(d-d_{i-2}) \tag{29}$$

$$\leq \sum_{i=2}^{r}(d_{i-1}-d_i)(d-1-d_{i-1})+(d-1-d_{i-1}) \tag{30}$$

$$\leq (d-1-d_{r-1})\sum_{i=2}^{r}(d_{i-1}-d_i+1) \tag{31}$$

$$\leq (d-2-d_r)(d_1-d_r+r-1) \tag{32}$$

$$\leq (d-2-\frac{d-1}{2})(d-2-\frac{d-1}{2}+t-1) \tag{33}$$

$$\leq \frac{d-3}{2}\frac{d+2t-5}{2} \tag{34}$$

$$\leq \frac{d-3}{2}(2d-2t-4) \tag{35}$$

$$\leq 2t(d-t-2) \tag{36}$$

式 (36) の値は，$r=1$ のときの演算回数 (=0) の上界でもある．よって，Step 2 においては，上記で示した演算回数の上界が求められる．

B. Step 3 の詳細

Step 3 は以下の 1,2 によって行われる．以下の 1,2 を以降では Step 3-1,3-2 と称する．入力は線形化多項式 $\Lambda(x)$ であり，出力は t 以下の自然数 r および $\mathbb{F}_q$ 上線形独立な r 個の根 $E_1,\ldots,E_r \in \mathbb{F}_{q^m}$ である．ただし，$\Lambda(x)$ の根全体の集合がなす $\mathbb{F}_q$ 上線形空間の次元を r とおく．

1) $\Lambda(v_1),\ldots,\Lambda(v_m)$ を求める．
   $\mathbb{F}_{q^m}$ 上乗算が $(t+1)m$ 回，$\mathbb{F}_{q^m}$ 上乗算が tm 回である．
2) $\Lambda \in \mathbb{F}_q^{m\times m}$ を，

$$\boldsymbol{f}^m\left((\Lambda(v_1),\ldots,\Lambda(v_m))^\top\right) \in \mathbb{F}_{q^m}^m \tag{37}$$

と定義する．$\boldsymbol{U}\Lambda=\boldsymbol{0}$ が成立するフルランク行列 $\boldsymbol{U} \in \mathbb{F}_q^{r\times m}$ を $\mathbb{F}_q$ 上ガウス消去法で求め，各 $i \in [r]$ に対し $E_i = u_{i1}v_1+\cdots+u_{im}v_m$ とおく．

$\mathbb{F}_q$ 上四則演算回数の合計の上界は，合計 $m^2+m-1+\frac{2}{3}(m(m-1)(2m-1)-t(t-1)(2t-1))-(t-2)(m-t)-(t-1)(m^2-m-t^2+t)$ 回である．

このとき，$r=m-\mathrm{rank}\Lambda$ が成立する．

以下で $\mathbb{F}_q$ 上四則演算回数の上界を導出する．Step 3 は，線形化多項式 $\Lambda(x) \in \mathbb{F}_{q^m}[x]$ を入力したら，$\mathbb{F}_q$ 上線形独立な $r(\geq t)$ 個の根 $E_1,\ldots,E_r \in \mathbb{F}_{q^m}$ を出力するアルゴリズム．

Proposition A.4: 正整数 $r \in [t]$ を取る．各 $i \in [r]$ に対し，E_i を（$v_i=(v_1)^{q^{i-1}}$ を満たす）$\mathbb{F}_{q^m}$ の $\mathbb{F}_q$ 上正規基底 $\{v_1,\ldots,v_m\} \subset \mathbb{F}_{q^m}$ の $\mathbb{F}_q$ 上線形結合で表す，すなわち $E_i = u_{i1}v_1+\cdots+u_{im}v_m$，$u_{i1},\ldots,u_{im} \in \mathbb{F}_q$．$E_1,\ldots,E_r \in \mathbb{F}_{q^m}$ が線形化多項式 $\Lambda(x) \in \mathbb{F}_{q^m}[x]$ の根であり，$\mathbb{F}_q$ 上一次独立である必要十分条件は，$\boldsymbol{U}\Lambda=\boldsymbol{0}$ かつ $\mathrm{rank}(\boldsymbol{U})=r$ が成立することである．

Proof : $E_1,\ldots,E_r \in \mathbb{F}_{q^m}$ が $\mathbb{F}_q$ 上線形独立である必要十分条件は，u_{ij} を並べてできる $\mathbb{F}_q$ 上 $r\times m$ 行列 $\boldsymbol{U}$ がランク $\min\{m,r\}=r$ の行列であることである．実際，以下の式で，$\boldsymbol{a}=(a_1,\ldots,a_r)$ とおくと，

$$\forall \boldsymbol{a} \in \mathbb{F}_q^r, \sum_{i\in[t]} a_iE_i = 0 \Rightarrow \boldsymbol{a}=\boldsymbol{0} \tag{38}$$

$$\Leftrightarrow \forall \boldsymbol{a} \in \mathbb{F}_q^r, \sum_{i\in[t]} a_i \sum_{j\in[m]} u_{ij}v_j = 0 \Rightarrow \boldsymbol{a}=\boldsymbol{0} \tag{39}$$

$$\Leftrightarrow \forall \boldsymbol{a} \in \mathbb{F}_q^r, \sum_{j\in[m]} \left(\sum_{i\in[t]} a_iu_{ij}\right) v_j = 0 \Rightarrow \boldsymbol{a}=\boldsymbol{0} \tag{40}$$

$$\Leftrightarrow \forall \boldsymbol{a} \in \mathbb{F}_q^r, \forall j \in [m], \sum_{i\in[t]} a_iu_{ij} = 0 \Rightarrow \boldsymbol{a}=\boldsymbol{0} \tag{41}$$

$$\Leftrightarrow \forall \boldsymbol{a} \in \mathbb{F}_q^r, \forall j \in [m], \sum_{i\in[t]} a_iu_{ij} = 0 \Rightarrow \boldsymbol{a}=\boldsymbol{0} \tag{42}$$

$$\Leftrightarrow \forall \boldsymbol{a} \in \mathbb{F}_q^r, \boldsymbol{U}\boldsymbol{a}=\boldsymbol{0} \Rightarrow \boldsymbol{a}=\boldsymbol{0} \tag{43}$$

$$\Leftrightarrow \mathrm{rank}(\boldsymbol{U}) = \min\{m,r\} = r. \tag{44}$$

$E_1,\ldots,E_r$ が線形化多項式 $\Lambda(x)$ の根である必要十分条件は，$\boldsymbol{U}\Lambda=\boldsymbol{0}$ が成立することである．

$$\forall i \in [r], \Lambda(E_i)=0 \Leftrightarrow \forall i \in [r], \Lambda(u_{i1}v_1+\cdots+u_{im}v_m)=0 \tag{45}$$

$$\Leftrightarrow \forall i \in [r], u_{i1}\Lambda(v_1)+\cdots+u_{im}\Lambda(v_m)=0 \tag{46}$$

$$\Leftrightarrow \begin{pmatrix} u_{11} & \ldots & u_{1m} \\ \vdots & \ddots & \vdots \\ u_{r1} & \ldots & u_{rm} \end{pmatrix} \begin{pmatrix} \Lambda(v_1) \\ \vdots \\ \Lambda(v_m) \end{pmatrix} = \boldsymbol{0}(\in \mathbb{F}_{q^m}^r) \tag{47}$$

$$\Leftrightarrow \boldsymbol{U}\Lambda\boldsymbol{v}=\boldsymbol{0}(\in \mathbb{F}_{q^m}^r) \tag{48}$$

$$\Leftrightarrow \boldsymbol{U}\Lambda=\boldsymbol{0}(\in \mathbb{F}_q^{r\times m}). \tag{49}$$

以上のことから，$E_1,\ldots,E_r$ が線形化多項式 $\Lambda(x)$ の根であり $\mathbb{F}_q$ 上一次独立である必要十分条件は，$\boldsymbol{U}\Lambda=\boldsymbol{0}$ かつ $\mathrm{rank}(\boldsymbol{U})=r$ が成立することである．□

このことから，線形化多項式 $\Lambda(x)$ が入力されたら，$(\Lambda(v_1),\ldots,\Lambda(v_m))^\top \in \mathbb{F}_{q^m}^m$ を求めた後に，上記の条件を

満たす $\boldsymbol{U} \in \mathbb{F}_q^{r\times m}$ を $\mathbb{F}_q$ 上ガウス消去法で求め，各 $i \in [r]$ に対し，$E_i = u_{i1}v_1 + \cdots + u_{im}v_m$ を出力すればよい．

行列 $\boldsymbol{U}$ は *Algorithm 1* によって求められる．ただし簡単のため次の仮定を置いた．行列 Λ の第 (i,j) 成分を λ_{ij} とおく．記述の簡便さのため，Λ に対し *Algorithm 1*，すなわち列基本変形による簡約化を行うと以下のような行列の形になるとする[3]．

$$\Lambda' = \begin{pmatrix} \boldsymbol{I}_{(m-r)\times(m-r)} & \boldsymbol{0}_{(m-r)\times r} \\ \Lambda'_{[m-r+1,m],[m-r]} & \boldsymbol{0}_{r\times r} \end{pmatrix}. \tag{50}$$

ただし，$\boldsymbol{I}_{(m-r)\times(m-r)}$ は $m-r$ 次単位行列である．また，$\Lambda'_{[m-r+1,m],[m-r]} \in \mathbb{F}_q^{r\times(m-r)}$ は何らかの行列である．このとき，行列 $\boldsymbol{U}$ を $\left(\Lambda'_{[m-r+1,m],[m-r]}, \boldsymbol{I}_{r\times r}\right)$ とすればよい．

Algorithm 1 Step 3

Require: Λ
Ensure: $\boldsymbol{U}$
1: **for** $k \in [m-r]$ **do**
2: $\quad\lambda_{kk}^{-1}$ を計算
3: $\quad$**for** $j \in [k+1, m]$ **do**
4: $\quad\quad\lambda_{kj}\lambda_{kk}^{-1}$ を計算
5: $\quad\quad$**for** $i \in [k+1, m]$ **do**
6: $\quad\quad\quad\lambda_{ij} - \left(\lambda_{kj}\lambda_{kk}^{-1}\right)\lambda_{ik}$ を計算
7: $\quad\quad$**end for**
8: $\quad$**end for**
9: **end for**
10: $\boldsymbol{U} \coloneqq \left(\Lambda_{[m-r+1,m],[m-r]}, \boldsymbol{I}_{r\times r}\right)$

最初に，Step 3-1 の演算回数を説明する．各 $i' \in [m]$ に対し，$\Lambda(x) = \sum_{j\in[0,t]} \lambda_j x^{q^j} \in \mathbb{F}_{q^m}[x]$ が入力された下で $\Lambda(v_{i'})$ を出力するために，$\mathbb{F}_{q^m}$ 上乗算を $t+1$ 回，加算を t 回行えば求められることを示す．

$$\Lambda(v_{i'}) = \sum_{j\in[0,t]} \lambda_j v_{i'}^{q^j} = \sum_{j\in[0,t]} \lambda_j v_1^{q^{i'+j-1}} = \sum_{j\in[0,t]} \lambda_j v_{i'+j}. \tag{51}$$

正規基底の元 $v_{i'}, \ldots, v_{i'+t}$ の値はすでに求められているため，$\sum_{j\in[0,t]} \lambda_j v_{i'+j}$ の値は $\mathbb{F}_{q^m}$ 上乗算を $t+1$ 回，加算を t 回行えば求められる．

次に，Step 3-2 の演算回数を説明する．*Algorithm 3* の 1 行目から 9 行目にかけての $\mathbb{F}_q$ 上四則演算回数は，以下の通りである．

$$\sum_{k=1}^{m-r}\left(1 + \sum_{j=k+1}^{m}\left(1 + \sum_{i=k+1}^{m} 2\right)\right) \tag{52}$$

$$= \frac{1}{3}(m(m-1)(2m-1) - r(r-1)(2r-1)) + (m-r) + \frac{1}{2}(m(m-1) - r(r-1)). \tag{53}$$

よって，四則演算回数の合計の上界は，以下の通りである．

$$\frac{1}{3}(m(m-1)(2m-1) - r(r-1)(2r-1)) + (m-r) + \frac{1}{2}(m(m-1) - r(r-1)) \tag{54}$$

$$\leq \frac{2}{3}(m(m-1)(2m-1) - r(r-1)(2r-1)) + (2-r)(m-r) + (1-r)(m(m-1) - r(r-1)) \tag{55}$$

$$\leq \frac{2}{3}(m(m-1)(2m-1) - t(t-1)(2t-1)) + (2-t)(m-t) + (1-t)(m(m-1) - t(t-1)). \tag{56}$$

[3]必要であれば，行番号の置換によって式 (50) の形に変形したあと，それ以降の操作を行い $\boldsymbol{U}$ に対応する行列を求め，最後に行番号の逆置換を行った行列を改めて $\boldsymbol{U}$ と置けばよい．