SIAM J. COMPUT.
Vol. 49, No. 2, pp. 284–317

ROBUST POSITIONING PATTERNS WITH LOW REDUNDANCY*

YEOW MENG CHEE†, DUC TU DAO‡, HAN MAO KIAH‡, SAN LING‡, AND HENGJIA WEI§

Abstract. A robust positioning pattern is a large array that allows a mobile device to locate its position by reading a possibly corrupted small window around it. In this paper, we provide constructions of binary positioning patterns, equipped with efficient locating algorithms, that are robust to a constant number of errors and have redundancy within a constant factor of optimality. Furthermore, we modify our constructions to correct rank errors and obtain binary positioning patterns robust to any errors of rank less than a constant number. Additionally, we construct q-ary robust positioning sequences robust to a large number of errors, some of which have length attaining the upper bound. Our construction of binary positioning sequences that are robust to a constant number of errors has the least known redundancy among those explicit constructions with efficient locating algorithms. On the other hand, for binary robust positioning arrays, our construction is the first explicit construction whose redundancy is within a constant factor of optimality. The locating algorithms accompanying both constructions run in time cubic in sequence length or array dimension.

Key words. robust positioning patterns, Gray codes, Reed–Solomon codes, maximum rank distance codes

AMS subject classifications. 05B30, 94C30

DOI. 10.1137/19M1253472

1. Introduction.

Consider the problem of determining the *global* position of a mobile device in a wide environment by simply sensing a small *local* area around the device. This problem is fundamental in robotics and has practical applications in areas such as robot localization [17], camera localization [19], three-dimensional surface imaging by structured light [7], projected touchscreens [4], and smart styli [18].

A classic solution is via the use of positioning patterns. A *positioning pattern* is a large array of dimension $N_1 \times N_2$, in which all contiguous subarrays of dimension $n_1 \times n_2$ are distinct from each other. The dimension $n_1 \times n_2$ is called the *strength* of the positioning pattern. In the special case where $N_1 = n_1 = 1$, we refer to the one-dimensional positioning pattern as a *positioning sequence*. In practical applications, the positioning pattern is embedded in the wide area, and the mobile device reads a small *window* of the pattern, i.e., a subword of length n or a subarray of dimension $n_1 \times n_2$. Then due to the uniqueness of the window's subpattern, we are able to infer the position of the device. Positioning patterns have been extensively studied [13, 10, 15, 14, 5] and classical examples include de Bruijn sequences, m-sequences, perfect maps (also known as de Bruijn tori), and pseudorandom arrays.

*Received by the editors April 1, 2019; accepted for publication (in revised form) January 27, 2020; published electronically March 23, 2020. Part of results in this paper were presented at *Proceedings of SODA* 2019.

https://doi.org/10.1137/19M1253472

Funding: The research of the first, third, fourth, and fifth authors was supported in part by the Singapore Ministry of Education under grant MOE2015-T2-2-086. The research of the fourth author was also supported in part by the Nanyang Technological University grant M4080456.

†Department of Industrial Systems Engineering and Management, National University of Singapore, Singapore 117576 (pvocym@nus.edu.sg).

‡School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore 637371 (daoductu001@ntu.edu.sg, hmkiah@ntu.edu.sg, lingsan@ntu.edu.sg).

§Department of Electrical and Computer Engineering, Ben-Gurion University of the Negev, Beer Sheva 84105, Israel (hjwei05@gmail.com).

In reality, physical devices are prone to error and we want to locate a device even when we read a small window erroneously. To this end, we study a class of positioning patterns, called *robust positioning patterns*, where the subpatterns in distinct windows are far apart from each other. In other words, the subpatterns in all windows of a robust positioning pattern form an error-correcting code.

The study on robust positioning focuses on arrays or sequences in the Hamming metric and history can be traced back to the work of Kumar and Wei [10] on the minimum distance of partial periods of an m-sequence. Recently, Berkowitz and Kopparty [2] presented explicit constructions of robust positioning patterns, along with *efficient* locating algorithms. In particular, Berkowitz and Kopparty constructed high-rate q-ary robust positioning patterns (both one- and two-dimensional patterns) that locate a position even if a constant *fraction* of entries in a window are erroneous. In the regime where a constant *number* of errors are present in a window, the authors provided constructions with redundancy within a constant factor of optimality when the alphabet size is sufficiently large. When $q = 2$, the authors provided one-dimensional positioning patterns robust to a constant number of errors, but the result relies on the existence of suitable Mersenne-like primes. Efficient positioning in binary two-dimensional patterns robust to a constant number of errors remains open.

In this paper, we study both binary positioning patterns and q-ary positioning sequences. For *binary* positioning patterns that are robust to a constant *number* of errors, without relying on any unproven conjectures, we provide constructions for both one- and two-dimensional patterns whose redundancies are within a constant factor of optimality (in fact, we reduce the constant factor in the case for the one-dimensional pattern). Along with these patterns, we propose efficient locating algorithms with complexity $O(n^3)$ or $O((n_1n_2)^3)$, where n or $n_1 \times n_2$ is the strength of the pattern. Our construction is based on d-auto-cyclic vectors, Reed–Solomon codes, and Gray codes and can be further modified to correct errors of *rank* less than a constant *number*. For *q-ary* positioning sequences, we modify Berkowitz and Kopparty's construction to produce sequences robust to larger fraction of errors. We also determine the maximum length of some robust positioning sequences when the distance is large enough.

2. Preliminaries and contributions. For integers i, j with $i < j$, let $[i, j]$ denote the set of integers $\{i, i + 1, i + 2, \ldots, j\}$. For an integer $N \geq 2$, let $[\![N]\!]$ denote the set $[0, N - 1]$. Let Σ be an alphabet with q symbols and we index an array of dimension $N_1 \times N_2$ using the set $[\![N_1]\!] \times [\![N_2]\!]$. In particular, for an array $\mathbb{A} = (a_{ij}) \in \Sigma^{N_1 \times N_2}$, we use $\mathbb{A}[i, i+n_1-1][j, j+n_2-1]$ to denote the $n_1 \times n_2$ cyclical contiguous subarray of $\mathbb{A}$ whose top-left cell is a_{ij}; in the one-dimensional case, for a sequence $\boldsymbol{s} = s_0s_1s_2 \cdots s_{N-1} \in \Sigma^N$, we use $\boldsymbol{s}[i, i + n - 1]$ to denote the length-n cyclical contiguous subword of $\boldsymbol{s}$ starting at s_i.

Denote the Hamming weight of a matrix $\mathbb{V}$ by $w_H(\mathbb{V})$. For two matrices $\mathbb{V}$ and $\mathbb{W}$ of the same dimension $N_1 \times N_2$, let $\mathrm{agree}(\mathbb{V}, \mathbb{W})$ be the number of positions at which the corresponding entries are the same and $d_H(\mathbb{V}, \mathbb{W})$ be the Hamming distance between them. In other words, $\mathrm{agree}(\mathbb{V}, \mathbb{W}) + d_H(\mathbb{V}, \mathbb{W}) = N_1N_2$.

Without loss of generality, we assume that $n_1 \leq n_2$. For an $n_1 \times n_2$ window, define its *area* to be n_1n_2 and its *thickness* to be $(\log_q n_1)/(\log_q n_2)$.

A q-ary *robust positioning array* (RPA) of strength $n_1 \times n_2$ and distance d is an array $\mathbb{A}$ over Σ in which every pair of rectangular subarrays of dimension $n_1 \times n_2$ is of Hamming distance at least d apart. In other words, $d_H(\mathbb{A}[i, i + n_1 - 1][j, j + n_2 - 1], \mathbb{A}[i', i' + n_1 - 1][j', j' + n_2 - 1]) \geq d$ for all distinct $(i, j), (i', j') \in [\![N_1 - n_1 + 1]\!] \times [\![N_2 - n_2 + 1]\!]$. We denote such array as an $(n_1 \times n_2, d)_q$-RPA. For an $(n_1 \times n_2, d)_q$-

RPA of dimension $N_1 \times N_2$, define its *rate* to be $(\log_q N_1N_2)/(n_1n_2)$ and define its *redundancy* to be $n_1n_2 - \log_q(N_1N_2)$. Given q, n_1, n_2, and d, we are interested in the minimum redundancy of an $(n_1 \times n_2, d)_q$-RPA of dimension $N_1 \times N_2$ and denote this quantity by $\mathrm{red}_q(n_1 \times n_2, d)$. When $q = 2$, we suppress q in the notation.

Since all the subarrays of dimension $n_1 \times n_2$ in an $(n_1 \times n_2, d)_q$-RPA form an error-correcting code of size $(N_1 - n_1 + 1)(N_2 - n_2 + 1)$ with minimum distance d, we have the following bound on $\mathrm{red}_q(n_1 \times n_2, d)$.

PROPOSITION 2.1 (sphere-packing bound). *For all q, n_1, n_2, and d, we have that*

$$\mathrm{red}_q(n_1 \times n_2, d) \geq t \log_q(n_1n_2) + O(1),$$

where $t = \lfloor (d-1)/2 \rfloor$.

In the special case where $N_1 = n_1 = 1$, we refer to the one-dimensional q-ary robust positioning array of strength $1 \times n$ and distance d as *robust positioning sequence* (RPS) and denote it as $(n, d)_q$-RPS. The maximum length of an $(n, d)_q$-RPS is denoted by $P_q(n, d)$. So the minimum redundancy $\mathrm{red}_q(n, d) = n - \log_q P_q(n, d) \geq t \log_q n + O(1)$, where $t = \lfloor (d-1)/2 \rfloor$.

2.1. Previous work.

One-dimensional RPS. De Bruijn sequences and m-sequences are examples of positioning sequences. Decodable de Bruijn sequences can be found in Mitchell, Etzion, and Paterson [14]. In 1992, Kumar and Wei [10] studied m-sequences with error-correcting ability. Using random irreducible linear feedback shift register sequences, they showed the existence of a binary sequence of length $2^n - 1$ in which any pair of subwords of length approximately $n + d \log n$ has Hamming distance at least d (and at most $2d$) for $d \leq \sqrt{n}$. Notably, this shows the existence of a sequence that achieves the GV bound whenever $d \leq \sqrt{n}$. In 2008, Hagita et al. [9] presented constructions for almost optimal $(n, 3)$-RPSs. However their constructions are based on a conjecture on the existence of a certain type of primitive polynomials. In these constructions, no efficient locating algorithm was provided.

Recently, Berkowitz and Kopparty [2] presented explicit constructions of robust positioning sequences with efficient locating algorithms. For $q = n + 1$, they constructed positioning sequences of length $q^{n-3d-O(1)}$. For binary robust positioning sequences, they proposed an "augmented" code concatenation scheme and constructed a class of binary sequences with constant relative distance δ. They also studied binary sequences with constant distance d. However, their result relies on an open conjecture on the existence of suitable Mersenne-like primes. Furthermore, assuming the correctness of the conjecture, the redundancy of the (n, d)-RPS in their construction is at least $9d \log n$. More recently, Wang and co-authors studied the problem under a probabilistic noise model and provided efficient algorithms to locate the position with high probability.

Two-dimensional RPA. When $d = 1$, perfect maps and pseudorandom arrays have been studied extensively as the two-dimensional generalization of de Bruijn sequences and m-sequence [5, 15, 13]. For large values of d, Bruckstein et al. [3] constructed a class of binary RPAs that correctly finds the location provided less than a quarter of the bits in each row and less than half of the bits in each column are in error. Berkowitz and Kopparty [2] provided efficient constructions of high rate, constant relative distance RPAs over large q-ary alphabets. In the same paper, they mentioned that these q-ary arrays can be used to construct binary RPAs of high rate

and constant relative distance. They also remarked that their methods were unable to construct RPAs with optimal redundancy when the distance is constant.

2.2. Our contributions. We provide explicit constructions for binary RPSs and RPAs with efficient locating algorithms for fixed d. The locating algorithms run in time cubic in window length or window area, independent of the distance d. We also construct RPSs of high rate and asymptotically optimal RPSs when d is large. Our contributions are as follows.

(A) In section 3, we provide an explicit construction of (n, d)-RPSs with redundancy at most $3d \log n + 6.5 \log n + O(1)$, along with an efficient locating algorithm of complexity $O(n^3)$, for fixed d. This improves on Berkowitz and Kopparty's [2] construction that requires $9d \log n$ redundancy. Note that the sphere-packing bound suggests that the redundancy is $\lfloor \frac{d-1}{2} \rfloor \log n - O(1)$.

(B) Let $\mathbb{W}$ be a window of area A and thickness bounded by a constant. In section 4, we provide an explicit construction of binary RPAs for $\mathbb{W}$ with redundancy at most $4.21d \log A + 36.89 \log A + o(1)$, along with an efficient locating algorithm of complexity $O(A^3)$, for fixed distance d. This is the first infinite family of RPAs with efficient locating algorithms whose redundancy is within a constant factor of the optimality, i.e., $\lfloor \frac{d-1}{2} \rfloor \log A - O(1)$. In section 5, this construction is modified to produce positioning arrays which are robust to any errors of rank no more than a constant number.

(C) In section 6, we modify the construction of Berkowitz and Kopparty for $(n, \delta n)_q$-RPSs by doubling the size of the alphabet. The relative distance δ is improved from $\max\left\{\frac{1-R}{3}, 1-3R\right\}$ to $\max\left\{\frac{1-R}{2}, 1-2R\right\}$, where R is the rate. In contrast, the upper bound on the relative distance is $1 - R + o(1)$.

(D) We determine the exact value of $P(n, d)$ for $d \geq \lfloor 2n/3 \rfloor$ in section 7 and construct a class of asymptotically optimal $(n, n-1)_q$-RPS for $q = \Omega(n^{2+\epsilon})$ in section 8.

2.3. Our approach for fixed *d*. We describe the high-level ideas behind our construction of (n, d)-RPSs for fixed d. Following Berkowitz and Kopparty [2], we pick a q-ary code $\mathcal{C}$ whose block length corresponds to the window length and we concatenate the codewords of $\mathcal{C}$ in some ordering to obtain our RPS. Hence, whenever the window coincides with a possibly erroneous codeword, we simply leverage on the error-correcting capability of $\mathcal{C}$ to locate the window. The main challenge comes when the window does *not* coincide with a codeword. To overcome this, we borrow the following tools.

(i) *Gray codes.* We use Gray codes to order the codewords of $\mathcal{C}$ so that certain windows of the sequence are of high Hamming distance apart. In fact, this method was used by Berkowitz and Kopparty to construct q-ary RPSs of high rates.

(ii) *Markers.* To construct binary RPS, Berkowitz and Kopparty mapped the q-ary symbols of $\mathcal{C}$ to binary strings. Then they inserted short binary strings called markers into the binary sequence. These markers then allow one to locate the window's position relative to the codewords in $\mathcal{C}$. To further reduce redundancy, our construction utilizes a *d-auto-cyclic vector* as the marker.

We remark that d-auto-cyclic vectors were introduced by Levy and Yaakobi [11] in the context of DNA-based data storage. In the latter application, one objective is to design a set of primer sequences whose prefixes and suffixes satisfy a certain distance property (see Yazdi et al. [21] for more details). Not surprisingly, d-auto-cyclic vectors, which are useful in the primer sequence design, turn out to be a crucial ingredient of our construction.

3. Binary robust positioning sequences with constant distance. In this section, for fixed values of d, we propose an explicit construction for an (n, d)-RPS whose redundancy is $3d \log n + 6.5 \log n + O(1)$. As our construction is rather intricate, we first present the general ingredients required for constructing an RPS and later provide the specific parameters to achieve the desired redundancy.

First, we review Berkowitz and Kopparty's construction [2]. Let $\boldsymbol{v}^i$ denote the concatenation of i copies of the vector $\boldsymbol{v}$, and $\boldsymbol{vw}$ denote the concatenation of two vectors $\boldsymbol{v}$ and $\boldsymbol{w}$. Let $\mathcal{C}$ be an error-correcting code of length n and minimum distance d. Berkowitz and Kopparty picked *certain* words $\boldsymbol{s}_0, \boldsymbol{s}_1, \ldots, \boldsymbol{s}_{M-1}$ from $\mathcal{C}$ and concatenated them in some order to form a long sequence $\boldsymbol{S} = \boldsymbol{s}_0\boldsymbol{s}_1 \cdots \boldsymbol{s}_{M-1}$ of length $N = Mn$. Notice that from the choice of $\mathcal{C}$, we have that $d_H(\boldsymbol{s}_i, \boldsymbol{s}_j) \geq d$ for $0 \leq i < j \leq M-1$. In fact, via a careful choice of subwords and ordering, Berkowitz and Kopparty [2] are able to guarantee a certain distance property for *all* pairs of subwords in $\boldsymbol{S}$. We modify this technique to obtain a sequence with weaker property, where we guarantee the distance property for *some* pairs of subwords in $\boldsymbol{S}$. Formally, we have the following definitions.

DEFINITION 3.1. *Let $\boldsymbol{S}$ be a sequence. For two subwords $\boldsymbol{S}[i, i+n-1]$ and $\boldsymbol{S}[j, j+n-1]$ of length n in $\boldsymbol{S}$, we say that they start at the same modular position if $i \equiv j \pmod n$; otherwise, they start in different modular positions.*

A sequence $\boldsymbol{S}$ is called a q-ary modular robust positioning sequence of strength n and distance d, or $(n, d)_q$-MRPS for short, if

$$d_H(\boldsymbol{u}, \boldsymbol{v}) \geq d \text{ for } \boldsymbol{u}, \boldsymbol{v} \text{ in the same modular position.}$$

Next, to construct binary RPS, Berkowitz and Kopparty [2] used a short binary string called *marker* and a special mapping to transform symbols from a large alphabet to binary strings. Their construction then required at least $9d \log n$ bits of redundancy and is reliant on an open conjecture about Mersenne primes. To reduce the redundancy, we utilize another marker sequence and introduce the notion of d-auto-cyclic vectors.

DEFINITION 3.2 (Levy and Yaakobi [11]). *A vector $\boldsymbol{u} \in \Sigma^\ell$ is a d-auto-cyclic vector if*

$$d_H(\boldsymbol{u}, 0^i \boldsymbol{u}[0, \ell - i - 1]) \geq d$$

for all $1 \leq i \leq d$.

Levy and Yaakobi provided the following construction of d-auto-cyclic vectors.

PROPOSITION 3.3 (Levy and Yaakobi [11]). *Let $\ell = d\lceil \log d \rceil + 2d$. Set $\boldsymbol{u}$ to be the vector*

$$\begin{aligned} \boldsymbol{u} &= 1^d \boldsymbol{u}_0 \cdots \boldsymbol{u}_{\lceil \log d \rceil}, \text{ where} \\ \boldsymbol{u}_i &= ((1^{2^i} 0^{2^i})^d)[0, d-1]. \end{aligned} \tag{3.1}$$

Then $\boldsymbol{u}$ is a d-auto-cyclic vector.

Example 3.4. For $d = 3$, the sequence $\boldsymbol{u} = 111\ 101\ 110\ 111$ is a 3-auto-cyclic vector.

We also introduce the notion of window weight limited.

DEFINITION 3.5 (Levy and Yaakobi [11]). *Let N, k, d be positive integers such that $d < k < N$. We say a vector $\boldsymbol{v} \in \mathbb{F}_2^N$ satisfies the (d, k)-window weight limited (WWL) constraint and is called a (d, k)-WWL vector if $w_H(\boldsymbol{v}[i, i+k-1]) \geq d$ for any $0 \leq i \leq N - k$.*

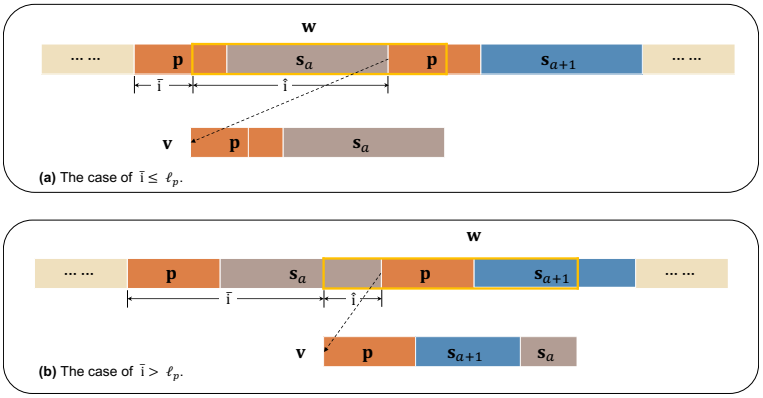

FIG. 1. *The vector* $\boldsymbol{v}$ *obtained by shifting* $\boldsymbol{w}$ *cyclically leftward* $\hat{\imath}$ *times.*

We are ready to present our construction.

Construction 1. Given n and d, choose k such that $\ell < k$ and $k + \ell < n$, where $\ell = d\lceil \log d \rceil + 2d$. Let $\boldsymbol{u}$ be a d-auto-cyclic vector of length ℓ (e.g., the vector from Proposition 3.3) and set $\boldsymbol{p} = 0^k\boldsymbol{u}$ to be a vector of length $\ell_p = k + \ell$. In addition, set $n' = n - \ell_p$. Our construction comprises the sequence $\boldsymbol{p}$ and a list of length-n' binary vectors $\boldsymbol{s}_0, \boldsymbol{s}_1, \ldots, \boldsymbol{s}_{M-1}$ satisfying the following conditions:

(P1) $\boldsymbol{s}_i$ is a (d,k)-WWL vector for $i \in [\![M]\!]$;
(P2) $\boldsymbol{s}_{i+1}[0, j-1]\boldsymbol{s}_i[j, n'-1]$ is a (d,k)-WWL vector for $i \in [\![M]\!]$ and $j \in [\![n'-1]\!]$; and
(P3) the concatenation $\boldsymbol{s}_0\boldsymbol{s}_1\boldsymbol{s}_2\cdots\boldsymbol{s}_{M-1}$ is an (n',d)-MRPS.

Set $\boldsymbol{S} \triangleq \boldsymbol{p}\boldsymbol{s}_0\boldsymbol{p}\boldsymbol{s}_1\boldsymbol{p}\boldsymbol{s}_2\cdots\boldsymbol{p}\boldsymbol{s}_{M-1}$.

In the next subsection, we specify the values of k and ℓ and provide an explicit method to construct $\boldsymbol{s}_i$'s. Consequently, we obtain the sequence $\boldsymbol{S}$ and show that it has the desired redundancy. Prior to this, we prove that $\boldsymbol{S}$ is indeed an (n,d)-RPS. Note that (P3) implies $\boldsymbol{S}$ is an (n,d)-MRPS. Hence, it remains to show that every two subwords in different modular positions have distance at least d. To do so, we have the following technical lemma.

LEMMA 3.6. *Consider the subword* $\boldsymbol{w} = \boldsymbol{S}[i_0, i_0 + n - 1]$ *in* $\boldsymbol{S}$. *Pick* $i \in [\![n]\!]$. *Then the following hold:*

(i) *If* $i + i_0 \equiv 0 \pmod{n}$, *then* $\boldsymbol{w}[i, i + \ell_p - 1] = \boldsymbol{p}$.
(ii) *If* $i + i_0 \not\equiv 0 \pmod{n}$, *then* $d_H(\boldsymbol{w}[i, i + \ell_p - 1], \boldsymbol{p}) \ge d$.

Proof. Let $\hat{\imath}$ be the unique integer of $[\![n]\!]$ such that $\hat{\imath} + i_0 \equiv 0 \pmod{n}$. We consider the vector $\boldsymbol{v}$, which is obtained by shifting $\boldsymbol{w}$ cyclically leftwards $\hat{\imath}$ times. Then it suffices to show that $\boldsymbol{v}[0, \ell_p - 1] = \boldsymbol{p}$ and $d_H(\boldsymbol{v}[i, i + \ell_p - 1], \boldsymbol{p}) \ge d$ for $i \in [1, n-1]$. Suppose that $i_0 = an + \bar{\imath}$, where $\bar{\imath} \in [n]$. From Construction 1 (see Figure 1), we have that

$$\boldsymbol{v} = \begin{cases} \boldsymbol{p}\boldsymbol{s}_a & \text{if } \bar{\imath} \le \ell_p; \\ \boldsymbol{p}\boldsymbol{s}_{a+1}[0, \bar{\imath} - \ell_p - 1]\boldsymbol{s}_a[\bar{\imath} - \ell_p, n' - 1] & \text{if } \bar{\imath} > \ell_p. \end{cases}$$

Hence $\boldsymbol{v}[0, \ell_p - 1] = \boldsymbol{p}$.

Now we consider $\boldsymbol{v}[i, i + \ell_p - 1]$ with $i \neq 0$. Since $\boldsymbol{s}_a$ and $\boldsymbol{s}_{a+1}$ satisfy the conditions (P1) and (P2), we can always assume that $\boldsymbol{v} = \boldsymbol{p}\boldsymbol{x}$ for some (k,d)-WWL vector $\boldsymbol{x}$ of length n'. We proceed by cases.

Case 1. $i \in [1, d]$. Then

$$\boldsymbol{v}[i+k-i, i+k-i+\ell-1] = \boldsymbol{v}[k, k+\ell-1] = \boldsymbol{u},$$
$$\boldsymbol{p}[k-i, k-i+\ell-1] = 0^i\boldsymbol{u}[0, \ell-i-1].$$

Since $\boldsymbol{u}$ is d-auto-cyclic, we have

$$\begin{aligned} d_H(\boldsymbol{v}[i, i+\ell_p-1], \boldsymbol{p}) &\geq d_H(\boldsymbol{v}[k, k+\ell-1], \boldsymbol{p}[k-i, k-i+\ell-1]) \\ &= d_H(\boldsymbol{u}, 0^i\boldsymbol{u}[0, \ell-i-1]) \geq d. \end{aligned}$$

Case 2. $i \in [d+1, \ell_p - d]$. Notice that $\boldsymbol{v} = 0^k\boldsymbol{u}\boldsymbol{x}$. Since $\ell < k$, the subword $\boldsymbol{v}[i, i+k-1]$ should contain either the length-d prefix of $\boldsymbol{u}$ or the length-d suffix of $\boldsymbol{u}$, both of which are 1^d. So the weight of $\boldsymbol{v}[i, i+k-1]$ is at least d. It follows that

$$d_H(\boldsymbol{v}[i, i+\ell_p-1], \boldsymbol{p}) \geq d_H(\boldsymbol{v}[i, i+k-1], \boldsymbol{p}[0, k-1]) = d_H(\boldsymbol{v}[i, i+k-1], 0^k) \geq d.$$

Case 3. $i \in [\ell_p - d + 1, n-k]$. The subword $\boldsymbol{v}[i, i+k-1]$ is contained in $1^d\boldsymbol{x}$. Since $\boldsymbol{x}$ is a (d, k)-WWL vector, the weight of $\boldsymbol{v}[i, i+k-1]$ is at least d. Again, we have

$$d_H(\boldsymbol{v}[i, i+\ell_p-1], \boldsymbol{p}) \geq d_H(\boldsymbol{v}[i, i+k-1], \boldsymbol{p}[0, k-1]) = d_H(\boldsymbol{v}[i, i+k-1], 0^k) \geq d.$$

Case 4. $i \in [n-k+1, n-d]$. Since $i+k-n \geq 1$ and $i+k+d-1-n \leq k-1$, we have $\boldsymbol{v}[i+k, i+k+d-1] = \boldsymbol{v}[i+k-n, i+k+d-1-n] = 0^d$. Note that $\boldsymbol{p}[k, k+d-1] = \boldsymbol{u}[0, d-1] = 1^d$. It follows that $d_H(\boldsymbol{v}[i, i+\ell_p-1], \boldsymbol{p}) \geq d$.

Case 5. $i \in [n-d+1, n-1]$. Let $\delta = n-i$, then $\delta \in [1, d-1]$. We have

$$\begin{aligned} \boldsymbol{v}[i+k, i+k+\ell-1] &= \boldsymbol{v}[n+k-\delta, n+k+\ell-\delta-1] \\ &= \boldsymbol{v}[k-\delta, k+\ell-\delta-1] = 0^\delta\boldsymbol{u}[0, \ell-\delta-1]. \end{aligned}$$

Since $\boldsymbol{p}[k, k+\ell-1] = \boldsymbol{u}$ and $\delta \in [1, d-1]$, we have

$$d_H(\boldsymbol{v}[i, i+\ell_p-1], \boldsymbol{p}) \geq d_H(\boldsymbol{v}[i+k, i+k+\ell-1], \boldsymbol{p}[k, k+\ell-1]) \geq d,$$

which completes the proof. ∎

Next, we prove that the construction is correct.

THEOREM 3.7. *Let $\boldsymbol{S}$ be the sequence constructed in Construction* 1. *Then $\boldsymbol{S}$ is an (n, d)-RPS.*

Proof. Let $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ be two distinct subwords of length n in $\boldsymbol{S}$. Assume that $\boldsymbol{w}_1 = \boldsymbol{S}[i, i+n-1]$ and $\boldsymbol{w}_2 = \boldsymbol{S}[j, j+n-1]$, where $i \neq j$. Since $\boldsymbol{s}_0\boldsymbol{s}_1\cdots\boldsymbol{s}_{M-1}$ is an (n', d)-MRPS, we have that $\boldsymbol{S}$ is an (n, d)-MRPS. Hence, $d_H(\boldsymbol{w}_1, \boldsymbol{w}_2) \geq d$ whenever $i \equiv j \pmod{n}$.

It remains to consider the case where $i \not\equiv j \pmod{n}$. Let $\hat{i}$ be the integer of $[n]$ such that $i + \hat{i} \equiv 0 \pmod{n}$. Hence, we have $j + \hat{i} \not\equiv 0 \pmod{n}$. Lemma 3.6 implies that $\boldsymbol{w}_1[\hat{i}, \hat{i}+\ell_p-1] = \boldsymbol{p}$ and $d_H(\boldsymbol{w}_2[\hat{i}, \hat{i}+\ell_p-1], \boldsymbol{p}) \geq d$. Hence, $d_H(\boldsymbol{w}_1, \boldsymbol{w}_2) \geq d_H(\boldsymbol{w}_1[\hat{i}, \hat{i}+\ell_p-1], \boldsymbol{w}_2[\hat{i}, \hat{i}+\ell_p-1]) \geq d$. ∎

3.1. Sequence construction. Given n and d, we provide the choice of k and ℓ and construct the vectors $\boldsymbol{u}, \boldsymbol{s}_0, \boldsymbol{s}_1, \ldots, \boldsymbol{s}_{M-1}$ to satisfy conditions (P1), (P2), and (P3).

First, we set ℓ and $\boldsymbol{u}$ as in (3.1). Next, set $m \triangleq (3/2)\log n$, $k = 3m$ and $q \triangleq 2^m$. Let $\alpha_0, \alpha_1, \ldots, \alpha_{q-1}$ be the q distinct elements in $\mathbb{F}_q$ and let ϕ be an arbitrary bijection from $\mathbb{F}_q$ to $\mathbb{F}_2^m$. Set $X \triangleq \{\alpha_j \in \mathbb{F}_q : w_H(\phi(\alpha_j)) \geq d\}$. Let $r = \sum_{i=d}^{m} \binom{m}{i}$, which is the cardinality of X. Recall that $n' = n-(k+\ell)$ and our objective is to construct an (n', d)-MRPS. To this end, we require the concepts of Gray codes and Reed–Solomon codes.

DEFINITION 3.8 (Gray codes). *Let Σ be an alphabet with q symbols and $\mathcal{G} = (\boldsymbol{\sigma}_0, \boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_{q^n-1})$ be a sequence of all the vectors in Σ^n. Then $\mathcal{G}$ is called an (n, q)-*Gray code *if any two adjacent vectors $\boldsymbol{\sigma}_i$ and $\boldsymbol{\sigma}_{i+1}$ in $\mathcal{G}$ differ in only one position.*

THEOREM 3.9 (decoding for Gray codes [8]). *Let q and n be two positive integers. There exists an (n, q)-Gray code $\mathcal{G} = (\boldsymbol{\sigma}_0, \boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_{q^n-1})$ and a decoding function $\mathrm{dec}_{\mathrm{Gray}} : \mathbb{F}_q^n \to [\![q^n]\!]$ such that $\mathrm{dec}_{\mathrm{Gray}}(\boldsymbol{\sigma}_i) = i$ for all $i \in [\![q^n]\!]$. Furthermore, $\mathrm{dec}_{\mathrm{Gray}}$ can be computed in $O(n \log^2 q)$ time.*

THEOREM 3.10 (Reed–Solomon code [20]). *Let q be a prime power. Suppose that $k_R < n_R \leq q$. Then there exists a linear code $\mathcal{C}_{\mathrm{RS}}$ of length n_R, dimension k_R, and minimum distance $d_R \triangleq n_R - k_R + 1$. Furthermore, there exist encoding function $\mathrm{enc}_{\mathrm{RS}}^{(n_R,k_R)} : \mathbb{F}_q^{k_R} \to \mathcal{C}_{\mathrm{RS}}$ and decoding function $\mathrm{dec}_{\mathrm{RS}}^{(n_R,k_R)} : \mathbb{F}_q^{n_R} \to \mathcal{C}_{\mathrm{RS}}$ such that the following hold:*

(i) *For all $\boldsymbol{\sigma} \in \mathbb{F}_q^{k_R}$, the k_R-prefix of $\mathrm{enc}_{\mathrm{RS}}^{(n_R,k_R)}(\boldsymbol{\sigma})$ is $\boldsymbol{\sigma}$. In other words,*

$$\mathrm{enc}_{\mathrm{RS}}^{(n_R,k_R)}(\boldsymbol{\sigma})[0, k_R - 1] = \boldsymbol{\sigma}.$$

(ii) *Choose $\boldsymbol{c} \in \mathcal{C}_{\mathrm{RS}}$ and suppose that $d_H(\bar{\boldsymbol{c}}, \boldsymbol{c}) \leq (d_R-1)/2$. Then $\mathrm{dec}_{\mathrm{RS}}^{(n_R,k_R)}(\bar{\boldsymbol{c}}) = \boldsymbol{c}$. Furthermore, $\mathrm{dec}_{\mathrm{RS}}^{(n_R,k_R)}$ can be computed in $O(n^3)$ time.*

In their construction for q-ary RPSs, Berkowitz and Kopparty [2] used a Gray code to give an ordering to a subset of codewords in a Reed–Solomon code and concatenated these codewords in this ordering to form the desired sequence. In our construction, we adapt the technique to obtain a family of q-ary vectors $\boldsymbol{c}_i$ such that $\boldsymbol{c}_0\boldsymbol{c}_1 \cdots \boldsymbol{c}_{M-1}$ is a q-ary MRPS. Then we apply the mapping ϕ to each $\boldsymbol{c}_i$ and append short sequences 1^d in proper positions to obtain the WWL vector $\boldsymbol{s}_i$.

Specifically, set $n_R \triangleq (n' - (2d+2)d)/m$ and $k_R \triangleq n_R - 2d - 2$. Consider the Reed–Solomon code $\mathcal{C}_{\mathrm{RS}}$ from Theorem 3.10. Now we provide our construction of $\boldsymbol{s}_i$ for $i \in [\![r^{k_R}]\!]$, and consequently, the sequence $\boldsymbol{S}$.

Construction 1A. Let $M = r^{k_R}$ and $\mathcal{G} = (\boldsymbol{\sigma}_0, \boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_{M-1})$ be a (k_R, r)-Gray code over X. For $i \in [\![M]\!]$, set $\boldsymbol{c}_i = \mathrm{enc}_{\mathrm{RS}}^{(n_R,k_R)}(\boldsymbol{\sigma}_i)$. Then for each $\boldsymbol{c}_i$, construct a binary vector $\boldsymbol{s}_i$ as

$$\boldsymbol{s}_i = \phi(\boldsymbol{c}_i[0])\phi(\boldsymbol{c}_i[1]) \cdots \phi(\boldsymbol{c}_i[k_R-1])1^d\phi(\boldsymbol{c}_i[k_R])1^d\phi(\boldsymbol{c}_i[k_R+1]) \cdots 1^d\phi(\boldsymbol{c}_i[n_R-1]).$$

Finally, let $\boldsymbol{p} = 0^k\boldsymbol{u}$, where $\boldsymbol{u}$ is the d-auto-cyclic vector in (3.1). Construct the sequence $\boldsymbol{S}$ as

$$\boldsymbol{S} = \boldsymbol{p}\boldsymbol{s}_0\boldsymbol{p}\boldsymbol{s}_1 \cdots \boldsymbol{p}\boldsymbol{s}_{M-1}.$$

We summarize in Table 1 all the parameters and notation involved in our construction (see also Figure 2). To simplify our exposition and analysis, we assume that all parameters are integers.

Observe that each $\boldsymbol{c}_i$ is a codeword whose length-k_R prefix is $\boldsymbol{\sigma}_i$. Hence, when $j < k_R$, the symbol $\boldsymbol{c}_i[j]$ belongs to X and $w_H(\phi(\boldsymbol{c}_i[j])) \geq d$. However, when $j \geq k_R$, the symbol $\boldsymbol{c}_i[j]$ may not belong to X and the weight of $\phi(\boldsymbol{c}_i[j])$ may be less than

TABLE 1
Notation summary for Construction 1A.

Notation	Remark
n	the strength of the RPS
d	the distance of the RPS
ℓ	the length of the d-auto-cyclic vector
m	$m \triangleq \frac{3}{2}\log n$
k	$k \triangleq 3m$
ℓ_p	$\ell_p \triangleq k + \ell$
r	$r \triangleq \sum_{i=d}^{m} \binom{m}{i}$
n'	$n' \triangleq n - \ell_p$
q	$q \triangleq 2^m$
n_R	$n_R \triangleq \frac{1}{m}(n' - (2d+2)d)$
k_R	$k_R \triangleq n_R - 2d - 2$
M	$M \triangleq r^{k_R}$
$\mathbb{F}_q$	the finite field with q elements
α_j	the element in $\mathbb{F}_q$
ϕ	a one-to-one map from $\mathbb{F}_q$ to $\mathbb{F}_2^m$
X	the set of α_j such that $w_H(\phi(\alpha_j)) \geq d$
$\mathcal{G}$	a (k_R, r)-Gray code
$\boldsymbol{\sigma}_i$	the ith vector in $\mathcal{G}$
$\boldsymbol{c}_i$	the codeword in a q-ary Reed–Solomon code of length n_R and dimension k_R such that $\boldsymbol{c}_i[0, k_R - 1] = \boldsymbol{\sigma}_i$
$\boldsymbol{s}_i$	the concatenation of the binary vectors $\phi[\boldsymbol{c}_i[j]]$, $j \in [k_R]$, as well as the vectors $1^d\phi[\boldsymbol{c}_i[j]]$, $k_R \leq j \leq n_R - 1$
$\boldsymbol{u}$	the d-auto-cyclic vector of length ℓ
$\boldsymbol{p}$	$\boldsymbol{p} \triangleq 0^k\boldsymbol{u}$

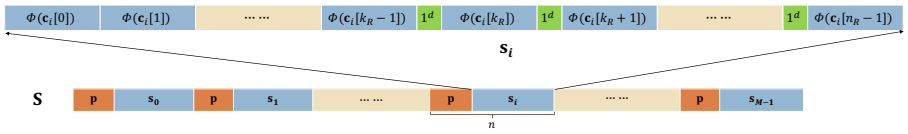

FIG. 2. *The sequence* $\boldsymbol{S}$ *in Construction* 1A.

d. So, we prepend a sequence 1^d at the head of $\phi(\boldsymbol{c}_i[j])$ for each $j \geq k_R$. Since $k = 3m \geq m + 2d$, it is easy to check that the vectors $\boldsymbol{s}_0, \boldsymbol{s}_1, \ldots, \boldsymbol{s}_{M-1}$ satisfy the conditions (P1) and (P2). For (P3), we have the following result on $\boldsymbol{S}$.

LEMMA 3.11. *The concatenation* $\boldsymbol{s}_0\boldsymbol{s}_1 \cdots \boldsymbol{s}_{M-1}$ *is an* (n', d)*-MRPS.*

Proof. Since each $\boldsymbol{s}_i$ is obtained by inserting the sequences 1^d at fixed positions in the concatenation of the binary strings $\phi[\boldsymbol{c}_i[j]]$, it suffices to show that the concatenation $\boldsymbol{c}_0\boldsymbol{c}_1 \cdots \boldsymbol{c}_{M-1}$ is an $(n_R, d+1)_q$-MRPS.

Assume that $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ start at position i and position j, respectively. Since $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ are in the same modular position, we may assume that $\bar{m} \equiv i \equiv j \pmod{n_R}$, where $\bar{m} \in [\![n_R]\!]$. Further let $i = an_R + \bar{m}$ and $j = bn_R + \bar{m}$. We proceed by cases.

Case 1. $\bar{m} = 0$. Then $\boldsymbol{w}_1 = \boldsymbol{c}_a$ and $\boldsymbol{w}_2 = \boldsymbol{c}_b$. Since $\boldsymbol{c}_a$ and $\boldsymbol{c}_b$ belong to $\mathcal{C}_{\mathrm{RS}}$, we have that $d_H(\boldsymbol{w}_1, \boldsymbol{w}_2) = d_H(\boldsymbol{c}_a, \boldsymbol{c}_b) \geq d_R = n_R - k_R + 1 > d + 1$.

Case 2. $\bar{m} \in [1, k_R]$. So, $\boldsymbol{w}_1 = \boldsymbol{c}_a[\bar{m}, n_R - 1]\boldsymbol{c}_{a+1}[0, \bar{m} - 1]$. Since $\mathcal{G}$ is a Gray code, $d_H(\boldsymbol{c}_a[0, \bar{m} - 1], \boldsymbol{c}_{a+1}[0, \bar{m} - 1]) = d_H(\boldsymbol{\sigma}_a, \boldsymbol{\sigma}_{a+1}) \leq 1$. In other words,

$$d_H(\boldsymbol{w}_1, \boldsymbol{c}_a[\bar{m}, n_R - 1]\boldsymbol{c}_a[0, \bar{m} - 1]) \leq 1.$$

Similarly, we have

$$d_H(\boldsymbol{w}_2, \boldsymbol{c}_b[\bar{m}, n_R - 1]\boldsymbol{c}_b[0, \bar{m} - 1]) \leq 1.$$

It follows that

$$d_H(\boldsymbol{w}_1, \boldsymbol{w}_2) \geq d_H(\boldsymbol{c}_a, \boldsymbol{c}_b) - 2 \geq n_R - k_R - 1 > d + 1.$$

Case 3. $\bar{m} \in [k_R + 1, n_r - 1]$. We partition the interval $[0, n_R - 1]$ into three pieces by setting

$$\begin{aligned} I_1 &= [0, n_R - \bar{m} - 1], \\ I_2 &= [n_R - \bar{m}, n_R - \bar{m} + k_R - 1], \text{ and} \\ I_3 &= [n_R - \bar{m} + k_R, n_R - 1]. \end{aligned}$$

For $j \in \{1, 2, 3\}$, let $\text{agree}_j = \text{agree}(\boldsymbol{w}_1[I_j], \boldsymbol{w}_2[I_j])$. Then

$$\begin{aligned} \text{agree}_2 + \text{agree}_3 &= \text{agree}(\boldsymbol{c}_{a+1}[0, \bar{m} - 1], \boldsymbol{c}_{b+1}[0, \bar{m} - 1]) \\ &\leq k_R - 1 \\ \text{agree}_1 + \text{agree}_3 &\leq |I_1| + |I_3| = n_R - |I_2| = n_R - k_R. \end{aligned}$$

Since $d_H(\boldsymbol{c}_a[0, k_R - 1]), \boldsymbol{c}_{a+1}[0, k_R - 1])) \leq 1$, and $d_H(\boldsymbol{c}_b[0, k_R - 1]), \boldsymbol{c}_{b+1}[0, k_R - 1])) \leq 1$, we have

$$\begin{aligned} \text{agree}_1 + \text{agree}_2 &= \text{agree}(\boldsymbol{c}_a[\bar{m}, n_R - 1]\boldsymbol{c}_{a+1}[0, k_R - 1], \boldsymbol{c}_b[\bar{m}, n_R - 1]\boldsymbol{c}_{b+1}[0, k_R - 1]) \\ &\leq \text{agree}(\boldsymbol{c}_a[\bar{m}, n_R - 1]\boldsymbol{c}_a[0, k_R - 1], \boldsymbol{c}_b[\bar{m}, n_R - 1]\boldsymbol{c}_b[0, k_R - 1]) + 2 \\ &\leq \text{agree}((\boldsymbol{c}_a, \boldsymbol{c}_b) + 2 \leq k_R + 1. \end{aligned}$$

Summing these three inequalities yields

$$\text{agree}_1 + \text{agree}_2 + \text{agree}_3 \leq \frac{n_R + k_R}{2}.$$

Therefore, we have

$$d_H(\boldsymbol{w}_1, \boldsymbol{w}_2) = n_R - \text{agree}(\boldsymbol{w}_1, \boldsymbol{w}_2) \geq \frac{n_R - k_R}{2} = d + 1,$$

which completes the proof. ∎

Therefore, Construction 1A yields an (n, d)-RPS as desired. We analyze the required redundancy in the following corollary.

COROLLARY 3.12. *For sufficient large n, there is an (n, d)-RPS with redundancy at most $3d \log n + 6.5 \log n + O(1)$.*

Proof. Recall that

$$\begin{aligned} r = \sum_{i=d}^{m} \binom{m}{i} &= 2^m \left[1 - \sum_{i=0}^{d-1} \binom{m}{i} \frac{1}{2^m}\right] \geq 2^m \left[1 - \exp\left(-\frac{2(m/2 - d + 1)^2}{m}\right)\right] \\ &\geq 2^m \left(1 - e^{-\frac{2m}{3 \log e}}\right) = 2^m \left(1 - \frac{1}{n}\right). \end{aligned}$$

The first inequality comes from Hoeffding's inequality for the tail of the binomial distribution while the second one holds as $(m/2-d+1)^2/m > m/(3\log e)$ when m is large enough. Then we have

$$\log r \geq m + \log\left(1-\frac{1}{n}\right) = m + \ln\left(1-\frac{1}{n}\right)\log e \geq m - \frac{\log e}{n-1}.$$

Note that $k_R = \frac{1}{m}[n-k-\ell-(2d+2)(m+d)] = \frac{n}{m} - 2d - 5 - O(\frac{1}{m})$. Hence the redundancy of $\boldsymbol{S}$ is

$$\begin{aligned} n - \log(nM) &= n - k_R \log r - \log n \\ &\leq n - \left(\frac{n}{m} - 2d - 5 - O\left(\frac{1}{m}\right)\right)\left(m - \frac{\log e}{n-1}\right) - \log n \\ &= 2dm + 5m - \log n + O(1) = \ 3d\log n + 6.5\log n + O(1). \end{aligned}$$ ∎

Remark 3.13. In Construction 1A, we convert the q-ary vectors into binary vectors by mapping the elements of $\mathbb{F}_q$ to the binary vectors of length m, and we append some short sequences 1^d so that the resulting sequences satisfy conditions (P1) and (P2). Alternatively, one may choose a prime power q that is at most $\sum_{i=d}^{m}\binom{m}{i}$ and map the elements in $\mathbb{F}_q$ to the binary length-m vectors with weight at least d. This approach then results in an (n,d)-RPS with redundancy $4.21d\log n + 9.53\log n + O(1)$, which is larger than that in Corollary 3.12. However, in next section, when we construct two-dimensional robust positioning arrays, we have to adopt this approach as it is difficult to tile some small patterns and large squares while keeping low redundancy.

3.2. Locating algorithm. We present a locating algorithm for the subwords of the sequence $\boldsymbol{S}$ in Construction 1A. In particular, the locating algorithm corrects up to $\lfloor (d-1)/2 \rfloor$ errors in $O(n^3)$ time, independent of parameter d.

Suppose that $\boldsymbol{w}$ is a subword of $\boldsymbol{S}$ that is corrupted at e positions with $e \leq (d-1)/2$. In other words, there is a unique index i such that $d_H(\boldsymbol{S}[i, i+n-1], \boldsymbol{w}) \leq (d-1)/2$ and our task is to recover i. Equivalently, if we write i as $an + \bar{i}$ with $\bar{i} \equiv i \pmod n$, then our task is to recover both a and $\bar{i}$. In what follows, we give a broad overview of the steps and the detailed implementation of the algorithm is provided in Algorithm 3.1.

(I) We determine $\bar{i}$. To do so, we determine the unique index $\hat{i}$ such that $d_H(\boldsymbol{w}[\hat{i}, \hat{i}+\ell_p-1], \boldsymbol{p}) \leq (d-1)/2$. Set $\bar{i} \in [\![n]\!]$ such that $\bar{i} + \hat{i} \equiv 0 \pmod n$

(II) Next, we cyclically rotate $\boldsymbol{w}$ leftward by $\hat{i}$ positions to obtain $\boldsymbol{v}$. Observe that $\boldsymbol{v}$ is the binary image obtained from either a q-ary codeword $\boldsymbol{c}_a$ or a concatenation $\boldsymbol{c}_a[n_R - j + 1, n_R]\boldsymbol{c}_{a+1}[0, j]$ for some $j \in [\![n_R]\!]$. Since $\boldsymbol{v}$ is obtained via the map ϕ and prepending the string $\boldsymbol{p}$ and inserting $n_R - k_R$ strings 1^d, we reverse this process to obtain the q-ary estimate $\bar{\boldsymbol{c}}$.

(III) Finally, depending on the value of $\bar{i}$, we apply the Reed–Solomon decoding algorithm $\mathrm{dec}_{\mathrm{RS}}$ to find either $\boldsymbol{c}_a$ or $\boldsymbol{c}_{a+1}$ or some shortened versions of these words. Therefore, we determine a and hence obtain $i = an + \hat{i}$.

THEOREM 3.14. *Suppose $\boldsymbol{w}$ is a corrupted subword of the sequence $\boldsymbol{S}$ with exactly e errors. If $2e < d$, then Algorithm* 3.1 *can determine the position of $\boldsymbol{w}$ in the sequence $\boldsymbol{S}$ in $O(n^3)$ time.*

Proof. Suppose the corrupted subword $\boldsymbol{w}$ starts at position $\alpha n + \beta$, where $\alpha \in [\![M]\!]$ and $\beta \in [\![n]\!]$. Denote the original subword $\boldsymbol{S}[\alpha n + \beta, (\alpha+1)n + \beta - 1]$ as $\boldsymbol{w}^\circ$, and so, $d_H(\boldsymbol{w}, \boldsymbol{w}^\circ) = e$. Lemma 3.6 implies that $d_H(\boldsymbol{w}[i, i+\ell_p-1], \boldsymbol{p}) \leq e$ when

Algorithm 3.1 Locating algorithm for the sequence $\boldsymbol{S}$ in Construction 1A.

Input: a sequence $\boldsymbol{w}$ of length n
Output: a position $i \triangleq an + \bar{i}$ such that $d_H(\boldsymbol{S}[i, i+n-1], \boldsymbol{w}) \le (d-1)/2$

$\hat{i} \leftarrow$ unique index such that $d_H(\boldsymbol{w}[\hat{i}, \hat{i}+\ell_p-1], \boldsymbol{p}) \le (d-1)/2$
Set $\bar{i} \in [\![n]\!]$ such that $\bar{i} + \hat{i} \equiv 0 \pmod n$

$\boldsymbol{v} \leftarrow$ the vector obtained by rotating $\boldsymbol{w}$ cyclically leftward $\hat{i}$ positions
$\hat{\boldsymbol{v}} \leftarrow$ the vector obtained from $\boldsymbol{v}$ by deleting $\boldsymbol{v}[0, \ell_p-1]$ and $\boldsymbol{v}[\ell_p + mk_R + (m+d)j, \ell_p + mk_R + (m+d)j + d - 1]$ for all $j \in [n_R - k_R]$
$\bar{\boldsymbol{c}} \leftarrow \phi^{-1}(\hat{\boldsymbol{v}}[0, m-1])\phi^{-1}(\hat{\boldsymbol{v}}[m, 2m-1]) \cdots \phi^{-1}(\hat{\boldsymbol{v}}[m(n_R-1), mn_R - 1])$

if $\bar{i} \in [0, \ell_p + mk_R - 1]$ **then**
  $\boldsymbol{c} \leftarrow \mathrm{dec}_{\mathrm{RS}}^{(n_R, k_R)}(\bar{\boldsymbol{c}})$
  $a \leftarrow \mathrm{dec}_{\mathrm{Gray}}(\boldsymbol{c}[0, k_R - 1])$
else if $\bar{i} \in [\ell_p + mk_R, \ell_p + mk_R + (d+1)(m+d) - 1]$ **then**
  $\bar{\boldsymbol{c}}^s \leftarrow$ the shortened codeword $\bar{\boldsymbol{c}}[0, k_R - 1]\bar{\boldsymbol{c}}[k_R + (d+1), n_R - 1]$
  $\boldsymbol{c}^s \leftarrow \mathrm{dec}_{\mathrm{RS}}^{(n_R-(d+1), k_R)}(\bar{\boldsymbol{c}}^s)$
  $a \leftarrow \mathrm{dec}_{\mathrm{Gray}}(\boldsymbol{c}^s[0, k_R - 1])$
else
  $\bar{\boldsymbol{c}}^s \leftarrow$ the shortened codeword $\bar{\boldsymbol{c}}[0, k_R + (d+1) - 1]$
  $\boldsymbol{c}^s \leftarrow \mathrm{dec}_{\mathrm{RS}}^{(n_R-(d+1), k_R)}(\bar{\boldsymbol{c}}^s)$
  $a + 1 \leftarrow \mathrm{dec}_{\mathrm{Gray}}(\boldsymbol{c}^s[0, k_R - 1])$
return $an + \bar{i}$

$i + \beta \equiv 0 \pmod n$, and $d_H(\boldsymbol{w}[i, i+\ell_p-1], \boldsymbol{p}) \ge d - e$ when $i + \beta \not\equiv 0 \pmod n$. Since $d - e > e$, the value β can be uniquely determined and we have $\bar{i} = \beta$. In order to determine α, we consider the following cases.

Case 1. $\bar{i} \in [\![\ell_p]\!]$. By shifting the original subword $\boldsymbol{w}^\circ$ leftward $\hat{i}$ times, we obtain $\boldsymbol{p}\boldsymbol{s}_\alpha$. Since shifting both $\boldsymbol{w}$ and $\boldsymbol{w}^\circ$ simultaneously does not increase the Hamming distance, we have $d_H(\boldsymbol{v}, \boldsymbol{p}\boldsymbol{s}_\alpha) = e$. After removing the sequences $\boldsymbol{p}$ and 1^d from $\boldsymbol{p}\boldsymbol{s}_\alpha$ and the corresponding subwords from $\boldsymbol{v}$, we have that $d_H(\hat{\boldsymbol{v}}, \phi(\boldsymbol{c}_\alpha[0])\phi(\boldsymbol{c}_\alpha[1]) \cdots \phi(\boldsymbol{c}_\alpha[n_R - 1])) \le e$. Next, we apply the inverse function ϕ^{-1} and observe that the Hamming distance does not increase. Therefore, $d_H(\bar{\boldsymbol{c}}, \boldsymbol{c}_\alpha) \le e < d/2$.

Since $d_R = 2d+2$, we have that $d_H(\bar{\boldsymbol{c}}, \boldsymbol{c}_\alpha) \le (d_R - 1)/2$, and applying the decoding algorithm $\mathrm{dec}_{\mathrm{RS}}^{(n_R, k_R)}$ on $\bar{\boldsymbol{c}}$ recovers $\boldsymbol{c}_\alpha$. Finally, we apply $\mathrm{dec}_{\mathrm{Gray}}$ to $\boldsymbol{c}_\alpha[0, k_R - 1]$ to recover α and hence the output a is indeed α.

Case 2. $\bar{i} \in [\ell_p, \ell_p + mk_R - 1]$. By shifting $\boldsymbol{w}^\circ$ leftward $\hat{i}$ positions, we obtain the sequence $\boldsymbol{p}\boldsymbol{s}_{\alpha+1}[0, \bar{i} - \ell_p - 1]\boldsymbol{s}_\alpha[\bar{i} - \ell_p, n' - 1]$. So, $d_H(\boldsymbol{v}, \boldsymbol{p}\boldsymbol{s}_{\alpha+1}[0, \bar{i} - \ell_p - 1]\boldsymbol{s}_\alpha[\bar{i} - \ell_p, n' - 1]) = e$.

Set $j = \lfloor (\bar{i} - \ell_p)/m \rfloor$, and so, $j < k_R$. Note that there may be a subword of length m which covers both the tail of $\boldsymbol{s}_{\alpha+1}[0, \bar{i} - \ell_p - 1]$ and the head of $\boldsymbol{s}_\alpha[\bar{i} - \ell_p, n' - 1]$. Hence, we have $d_H(\bar{\boldsymbol{c}}, \boldsymbol{c}_{\alpha+1}[0, j]\boldsymbol{c}_\alpha[j+1, n_R - 1]) \le e + 1$. It follows that

$$\begin{aligned} d_H(\bar{\boldsymbol{c}}, \boldsymbol{c}_\alpha) &= d_H(\bar{\boldsymbol{c}}, \boldsymbol{c}_\alpha[0, j]\boldsymbol{c}_\alpha[j+1, n_R - 1]) \\ &\le d_H(\bar{\boldsymbol{c}}, \boldsymbol{c}_{\alpha+1}[0, j]\boldsymbol{c}_\alpha[j+1, n_R - 1]) + 1 \le e + 2 \le (d_R - 1)/2. \end{aligned}$$

Similar to Case 1, we apply $\mathrm{dec}_{\mathrm{RS}}^{(n_R, k_R)}$ to $\bar{\boldsymbol{c}}$ to recover $\boldsymbol{c}_\alpha$ and then apply $\mathrm{dec}_{\mathrm{Gray}}$ to recover $a = \alpha$.

Case 3. $\bar{i} \in [\ell_p + mk_R, \ell_p + mk_R + (d+1)(m+d) - 1]$. Similar to Case 2, we have

$$d_H(\boldsymbol{v}, \boldsymbol{p}\boldsymbol{s}_{\alpha+1}[0, \bar{i} - \ell_p - 1]\boldsymbol{s}_\alpha[\bar{i} - \ell_p, n' - 1]) \leq e.$$

Due to the range of $\bar{i}$, $\boldsymbol{s}_{\alpha+1}[0, \bar{i} - \ell_p - 1]$ contains the subword $\boldsymbol{s}_{\alpha+1}[0, mk_R - 1]$ and $\boldsymbol{s}_\alpha[\bar{i} - \ell_p, n' - 1]$ contains the subword $\boldsymbol{s}_\alpha[mk_R + (d+1)(m+d), n' - 1]$. It follows that the distance between the shortened vector $\bar{\boldsymbol{c}}^s = \bar{\boldsymbol{c}}[0, k_R - 1]\bar{\boldsymbol{c}}[k_R + (d+1), n_R - 1]$ and the vector $\boldsymbol{c}_{\alpha+1}[0, k_R - 1]\boldsymbol{c}_\alpha[k_R + (d+1), n_R - 1]$ is no more than e. Since $d_H(\boldsymbol{c}_\alpha[0, k_R - 1], \boldsymbol{c}_{\alpha+1}[0, k_R - 1]) \leq 1$, we have

$$d_H(\bar{\boldsymbol{c}}^s, \boldsymbol{c}_\alpha[0, k_R - 1]\boldsymbol{c}_\alpha[k_R + (d+1), n_R - 1]) \leq e + 1.$$

The shortened vector $\boldsymbol{c}_\alpha[0, k_R - 1]\boldsymbol{c}_\alpha[k_R + (d+1), n_R - 1]$ can be treated as a codeword of a Reed–Solomon code of length $n_R - (d+1)$ and dimension k_R. Since $e + 1 \leq (n_R - (d+1) - k_R)/2$, we apply the decoding algorithm $\mathrm{dec}_{\mathrm{RS}}^{(n_R-(d+1),k_R)}$ to recover $\boldsymbol{c}_\alpha[0, k_R - 1]$. As before, we apply $\mathrm{dec}_{\mathrm{Gray}}$ to recover α.

Case 4. $\bar{i} \in [\ell_p + mk_R + (d+1)(m+d), n - 1]$. In this case, we consider the shortened vector $\bar{\boldsymbol{c}}^s = \bar{\boldsymbol{c}}[0, k_R + (d+1) - 1]$. Similar to Case 3, we have

$$d_H(\bar{\boldsymbol{c}}^s, \boldsymbol{c}_{\alpha+1}[0, k_R + (d+1) - 1]) \leq e.$$

As before, we can $\mathrm{dec}_{\mathrm{RS}}^{(n_R-(d+1),k_R)}$ to recover $\boldsymbol{c}_{\alpha+1}[0, k_R - 1]$ and hence recover $\alpha + 1$.

We analyze the running time. To determine $\hat{i}$, we require $\ell_p n$ comparisons. Next, the Reed–Solomon decoding $\mathrm{dec}_{\mathrm{RS}}^{(n_R,k_R)}$ runs in $O(n_R^3) = O(n^3)$ time. Finally, the decoding of Gray codes $\mathrm{dec}_{\mathrm{Gray}}$ runs in $O(k_R \log^2 q) = O(nm^2) = O(n \log^2 n)$ time. Therefore, Algorithm 3.1 computes the location in $O(n^3)$ time. ∎

4. Binary robust positioning arrays with constant distance. Let $\mathbb{W}$ be an $n_1 \times n_2$ window of area A and thickness bounded by a constant. We generalize Construction 1A to produce binary RPAs for $\mathbb{W}$ with constant distance d. To this end, we require the following number theoretic result.

LEMMA 4.1 (Baker, Harman, and Pintz [1]). *Let $\theta = 0.525$. There exists x_0 such that for every $x \geq x_0$, the interval $[x - x^\theta, x]$ contains a prime.*

Fix d. Set $m = \frac{\log A}{1-\theta}$ and $r = \sum_{i=d}^{m} \binom{m}{i}$, where $\theta = 0.525$. Lemma 4.1 then provides a prime q such that $r - r^\theta \leq q \leq r$. Take an arbitrary injective map ψ from $\mathbb{F}_q$ to $\mathbb{F}_2^m$ such that $w_H(\psi(x)) \geq d$ for all $x \in \mathbb{F}_q$. In other words, ψ maps symbols in $\mathbb{F}_q$ to binary sequences of length m with weight at least d. For a vector $\boldsymbol{x} = x_0 x_1 \cdots x_{n-1} \in \mathbb{F}_q^n$, let $\psi(\boldsymbol{x}) = \psi(x_0)\psi(x_1)\cdots\psi(x_{n-1})$.

Suppose that n_2 is divisible by m. Set $n_R \triangleq n_1(n_2/m) - 4$ and $k_R \triangleq n_R - 2(d+7)$. Then we have $k_R < n_R < q$ and set $M \triangleq q^{k_R/2}$. Now we provide our construction of robust positioning arrays.

Construction 2. Let $\mathcal{G} = (\boldsymbol{\sigma}_0, \boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_{M-1})$ be a $(k_R/2, q)$-Gray code and consider a Reed–Solomon code of length n_R and dimension k_R over $\mathbb{F}_q$. For each $0 \leq i, j \leq M - 1$, set $\boldsymbol{c}_{ij} = \mathrm{enc}_{\mathrm{RS}}^{(n_R,k_R)}(\boldsymbol{\sigma}_i \boldsymbol{\sigma}_j)$.

Let $\boldsymbol{p} = 0^k \boldsymbol{u}$, where $k = 4m - \ell$ and $\boldsymbol{u}$ is the d-auto-cyclic vector provided in (3.1). For each $\boldsymbol{c}_{ij}$, the concatenation $\boldsymbol{p}\psi(\boldsymbol{c}_{ij})$ has length $n_1 n_2$ as $n_R = n_1(n_2/m) - 4$. Then let $\mathbb{A}_{ij}$ be the $n_1 \times n_2$ array whose rows can be concatenated to form $\boldsymbol{p}\psi(\boldsymbol{c}_{ij})$ (see Figure 3).

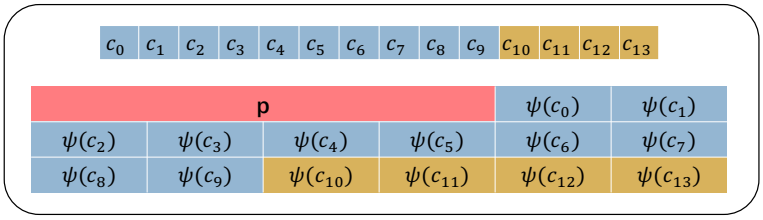

FIG. 3. *A codeword from a Reed–Solomon code of length* 14 *and dimension* 10 *and its corresponding* $n_1 \times n_2$ *array with* $n_1 = 3$ *and* $n_2 = 6m$. *The blue cells represent the message bits and the yellow cells represent the check bits.*

Finally, construct a large array $\mathbb{A}$ as

$$\mathbb{A} = \begin{pmatrix} \mathbb{A}_{00} & \mathbb{A}_{01} & \cdots & \mathbb{A}_{0,M-1} \\ \mathbb{A}_{10} & \mathbb{A}_{11} & \cdots & \mathbb{A}_{1,M-1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbb{A}_{M-1,0} & \mathbb{A}_{M-1,1} & \cdots & \mathbb{A}_{M-1,M-1} \end{pmatrix}.$$

For each $\mathbb{A}_{ij}$, we refer to the zeros and ones in $\psi(\boldsymbol{c}_{ij}[\ell])$ with $\ell < k_R$ as message bits and refer to those in $\psi(\boldsymbol{c}_{ij}[\ell])$ with $\ell \geq k_R$ as check bits (see Figure 3).

For an array $\mathbb{M} = (m_{i,j})$, we use $\mathbb{M}[i_0, i_0 + a - 1][j_0, j_0 + b - 1]$ to denote the $a \times b$ cyclical subarray of $\mathbb{M}$ whose top-left cell is $m_{i_0 j_0}$. The following result is an analogue to Lemma 3.6 and helps to locate the modular position efficiently.

LEMMA 4.2. *Consider the subarray* $\mathbb{W} = \mathbb{A}[i_0, i_0 + n_1 - 1][j_0, j_0 + n_2 - 1]$ *in* $\mathbb{A}$. *Pick* $i \in [\![n_1]\!]$ *and* $j \in [\![n_2]\!]$. *Then the following hold:*

(i) *If* $i + i_0 \equiv 0 \pmod{n_1}$ *and* $j + j_0 \equiv 0 \pmod{n_2}$, *then* $\mathbb{W}[i, i][j, j + 4m - 1] = \boldsymbol{p}$.

(ii) *If* $i + i_0 \not\equiv 0 \pmod{n_1}$ *or* $j + j_0 \not\equiv 0 \pmod{n_2}$, *then* $d_H(\mathbb{W}[i, i][j, j + 4m - 1], \boldsymbol{p}) \geq d$.

Proof. Let $\hat{i} \in [n_1]$ and $\hat{j} \in [n_2]$ such that $\hat{i} + i_0 \equiv 0 \pmod{n_1}$ and $\hat{j} + j_0 \equiv 0 \pmod{n_2}$. We consider the array $\mathbb{V}$, which is obtained by shifting $\mathbb{W}$ cyclically upward $\hat{i}$ times and leftward $\hat{j}$ times. Then $\mathbb{V}[0, 0][0, 4m - 1] = \boldsymbol{p}$ (see Figure 4) and it suffices to show $d_H(\mathbb{V}[i, i][j, j + 4m - 1], \boldsymbol{p}) \geq d$ when $i \in [1, n_1 - 1]$ or $j \in [1, n_2 - 1]$.

For $i \in [1, n_1 - 1]$, since $k = 4m - \ell > 3m$, $\mathbb{V}[i, i][j, j + k - 1]$ must contain a length-m vector $\psi(x_0)$ for some $x_0 \in \mathbb{F}_q$ (see Figure 4). Observe that $\psi(x_0)$ has weight at least d. Hence, we have

$$\begin{aligned} d_H(\mathbb{V}[i, i][j, j + 4m - 1], \boldsymbol{p}) &\geq d_H(\mathbb{V}[i, i][j, j + k - 1], \boldsymbol{p}[0, k - 1]) \\ &= d_H(\mathbb{V}[i, i][j, j + k - 1], 0^k) \geq d. \end{aligned}$$

For $i = 0$ and $j \in [1, n_2 - 1]$, the proof follows from Lemma 3.6. ∎

We regard an array of dimension $a \times (bm)$ as a $a \times b$ partitioned matrix with each block being a vector of length m. Given a pair of $a \times (bm)$ arrays $\mathbb{M}_1$ and $\mathbb{M}_2$, we denote the Hamming distance of their corresponding partitioned matrices as $d_B(\mathbb{M}_1, \mathbb{M}_2)$. In other words, $d_B(\mathbb{M}_1, \mathbb{M}_2)$ counts the number of different blocks in $\mathbb{M}_1$ and $\mathbb{M}_2$. Therefore, in Construction 2, we have

$$d_H(\mathbb{A}_{ij}, \mathbb{A}_{i'j'}) \geq d_B(\mathbb{A}_{ij}, \mathbb{A}_{i'j'}) = d_H(\boldsymbol{c}_{ij}, \boldsymbol{c}_{i'j'}).$$

For a pair of $a \times b$ arrays $\mathbb{M}_1$ and $\mathbb{M}_2$ with b not divisible by m, we can repeat the last columns to form two arrays $\mathbb{M}_1'$ and $\mathbb{M}_2'$ of dimension $a \times (\lceil b/m \rceil m)$. Then de-

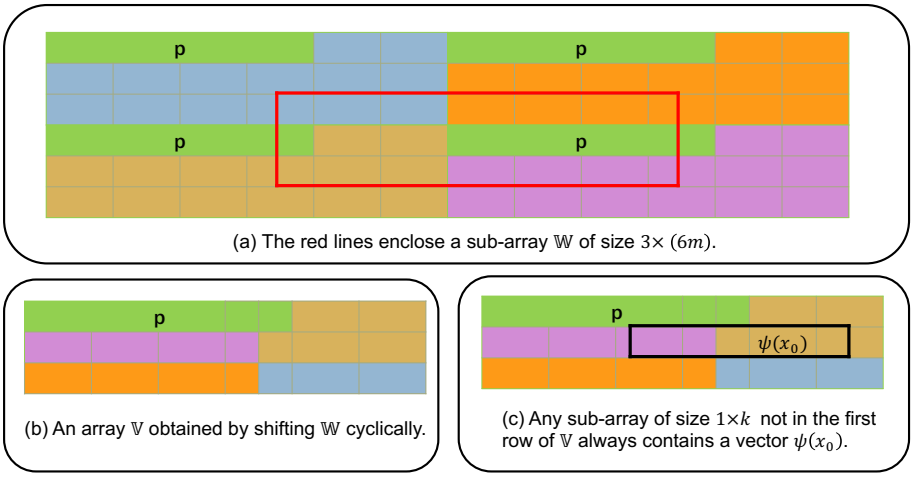

FIG. 4. *An example with* $n_1 = 3$ *and* $n_2 = 6m$ *to illustrate the proof of Lemma* 4.2. (a) *The red lines enclose the subarray* $\mathbb{W} = \mathbb{A}[2, 5][3.5m, 9.5m - 1]$. (b) *Shifting* $\mathbb{W}$ *upward one time and leftward* $2.5m$ *times, we got the array* $\mathbb{V}$. (c) *Any subarray* $\mathbb{V}[i, i][j, j + k - 1]$ *with* $i \neq 0$ *always contains a vector* $\psi(x_0)$ *for some* $x_0 \in \mathbb{F}_q$.

note $d_B(\mathbb{M}_1, \mathbb{M}_2) := d_B(\mathbb{M}_1', \mathbb{M}_2')$. Hence, $d_B(\mathbb{M}_1, \mathbb{M}_2)$ counts the number of different (truncated) blocks in $\mathbb{M}_1$ and $\mathbb{M}_2$.

LEMMA 4.3. *For any two subarrays* $\mathbb{W} = \mathbb{A}[i, i + n_1 - 1][j, j + n_2 - 1]$ *and* $\mathbb{W}' = \mathbb{A}[i', i'+n_1-1][j', j'+n_2-1]$ *with* $i \equiv i' \pmod{n_1}$ *and* $j \equiv j' \pmod{n_2}$, *the Hamming distance between them is at least* d.

Proof. Suppose that $i = an + \bar{i}$ and $i' = a'n + \bar{i}$ for some $\bar{i} \in [\![n_1]\!]$, and $j = bn + \bar{j}$ and $j' = b'n + \bar{j}$ for some $\bar{j} \in [\![n_2]\!]$. Let $\hat{i} \in [\![n_1]\!]$ and $\hat{j} \in [\![n_2]\!]$ be the integers such that $\bar{i} + \hat{i} \equiv 0 \pmod{n_1}$ and $\bar{j} + \hat{j} \equiv 0 \pmod{n_2}$. Shift $\mathbb{W}$ cyclically upward $\hat{i}$ times and leftward $\hat{j}$ times and denote the resulting array as $\mathbb{V}$. Similarly, let $\mathbb{V}'$ be the corresponding shifted array of $\mathbb{W}'$. Then $d_H(\mathbb{W}, \mathbb{W}') = d_H(\mathbb{V}, \mathbb{V}')$. Since thickness is bounded by a constant, we have $\log n_1 / \log n_2 = O(1)$, and then $(n_R - k_R)m < n_2$. It follows that the check bits of $\mathbb{A}_{\alpha\beta}$ appear in the last row $\mathbb{A}_{\alpha\beta}[n_1 - 1, n_1 - 1][n_2 - (n_R - k_R)m, n_2 - 1]$. To estimate $d_H(\mathbb{V}, \mathbb{V}')$, we proceed in three cases, depending on where the check bits of $\mathbb{V}$ and $\mathbb{V}'$ come from.

Case 1. $\bar{j} \in [0, n_2 - (n_R - k_R)m - 1]$. As shown in Figure 5(a), we partition $\mathbb{W}$ into four blocks by setting

$$\mathbb{W}_{\mathrm{I}} = \mathbb{W}[0, \hat{i} - 1][0, \hat{j} - 1], \qquad \mathbb{W}_{\mathrm{II}} = \mathbb{W}[0, \hat{i} - 1][\hat{j}, n_2 - 1],$$
$$\mathbb{W}_{\mathrm{III}} = \mathbb{W}[\hat{i}, n_1 - 1][0, \hat{j} - 1], \qquad \mathbb{W}_{\mathrm{IV}} = \mathbb{W}[\hat{i}, n_1 - 1][\hat{j}, n_2 - 1]$$

and partition $\mathbb{A}_{ab}$ into four blocks by setting

$$\mathbb{A}_{\mathrm{I}} = \mathbb{A}_{ab}[\bar{i}, n_1 - 1][\bar{j}, n_2 - 1], \qquad \mathbb{A}_{\mathrm{II}} = \mathbb{A}_{ab}[\bar{i}, n_1 - 1][0, \bar{j} - 1],$$
$$\mathbb{A}_{\mathrm{III}} = \mathbb{A}_{ab}[0, \bar{i} - 1][\bar{j}, n_2 - 1], \qquad \mathbb{A}_{\mathrm{IV}} = \mathbb{A}_{ab}[0, \bar{i} - 1][0, \bar{j} - 1].$$

Then we have

$$\mathbb{V} = \begin{pmatrix} \mathbb{W}_{\mathrm{IV}} & \mathbb{W}_{\mathrm{III}} \\ \mathbb{W}_{\mathrm{II}} & \mathbb{W}_{\mathrm{I}} \end{pmatrix}, \mathbb{A}_{ab} = \begin{pmatrix} \mathbb{A}_{\mathrm{IV}} & \mathbb{A}_{\mathrm{III}} \\ \mathbb{A}_{\mathrm{II}} & \mathbb{A}_{\mathrm{I}} \end{pmatrix},$$

and $\mathbb{W}_{\mathrm{I}} = \mathbb{A}_{\mathrm{I}}$.

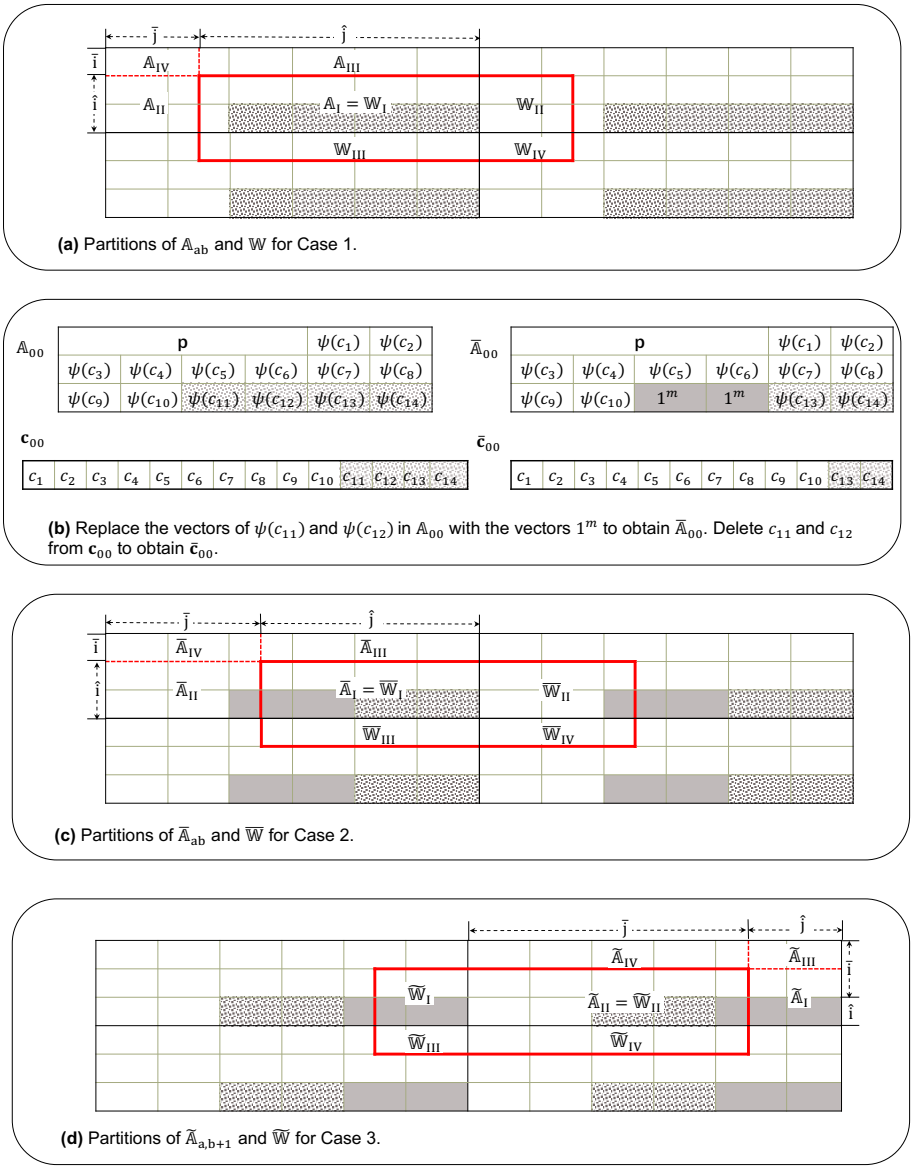

FIG. 5. *An example with $n_1 = 3$ and $n_2 = 6m$ to illustrate the proof of Lemma 4.3. The black lines enclose the arrays $\mathbb{A}_{ab}$, $\mathbb{A}_{a,b+1}$, $\mathbb{A}_{a+1,b}$, and $\mathbb{A}_{a+1,b+1}$. The red lines enclose the subarray $\mathbb{W}$. The empty blocks represent the vectors of message bits. The blocks with dots represent the vectors of check bits. The solid blocks represent the vectors 1^m.*

Notice that all these blocks, except $\mathbb{W}_{\rm I}$ and $\mathbb{A}_{\rm I}$, do not contain check bits, and $(\boldsymbol{\sigma}_0, \boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_{M-1})$ is a Gray code. It follows that

$$d_B(\mathbb{W}_{\rm II}, \mathbb{A}_{\rm II}) \le d_H(\boldsymbol{\sigma}_a\boldsymbol{\sigma}_b, \boldsymbol{\sigma}_a\boldsymbol{\sigma}_{b+1}) \le 1,$$
$$d_B(\mathbb{W}_{\rm III}, \mathbb{A}_{\rm III}) \le d_H(\boldsymbol{\sigma}_a\boldsymbol{\sigma}_b, \boldsymbol{\sigma}_{a+1}\boldsymbol{\sigma}_b) \le 1,$$
$$\text{and} \quad d_B(\mathbb{W}_{\rm IV}, \mathbb{A}_{\rm IV}) \le d_H(\boldsymbol{\sigma}_a\boldsymbol{\sigma}_b, \boldsymbol{\sigma}_{a+1}\boldsymbol{\sigma}_{b+1}) \le 2.$$

So,

$$d_B(\mathbb{V}, \mathbb{A}_{ab}) \le d_B(\mathbb{W}_{\mathrm{I}}, \mathbb{A}_{\mathrm{I}}) + d_B(\mathbb{W}_{\mathrm{II}}, \mathbb{A}_{\mathrm{II}}) + d_B(\mathbb{W}_{\mathrm{III}}, \mathbb{A}_{\mathrm{III}}) + d_B(\mathbb{W}_{\mathrm{IV}}, \mathbb{A}_{\mathrm{IV}}) \le 4.$$

With the same argument, we can get $d_B(\mathbb{V}', \mathbb{A}_{a'b'}) \le 4$. Hence,

$$d_H(\mathbb{V}, \mathbb{V}') \ge d_B(\mathbb{V}, \mathbb{V}') \ge d_B(\mathbb{A}_{ab}, \mathbb{A}_{a'b'}) - 8 = d_H(\boldsymbol{c}_{ab}, \boldsymbol{c}_{a'b'}) - 8 \ge d.$$

Case 2. $\bar{j} \in [n_2 - (n_R - k_R)m, n_2 - (n_R - k_R)m/2 - 1]$. For $\alpha, \beta \in [\![M]\!]$, we change the bits in the block $\mathbb{A}_{\alpha\beta}[n_1 - 1, n_1 - 1][n_2 - (n_R - k_R)m, n_2 - (n_R - k_R)m/2 - 1]$ to one and denote the resulting array as $\bar{\mathbb{A}}_{\alpha\beta}$. Let $\bar{\boldsymbol{c}}_{\alpha\beta}$ be the shortened codeword of $\boldsymbol{c}_{\alpha,\beta}$ by deleting the subword $\boldsymbol{c}_{\alpha\beta}[k_R, n_R - (n_R - k_R)/2 - 1]$ (see Figure 5(b)). Then we have

$$d_B(\bar{\mathbb{A}}_{\alpha\beta}, \bar{\mathbb{A}}_{\alpha'\beta'}) = d_H(\bar{\boldsymbol{c}}_{\alpha\beta}, \bar{\boldsymbol{c}}_{\alpha'\beta'}) \ge (n_R - (n_R - k_R)/2) - k_R + 1 \ge d + 8. \tag{4.1}$$

Now, let $\bar{\mathbb{W}}$, $\bar{\mathbb{W}}'$, $\bar{\mathbb{V}}$, and $\bar{\mathbb{V}}'$ be the corresponding arrays of $\mathbb{W}$, $\mathbb{W}'$, $\mathbb{V}$, and $\mathbb{V}'$ with some check bits being changed to one. As in Case 1, we can show that

$$d_B(\bar{\mathbb{V}}, \bar{\mathbb{A}}_{ab}) \le 4 \text{ and } d_B(\bar{\mathbb{V}}', \bar{\mathbb{A}}'_{ab}) \le 4. \tag{4.2}$$

The only difference is that $\bar{\mathbb{W}}_{\mathrm{II}}$, $\bar{\mathbb{A}}_{\mathrm{II}}$, $\bar{\mathbb{W}}'_{\mathrm{II}}$, and $\bar{\mathbb{A}}'_{\mathrm{II}}$ may contain the check bits. However, these bits are set to one, so we have $d_B(\bar{\mathbb{W}}_{\mathrm{II}}, \bar{\mathbb{A}}_{\mathrm{II}}) \le d_H(\boldsymbol{\sigma}_a\boldsymbol{\sigma}_b, \boldsymbol{\sigma}_a\boldsymbol{\sigma}_{b+1}) \le 1$ (see Figure 5(c)).

It follows from (4.1) and (4.2) that

$$d_B(\bar{\mathbb{V}}, \bar{\mathbb{V}}') \ge d_B(\bar{\mathbb{A}}_{ab}, \bar{\mathbb{A}}_{a'b'}) - 8 \ge d,$$

and then

$$d_H(\mathbb{V}, \mathbb{V}') \ge d_H(\bar{\mathbb{V}}, \bar{\mathbb{V}}') \ge d_B(\bar{\mathbb{V}}, \bar{\mathbb{V}}') \ge d.$$

Case 3. $\bar{j} \in [n_2 - (n_R - k_R)m/2, n_2 - 1]$. For $\alpha, \beta \in [\![M]\!]$, we change the bits in the block $\mathbb{A}_{\alpha\beta}[n_1 - 1, n_1 - 1][n_2 - (n_R - k_R)m/2, n_2 - 1]$ to one and denote the resulting array as $\tilde{\mathbb{A}}_{\alpha\beta}$. Let $\tilde{\boldsymbol{c}}_{\alpha\beta}$ be the shortened codeword of $\boldsymbol{c}_{\alpha,\beta}$ by deleting the last $(n_R - k_R)/2$ bits.

Now, let $\tilde{\mathbb{W}}$, $\tilde{\mathbb{W}}'$, $\tilde{\mathbb{V}}$, and $\tilde{\mathbb{V}}'$ be the corresponding arrays of $\mathbb{W}$, $\mathbb{W}'$, $\mathbb{V}$ and $\mathbb{V}'$. Again we use the strategy in Case 1 to show that

$$d_H(\mathbb{V}, \mathbb{V}') \ge d_H(\tilde{\mathbb{V}}, \tilde{\mathbb{V}}') \ge d_B(\tilde{\mathbb{V}}, \tilde{\mathbb{V}}') \ge d_B(\tilde{\mathbb{A}}_{a,b+1}, \tilde{\mathbb{A}}_{a',b'+1}) - 8 \ge d.$$

We note that in this case we need to partition $\tilde{\mathbb{A}}_{a,b+1}$ instead of $\tilde{\mathbb{A}}_{ab}$ (see Figure 5(d)). ∎

THEOREM 4.4. *The array* $\mathbb{A}$ *in Construction* 2 *is an* $(n_1 \times n_2, d)$*-RPA.*

Proof. Let $\mathbb{W}$ and $\mathbb{W}'$ be two distinct subarrays of dimension $n_1 \times n_2$ in $\mathbb{A}$. Assume that $\mathbb{W} = \mathbb{A}[i, i + n_1 - 1][j, j + n_2 - 1]$ and $\mathbb{W}' = \mathbb{A}[i', i' + n_1 - 1][j', j' + n_2 - 1]$, where $(i, j) \ne (i', j')$. From Lemma 4.3, we have $d_H(\mathbb{W}, \mathbb{W}') \ge d$ when $i \equiv i' \pmod{n_1}$ and $j \equiv j' \pmod{n_2}$. Now we consider the case where $i \not\equiv i' \pmod{n_1}$ or $j \not\equiv j' \pmod{n_2}$. Let $\hat{i} \in [n_1]$ and $\hat{j} \in [n_2]$ such that $i + \hat{i} \equiv 0 \pmod{n_1}$ and $j + \hat{j} \equiv 0 \pmod{n_2}$. So we have $i' + \hat{i} \not\equiv 0 \pmod{n_1}$ or $j' + \hat{j} \not\equiv 0 \pmod{n_2}$. It follows from Lemma 4.2 that $\mathbb{W}[\hat{i}, \hat{i}][\hat{j}, \hat{j} + 4m - 1] = \boldsymbol{p}$ and $d_H(\mathbb{W}'[\hat{i}, \hat{i}][\hat{j}, \hat{j} + 4m - 1], \boldsymbol{p}) \ge d$. Hence

$$d_H(\mathbb{W}, \mathbb{W}') \ge d_H(\mathbb{W}[\hat{i}, \hat{i}][\hat{j}, \hat{j} + 4m - 1], \mathbb{W}'[\hat{i}, \hat{i}][\hat{j}, \hat{j} + 4m - 1]) \ge d,$$

which completes the proof. ∎

COROLLARY 4.5. *Let $\mathbb{W}$ be a window of area A and thickness bounded by a constant. Then there is a binary RPA for $\mathbb{W}$ with distance d and redundancy at most*

$$4.21d\log A + 35.79\log A + o(1),$$

provided A is large enough.

Proof. The RPA $\mathbb{A}$ in Construction 2 has dimension $(n_1M)\times(n_2M)$, where $M = q^{k_R/2}$. So, its redundancy is given by

$$n_1n_2 - \log(n_1n_2M^2) = A - k_R\log q - \log A,$$

where $k_R = n_R - 2(d+7) = A/m - 2d - 18$, and

$$\log q \geq \log(r - r^\theta) = \log r + \log\left(1 - \frac{1}{r^{1-\theta}}\right) \geq \log r - \frac{\log e}{r^{1-\theta}-1}. \tag{4.3}$$

Recall that $\theta = 0.525$, and so $m = \log A/(1-\theta) \geq (3/2)\log A$. It follows that

$$\begin{aligned} r &= \sum_{i=d}^{m}\binom{m}{i} = 2^m\left[1 - \sum_{i=0}^{d-1}\binom{m}{i}\frac{1}{2^m}\right] \geq 2^m\left[1 - \exp\left(-\frac{2(m/2-d+1)^2}{m}\right)\right] \\ &\geq 2^m\left(1 - e^{-\frac{2m}{3\log e}}\right) \geq 2^m\left(1 - \frac{1}{A}\right). \end{aligned}$$

Then we have

$$\log r \geq m + \log\left(1 - \frac{1}{A}\right) \geq m - \frac{\log e}{A-1}. \tag{4.4}$$

On the other hand,

$$r^{1-\theta} \geq \left(\frac{2^m}{2}\right)^{1-\theta} = \frac{A}{2^{1-\theta}}. \tag{4.5}$$

Combining (4.3), (4.4), and (4.5), we get

$$\log q \geq m - O\left(\frac{1}{A}\right).$$

Hence, the redundancy of $\mathbb{A}$ is at most

$$\begin{aligned} &A - \left(\frac{A}{m} - 2d - 18\right)\left(m - O\left(\frac{1}{A}\right)\right) - \log A \\ &\quad = \frac{2d}{1-\theta}\log A + \frac{18}{1-\theta}\log A - \log A + o(1) \\ &\quad \approx 4.21d\log A + 36.89\log A + o(1). \end{aligned}$$

∎

To conclude, we provide an efficient locating algorithm for the array $\mathbb{A}$ in Construction 2. Let χ be a map from $\mathbb{F}_2^m$ to $\mathbb{F}_q$ such that $\chi(\psi(x)) = x$ for all $x \in \mathbb{F}_q$ and $\chi(\boldsymbol{v}) = 0$ for all $\boldsymbol{v} \notin \{\psi(x) : x \in \mathbb{F}_q\}$.

We briefly describe Algorithm 4.1. Suppose that $\mathbb{W}$ is an $n_1 \times n_2$ subarray of $\mathbb{A}$ that is corrupted at e positions with $e \leq (d-1)/2$. So there is a unique pair (i,j) such that $d_H(\mathbb{A}[i, i+n-1][j, j+n-1], \mathbb{W}) \leq (d-1)/2$. Assume that $i = an_1 + \bar{i}$ and $j = bn_2 + \bar{j}$ with $\bar{i} \equiv i \pmod{n_1}$ and $\bar{j} \equiv j \pmod{n_2}$. In what follows, we briefly describe how to determine $a, b, \bar{i}$, and $\bar{j}$.

(I) We first use Lemma 4.2 to determine $\bar{i}$ and $\bar{j}$.

(II) Next, we rotate $\mathbb{W}$ appropriately to obtain $\mathbb{V}$, so that the concatenation of the rows of $\mathbb{V}$, denoted as $\boldsymbol{v}$, is the binary image obtained from either a q-ary codeword $\boldsymbol{c}_{a,b}$ or a concatenation of some shortened codewords $\boldsymbol{c}_{a,b}$, $\boldsymbol{c}_{a+1,b}$, $\boldsymbol{c}_{a,b+1}$, and $\boldsymbol{c}_{a+1,b+1}$. Since $\boldsymbol{v}$ is obtained via the map ψ and prepending the string $\boldsymbol{p}$, we reverse this process to obtain the q-ary estimate $\boldsymbol{u}$.

(III) Finally, depending on the value of $\bar{j}$, we apply the Reed–Solomon decoding algorithm $\mathrm{dec}_{\mathrm{RS}}$ to find either $\boldsymbol{c}_{a,b}$ (when $\bar{j} \in [0, n_2 - 2(d+7)m - 1]$), some shortened version of $\boldsymbol{c}_{a,b}$ (when $\bar{j} \in [n_2 - 2(d+7)m, n_2 - (d+7)m - 1]$), or some shortened version of $\boldsymbol{c}_{a,b+1}$ (when $j \in [n_2 - (d+7)m, n_2 - 1]$). Therefore, we determine a and b and hence obtain $i = an_1 + \hat{i}$ and $j = bn_2 + \hat{j}$.

The first step above requires $\ell_p n_1 n_2$ comparisons. The Reed–Solomon decoding runs in $O(n_R^3) = O((n_1 n_2)^3)$ time, and the decoding of Gray codes runs in $O(k_R(\log q)^2) = O(n_1 n_2(\log(n_1 n_2))^2)$ time. Therefore, Algorithm 3.1 can determine the location in $O((n_1 n_2)^3)$ or equivalently $O(A^3)$ time.

Algorithm 4.1 Locating algorithm for the array $\mathbb{A}$ in Construction 2.

Input: an $n_1 \times n_2$ window $\mathbb{W}$ of area A and thickness bounded by a constant
Output: a position $(i, j) \triangleq (an_1 + \bar{i}, bn_2 + \bar{j})$ such that $d_H(\mathbb{A}[i, i + n_1 - 1][j, j + n_2 - 1], \mathbb{W}) \leq (d-1)/2$

$m \leftarrow \frac{\log A}{1-\theta}$
$n_R \leftarrow n_1(n_2/m) - 4$
$k_R \leftarrow n_R - 2(d+7)$

$(\hat{i}, \hat{j}) \leftarrow$ unique tuple such that $d_H(\mathbb{W}[\hat{i}, \hat{i}][\hat{j}, \hat{j} + 4m - 1], \boldsymbol{p}) \leq (d-1)/2$
Set $\bar{i} \in [\![n_1]\!]$ such that $\bar{i} + \hat{i} \equiv 0 \pmod{n_1}$
Set $\bar{j} \in [\![n_2]\!]$ such that $\bar{j} + \hat{j} \equiv 0 \pmod{n_2}$

$\mathbb{V} \leftarrow$ the array obtained by shifting $\mathbb{W}$ cyclically upward $\hat{i}$ times and leftward $\hat{j}$ times
$\boldsymbol{v} \leftarrow$ the concatenation of the rows of $\mathbb{V}$
$\boldsymbol{u} \leftarrow \chi(\boldsymbol{v}[4m, 5m-1])\chi(\boldsymbol{v}[5m, 6m-1])\chi(\boldsymbol{v}[6m, 7m-1]) \cdots \chi(\boldsymbol{v}[n_1 n_2 - m, n_1 n_2 - 1])$

if $\bar{j} \in [0, n_2 - 2(d+7)m - 1]$ **then**
  $\boldsymbol{c} \leftarrow \mathrm{dec}_{\mathrm{RS}}^{(n_R, k_R)}(\boldsymbol{u})$
else if $\bar{j} \in [n_2 - 2(d+7)m, n_2 - (d+7)m - 1]$ **then**
  $\boldsymbol{u}^s \leftarrow$ the shortened codeword $\boldsymbol{u}[0, k_R - 1]\boldsymbol{u}[k_R + (d+7), n_R - 1]$
  $\boldsymbol{c} \leftarrow \mathrm{dec}_{\mathrm{RS}}^{(n_R - (d+7), k_R)}(\boldsymbol{u}^s)$
else
  $\boldsymbol{u}^s \leftarrow$ the shortened codeword $\bar{\boldsymbol{c}}[0, k_R + d + 6]$
  $\boldsymbol{c} \leftarrow \mathrm{dec}_{\mathrm{RS}}^{(n_R - (d+7), k_R)}(\boldsymbol{u}^s)$
$a \leftarrow \mathrm{dec}_{\mathrm{Gray}}(\boldsymbol{c}[0, k_R/2 - 1])$
if $\bar{j} \in [0, n_2 - (d+7)m - 1]$ **then**
  $b \leftarrow \mathrm{dec}_{\mathrm{Gray}}(\boldsymbol{c}[k_R/2, k_R - 1])$
else
  $b + 1 \leftarrow \mathrm{dec}_{\mathrm{Gray}}(\boldsymbol{c}[k_R/2, k_R - 1])$
return $(an_1 + \bar{i}, bn_2 + \bar{j})$

5. Binary positioning arrays with constant rank distance. We continue our investigation of binary robust positioning arrays. In this section, we consider the scenario where the error patterns are confined to a certain number of rows or columns (or both). To correct for such errors, Roth demonstrated that it suffices to consider codes in the *rank distance* metric [16].

For two matrices $\mathbb{M}_1$ and $\mathbb{M}_2$ of the same dimension, the *rank distance* between them, denoted as $d_R(\mathbb{M}_1, \mathbb{M}_2)$, is defined as the rank of their difference, i.e., $d_R(\mathbb{M}_1, \mathbb{M}_2) \triangleq \operatorname{rank}(\mathbb{M}_1 - \mathbb{M}_2)$. In this section, we modify Construction 2 to produce a binary positioning array of strength $n_1 \times n_2$ and rank distance d, i.e., a large array in which the rank distance between any two $n_1 \times n_2$ submatrices is at least d. Since a code $\mathcal{M} \subseteq \mathbb{F}_q^{n_1 \times n_2}$ with minimum rank distance d satisfies the Singleton bound, i.e., $|\mathcal{M}| \leq q^{n_2(n_1-d+1)}$, the redundancy of such an array should be at least $n_2(d-1) - O(1)$.

To present our construction, we require the concept of maximum rank distance (MRD) codes.

THEOREM 5.1 (MRD code [6]). *Let q be a prime power. Suppose that $N_1 \leq N_2$. Then there exists a linear code $\mathcal{M} \leq F_q^{N_1 \times N_2}$ of rank distance d and dimension $N_2(N_1 - d + 1)$.*

We also need to choose a new marker $\mathbb{P}$. Fix d and let m be an integer such that $m(m-d+1) = \log(n_1 n_2)$. Let $\mathbb{P}$ be a $4m \times 4m$ array in which

(i) the diagonal is $0^{4m-\ell}\boldsymbol{u}$, where $\boldsymbol{u}$ is the d-auto-cyclic vector provided in (3.1);
(ii) the $d \times d$ subarrays at the right top corner and left bottom corner are identity matrices;
(iii) the symbols in all the other entries are 0.

Let $\mathcal{M} \subseteq \mathbb{F}_2^{m \times m}$ be an MRD code of rank distance d and dimension $m(m-d+1)$. Suppose that both n_1 and n_2 are divisible by m. Set $n_R \triangleq \frac{n_1 n_1}{m^2} - 16$ and $k_R \triangleq n_R - 24$. Choose a prime power q such that $n_R \leq q < 2^{m(m-d+1)}$ and set $M \triangleq q^{k_R/2}$. Take an arbitrary injective map ψ from $\mathbb{F}_q$ to $\mathcal{M} \backslash \{\mathbf{0}\}$, where $\mathbf{0}$ is the all-zero matrix. So ψ maps the elements of $\mathbb{F}_q$ to $m \times m$ matrices of rank at least d.

Construction 3. Let $\mathcal{G} = (\boldsymbol{\sigma}_0, \boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_{M-1})$ be a $(k_R/2, q)$-Gray code and consider a Reed–Solomon code of length n_R and dimension k_R over $\mathbb{F}_q$. For each $0 \leq i, j \leq M-1$, set $\boldsymbol{c}_{ij} = \operatorname{enc}_{\mathrm{RS}}^{(n_R, k_R)}(\boldsymbol{\sigma}_i \boldsymbol{\sigma}_j)$.

For each $\boldsymbol{c}_{ij}$, apply the map ψ to the symbols of $\boldsymbol{c}_{ij}$ to obtain n_R $m \times m$ matrices of rank at least d. Since $n_R = (n_1 n_2)/m^2 - 16$ and $n_R - k_R = 24$, tile these n_R matrices together with the marker $\mathbb{P}$ to form an $n_1 \times n_2$ array $\mathbb{A}_{ij}$ (see Figure 6) such that

(i) $\mathbb{A}_{ij}[0, 4m-1][0, 4m-1] = \mathbb{P}$;
(ii) for each $\ell \in \{1, 2, 3\}$, $\mathbb{A}_{ij}[n_1 - 2m\ell, n_1 - 2m(\ell-1) - 1][n_2 - 4m\ell, n_2 - 4m(\ell - 1) - 1]$ comprises eight $m \times m$ submatrices, each of which corresponds to a check bit of $\boldsymbol{c}_{ij}$.

Finally, construct a large array $\mathbb{A}$ as

$$\mathbb{A} = \begin{pmatrix} \mathbb{A}_{00} & \mathbb{A}_{01} & \cdots & \mathbb{A}_{0,M-1} \\ \mathbb{A}_{10} & \mathbb{A}_{11} & \cdots & \mathbb{A}_{1,M-1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbb{A}_{M-1,0} & \mathbb{A}_{M-1,1} & \cdots & \mathbb{A}_{M-1,M-1} \end{pmatrix}.$$

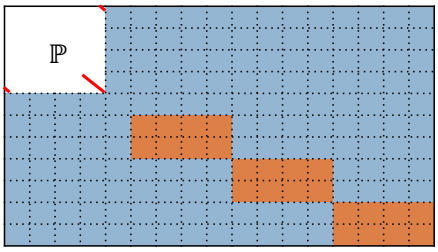

FIG. 6. *An example for the matrix A_{ij} in Construction 3 with $n_1 = 11m$ and $n_2 = 17m$. The blue cells represent the message bits and the yellow cells represent the check bits.*

LEMMA 5.2. *Consider the subarray $\mathbb{W} = \mathbb{A}[i_0, i_0 + n_1 - 1][j_0, j_0 + n_2 - 1]$ in $\mathbb{A}$. Pick $i \in [\![n_1]\!]$ and $j \in [\![n_2]\!]$. Then the following hold.*

(i) *If $i+i_0 \equiv 0 \pmod{n_1}$ and $j+j_0 \pmod{n_2}$, then $\mathbb{W}[i, i+4m-1][j, j+4m-1] = \mathbb{P}$.*

(ii) *If $i+i_0 \not\equiv 0 \pmod{n_1}$ or $j+j_0 \not\equiv 0 \pmod{n_2}$, then $d_R(\mathbb{W}[i, i+4m-1][j, j+4m-1], \mathbb{P}) \geq d$.*

Proof. For simplicity, we assume that $n_1 = n_2 = n$. The case of $n_1 \neq n_2$ can proceed similarly. Let $\hat{i}, \hat{j} \in [\![n]\!]$ such that $\hat{i} + i_0 \equiv 0 \pmod{n}$ and $\hat{j} + j_0 \equiv 0 \pmod{n}$. We consider the array $\mathbb{V}$, which is obtained by shifting $\mathbb{W}$ cyclically upward $\hat{i}$ times and leftward $\hat{j}$ times. Then $\mathbb{V}[0, 4m-1][0, 4m-1] = \mathbb{P}$ and it suffices to show that $d_R(\mathbb{V}[i, i+4m-1][j, j+4m-1], \mathbb{P}) \geq d$ for $(i, j) \in [\![n]\!]^2 \backslash \{(0,0)\}$. Write $\mathrm{Diag}(\boldsymbol{x})$ for a diagonal matrix whose diagonal is $\boldsymbol{x}$.

We first assume $i = j$. Similar to the proof of Lemma 3.6, we consider the following cases.

Case 1a. $i \in [1, d]$. Then $\mathbb{P}[4m-\ell-i, 4m-1-i][4m-\ell-i, 4m-1-i] = \mathrm{Diag}(0^i \boldsymbol{u}[0, \ell-i-1])$. On the other hand, the corresponding subarray in $\mathbb{V}[i, i+4m-1][j, j+4m-1]$ is $\mathbb{V}[4m-\ell, 4m-1][4m-\ell, 4m-1]$, which is equal to $\mathrm{Diag}(\boldsymbol{u})$. Due to the property of $\boldsymbol{u}$, the rank distance between them is at least d and so $d_R(\mathbb{V}[i, i+4m-1][j, j+4m-1], \mathbb{P}) \geq d$.

Case 1b. $i \in [d+1, 4m-d]$. Since $4m-\ell > \ell$, $\mathbb{V}[i, i+4m-\ell-1][i, i+4m-\ell-1]$ contains at least one subarray which is equal to $\mathrm{Diag}(1^d)$. Note that $\mathbb{P}[0, 4m-\ell-1][0, 4m-\ell-1] = \mathbf{0}$. The rank distance between $\mathbb{V}[i, i+4m-\ell-1][i, i+4m-\ell-1]$ and $\mathbb{P}[0, 4m-\ell-1][0, 4m-\ell-1]$ is at least d and so we have $d_R(\mathbb{V}[i, i+4m-1][j, j+4m-1], \mathbb{P}) \geq d$.

Case 1c. $i \in [4m-d+1, n-(4m-\ell)]$. Since $4m-\ell-d > 3m$, $\mathbb{V}[i+d, i+4m-\ell-1][i+d, i+4m-\ell-1]$ should contain at least one $m \times m$ subarray of rank at least d. Noting that $\mathbb{P}[i+d, i+4m-\ell-1][i+d, i+4m-\ell-1] = \mathbf{0}$, again we have $d_R(\mathbb{V}[i, i+4m-1][j, j+4m-1], \mathbb{P}) \geq d$.

Case 1d. $i \in [n-(4m-\ell)+1, n-d]$. Since $i+4m-\ell-n \geq 1$ and $i+4m-\ell+d-1-n \leq 4m-\ell-1$, we have $\mathbb{V}[i+4m-\ell, i+4m-\ell+d-1][i+4m-\ell, i+4m-\ell+d-1] = \mathbf{0}$. Note that $\mathbb{P}[4m-\ell, 4m-\ell+d-1] = \mathrm{Diag}(1^d)$. It follows that $d_R(\mathbb{V}[i, i+4m-1][j, j+4m-1], \mathbb{P}) \geq d$.

Case 1e. $i \in [n-d+1, n-1]$. Let $\delta = n-i$, then $\delta \in [1, d-1]$. We have

$$\begin{aligned}
&\mathbb{V}[i+4m-\ell, i+4m-1][i+4m-\ell, i+4m-1] \\
&\quad = \mathbb{V}[4m-\ell-\delta, 4m-1-\delta][4m-\ell-\delta, 4m-1-\delta] \\
&\quad = \mathrm{Diag}(0^\delta \boldsymbol{u}[0, \ell-\delta-1]).
\end{aligned}$$

Since $\mathbb{P}[4m-\ell,4m-1][4m-\ell,4m-1]=\mathrm{Diag}(\boldsymbol{u})$, the rank distance between them is at least d and so we have $d_R(\mathbb{V}[i,i+4m-1][j,j+4m-1],\mathbb{P})\geq d$.

In the following we assume that $i<j$; the case of $i>j$ can proceed in the same way.

Case 2a. $j\in[1,4m-2d]$. Then $d\leq 4m-d-j<4m-d-i$ and $4m-1-j<4m-1-i\leq 4m-1$. It follows that the subarray $\mathbb{P}[4m-d-i,4m-1-i][4m-d-i,4m-1-i]$ is an upper triangular matrix with all entries on the diagonal being 0. On the other hand, in $\mathbb{V}[i,i+4m-1][j,j+4m-1]$ the corresponding subarray $\mathbb{V}[4m-d,4m-1][4m-d,4m-1]$ is an identity matrix. Hence the rank distance between $\mathbb{V}[i,i+4m-1][j,j+4m-1]$ and $\mathbb{P}$ is at least d.

Case 2b. $j\in[4m-2d+1,n-(4m-\ell)]$. In this case we estimate the rank distance between the submatrices

$$\mathbb{V}[i+2d,i+4m-\ell-1][j+2d,j+4m-\ell-1]$$

and $\mathbb{P}[2d,4m-\ell-1][2d,4m-\ell-1]$. Since $4m\leq j+2d$, $j+4m-\ell-1\leq n-1$ and $4m-\ell-2d>3m$, the subarray $\mathbb{V}[i+2d,i+4m-\ell-1][j+2d,j+4m-\ell-1]$ always contains an $m\times m$ submatrix of rank at least d. On the other hand, $\mathbb{P}[2d,4m-\ell-1][2d,4m-\ell-1]=\mathbf{0}$. It follows that the rank distance between them is at least d and so $d_R(\mathbb{V}[i,i+4m-1][j,j+4m-1],\mathbb{P})\geq d$.

Case 2c. $j\in[n-(4m-\ell)+1,n-1]$ and $i\in[0,4m-\ell-d]$. Then

$$\mathbb{V}[i,i+d-1][j+4m-d,j+4m-1]=\mathbb{V}[i,i+d-1][j+4m-d-n,j+4m-1-n].$$

Since $0\leq i<i+d-1\leq 4m-\ell-1$ and $\ell-d<j+4m-d-n<j+4m-1-n<4m-1$, the subarray $\mathbb{V}[i,i+d-1][j+4m-d,j+4m-1]$ is an upper triangular matrix with all entries on the diagonal being 0. Note that in $\mathbb{P}$ the corresponding subarray $\mathbb{P}[0,d-1][4m-d,4m-1]=\mathbb{I}_d$. Thus the rank distance between $\mathbb{V}[i,i+4m-1][j,j+4m-1]$ and $\mathbb{P}$ is at least d.

Case 2d. $j\in[n-(4m-\ell)+1,n-1]$ and $i\in[4m-\ell-d+1,n-(4m-\ell)]$. Then $4m<i+\ell+d<i+4m-\ell-1\leq n-1$. Since $4m-2\ell-d>3m$, the subarray $\mathbb{W}[i+\ell+d,i+4m-\ell-1][j+\ell+d,j+4m-\ell-1]$ always contains an $m\times m$ submatrix of rank at least d. Note that the corresponding subarray $\mathbb{P}[\ell+d,4m-\ell-1][\ell+d,4m-\ell-1]=\mathbf{0}$. It follows that $d_R(\mathbb{V}[i,i+4m-1][j,j+4m-1],\mathbb{P})\geq d$.

Case 2e. $j\in[n-(4m-\ell)+1,n-1]$ and $i\in[n-(4m-\ell)+1,n-1]$. Then

$$\begin{aligned}&\mathbb{V}[i+4m-d,i+4m-1][j+4m-d,j+4m-1]\\&\quad=\mathbb{V}[i+4m-d-n,i+4m-1-n][j+4m-d-n,j+4m-1-n].\end{aligned}$$

Since $\ell-d<i+4m-d-n<j+4m-d-n$ and $i+4m-1-n<j+4m-1-n<4m-1$, the subarray $\mathbb{V}[i,i+d-1][j+4m-d,j+4m-1]$ is a lower triangular matrix with all entries on the diagonal being 0. Note that in $\mathbb{P}$ the corresponding subarray $\mathbb{P}[4m-d,4m-1][4m-d,4m-1]=\mathbb{I}_d$. Thus the rank distance between $\mathbb{V}[i,i+4m-1][j,j+4m-1]$ and $\mathbb{P}$ is at least d. ∎

LEMMA 5.3. *For any two subarrays* $\mathbb{W}=\mathbb{A}[i,i+n_1-1][j,j+n_2-1]$ *and* $\mathbb{W}'=\mathbb{A}[i',i'+n_1-1][j',j'+n_2-1]$ *with* $i\equiv i'\pmod{n_1}$ *and* $j\equiv j'\pmod{n_2}$, *the rank distance between them is at least* d.

Proof. Suppose that $i=an_1+\bar{i}$ and $i'=a'n_1+\bar{i}$ for some $\bar{i}\in[\![n_1]\!]$, and $j=bn_2+\bar{j}$ and $j'=b'n_2+\bar{j}$ for some $\bar{j}\in[\![n_2]\!]$. Let $\hat{i}\in[\![n_1]\!]$ and $\hat{j}\in[\![n_2]\!]$ be the integers such that $\bar{i}+\hat{i}\equiv 0\pmod{n_1}$ and $\bar{j}+\hat{j}\equiv 0\pmod{n_2}$. Shift $\mathbb{W}$ cyclically upward $\hat{i}$

times and leftward $\hat{j}$ times and denote the resulting array as $\mathbb{V}$. Similarly, let $\mathbb{V}'$ be the corresponding shifted array of $\mathbb{W}'$. Then $d_R(\mathbb{W}, \mathbb{W}') = d_R(\mathbb{V}, \mathbb{V}')$. To estimate $d_R(\mathbb{V}, \mathbb{V}')$, we proceed in three cases, depending on where the check bits of $\mathbb{V}$ and $\mathbb{V}'$ come from. Similar to the proof of Lemma 4.3, for any two subarrays $\mathbb{M}$ and $\mathbb{M}'$ in $\mathbb{A}$ which are of same dimension and in the same modular position, we use $d_S(\mathbb{M}, \mathbb{M}')$ to denote the number of different (truncated) $m \times m$ subarrays in $\mathbb{M}$ and $\mathbb{M}'$.

Case 1. $\bar{i} \in [0, n_1 - 2m - 1]$ and $\bar{j} \in [0, n_2 - 4m - 1]$. For $\alpha, \beta \in [\![M]\!]$, we change the bits in the subarrays $\mathbb{A}_{\alpha\beta}[n_1 - 6m, n_1 - 4m - 1][n_2 - 12m, n_2 - 8m - 1]$ and $\mathbb{A}_{\alpha\beta}[n_1 - 4m, n_1 - 2m - 1][n_2 - 8m, n_2 - 4m - 1]$ to one and denote the resulting array as $\bar{\mathbb{A}}_{\alpha\beta}$. Let $\bar{\boldsymbol{c}}_{\alpha\beta}$ be the corresponding shortened codeword of length $n_R - 16$. Then we have

$$d_S(\bar{\mathbb{A}}_{\alpha\beta}, \bar{\mathbb{A}}_{\alpha'\beta'}) = d_H(\bar{\boldsymbol{c}}_{\alpha\beta}, \bar{\boldsymbol{c}}_{\alpha'\beta'}) \geq (n_R - 16) - k_R + 1 \geq 9. \tag{5.1}$$

Now, let $\bar{\mathbb{W}}$, $\bar{\mathbb{W}}'$, $\bar{\mathbb{V}}$, and $\bar{\mathbb{V}}'$ be the corresponding arrays of $\mathbb{W}$, $\mathbb{W}'$, $\mathbb{V}$, and $\mathbb{V}'$ with some check bits being changed to one. We partition $\bar{\mathbb{W}}$ and $\bar{\mathbb{A}}_{ab}$ as in Figure 7(a). Then

$$\bar{\mathbb{V}} = \begin{pmatrix} \bar{\mathbb{W}}_{\mathrm{IV}} & \bar{\mathbb{W}}_{\mathrm{III}} \\ \bar{\mathbb{W}}_{\mathrm{II}} & \bar{\mathbb{W}}_{\mathrm{I}} \end{pmatrix}, \bar{\mathbb{A}}_{ab} = \begin{pmatrix} \bar{\mathbb{A}}_{\mathrm{IV}} & \bar{\mathbb{A}}_{\mathrm{III}} \\ \bar{\mathbb{A}}_{\mathrm{II}} & \bar{\mathbb{A}}_{\mathrm{I}} \end{pmatrix}, \text{ and } \bar{\mathbb{W}}_{\mathrm{I}} = \bar{\mathbb{A}}_{\mathrm{I}}.$$

Furthermore, we have

$$d_S(\bar{\mathbb{W}}_{\mathrm{II}}, \bar{\mathbb{A}}_{\mathrm{II}}) \leq 1, \ d_S(\bar{\mathbb{W}}_{\mathrm{III}}, \bar{\mathbb{A}}_{\mathrm{III}}) \leq 1, \text{ and } \ d_S(\bar{\mathbb{W}}_{\mathrm{IV}}, \bar{\mathbb{A}}_{\mathrm{IV}}) \leq 2.$$

It follows that $d_S(\bar{\mathbb{V}}, \bar{\mathbb{A}}_{ab}) \leq 4$. With the same argument, we can get $d_B(\bar{\mathbb{V}}', \bar{\mathbb{A}}_{a'b'}) \leq 4$. Hence,

$$d_S(\mathbb{V}, \mathbb{V}') \geq d_S(\bar{\mathbb{V}}, \bar{\mathbb{V}}') \geq d_S(\bar{\mathbb{A}}_{ab}, \bar{\mathbb{A}}_{a'b'}) - 8 \geq 1.$$

So we can find a pair of distinct $m \times m$ subarrays in the same position of $\mathbb{V}$ and $\mathbb{V}'$. Since these two subarrays are codewords of an MRD code, the rank distance between them is at least d. Thus $d_R(\mathbb{W}, \mathbb{W}') = d_R(\mathbb{V}, \mathbb{V}') \geq d$.

Case 2. $\bar{i} \in [0, n_1 - 4m - 1]$ and $\bar{j} \in [n_2 - 4m, n_2 - 1]$. We change the bits in the subarrays $\mathbb{A}_{\alpha\beta}[n_1 - 6m, n_1 - 4m - 1][n_2 - 12m, n_2 - 8m - 1]$ and $\mathbb{A}_{\alpha\beta}[n_1 - 2m, n_1 - 1][n_2 - 4m, n_2 - 1]$ to one and denote the resulting array as $\tilde{\mathbb{A}}_{\alpha\beta}$. Let $\tilde{\mathbb{W}}$, $\tilde{\mathbb{W}}'$, $\tilde{\mathbb{V}}$, and $\tilde{\mathbb{V}}'$ be the corresponding arrays of $\mathbb{W}$, $\mathbb{W}'$, $\mathbb{V}$, and $\mathbb{V}'$ with some check bits being changed to one. Partition $\tilde{\mathbb{W}}$ and $\tilde{\mathbb{A}}_{a,b+1}$ as in Figure 7(b). Then using the same strategy as in Case 1, we can show

$$d_S(\tilde{\mathbb{V}}, \tilde{\mathbb{A}}_{ab}) \leq 4, \ d_S(\tilde{\mathbb{V}}', \tilde{\mathbb{A}}'_{ab}) \leq 4 \text{ and } d_S(\tilde{\mathbb{V}}, \tilde{\mathbb{V}}') \geq d_S(\tilde{\mathbb{A}}_{ab}, \tilde{\mathbb{A}}_{a'b'}) - 8 \geq 1.$$

It follows that $d_R(\mathbb{W}, \mathbb{W}') = d_R(\mathbb{V}, \mathbb{V}') \geq d_R(\tilde{\mathbb{V}}, \tilde{\mathbb{V}}') \geq d$.

Case 3. $\bar{i} \in [n_1 - 4m, n_1 - 1]$ and $\bar{j} \in [n_2 - 4m, n_2 - 1]$. In the last case, we change the bits in the subarrays $\mathbb{A}_{\alpha\beta}[n_1 - 4m, n_1 - 2m - 1][n_2 - 8m, n_2 - 6m - 1]$ and $\mathbb{A}_{\alpha\beta}[n_1 - 2m, n_1 - 1][n_2 - 4m, n_2 - 1]$ to one and denote the resulting array as $\hat{\mathbb{A}}_{\alpha\beta}$. Let $\hat{\mathbb{W}}$, $\hat{\mathbb{W}}'$, $\hat{\mathbb{V}}$, and $\hat{\mathbb{V}}'$ be the corresponding arrays of $\mathbb{W}$, $\mathbb{W}'$, $\mathbb{V}$, and $\mathbb{V}'$. Then partition $\hat{\mathbb{A}}_{a+1,b+1}$ as in Figure 7(c). ∎

Using Lemmas 5.2 and 5.3, we have the following result.

THEOREM 5.4. *The array $\mathbb{A}$ in Construction* 3 *is a positioning array in which the rank distance between any two $n_1 \times n_2$ submatrices is at least d.*

It can be checked that the redundancy of $\mathbb{A}$ in Construction 3 is $n_2(d-1)\frac{n_1}{m} + O(\log(n_1 n_2))$, where $m(m - d + 1) = \log(n_1 n_2)$. In contrast, the Singleton bound suggests that the redundancy is at least $n_2(d-1)$.

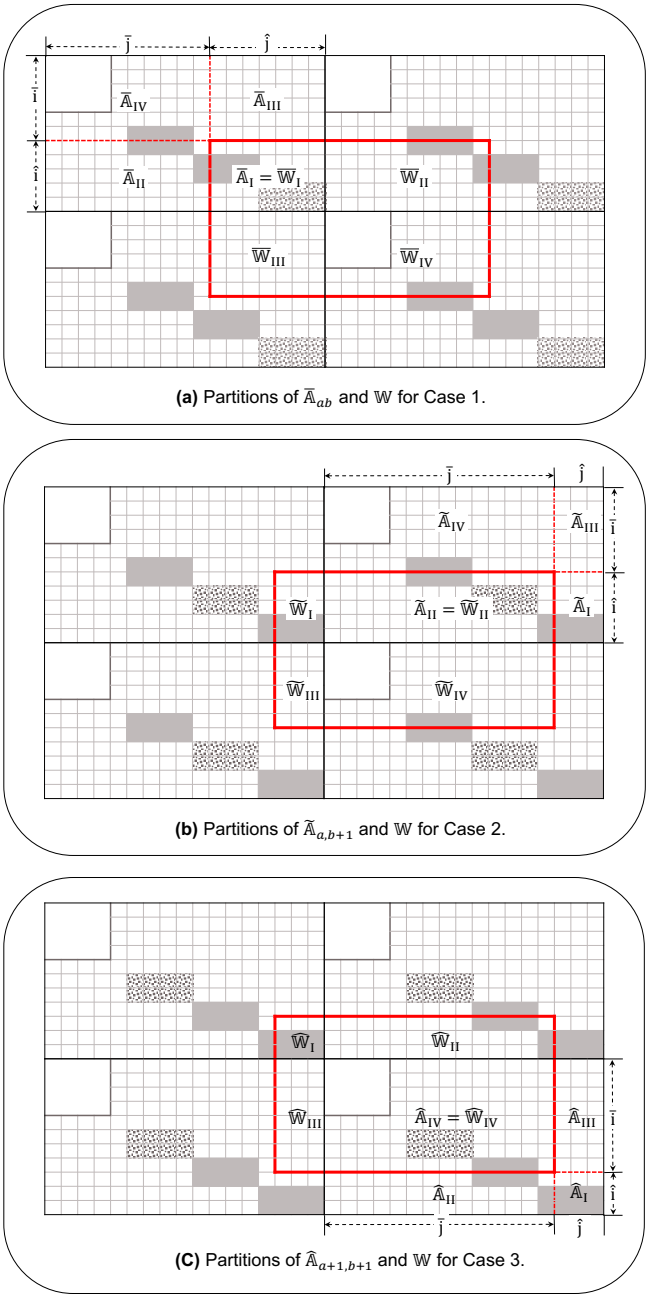

FIG. 7. *An example with $n_1 = 11m$ and $n_2 = 17m$ to illustrate the proof of Lemma* 5.3. *The red lines enclose the subarray $\mathbb{W}$. The empty $m \times m$ subarrays represent message bits. The subarrays with dots represent check bits. The solid subarrays represent the all-one matrices.*

6. *q*-ary robust positioning sequences. In this section, we modify the construction of Berkowitz and Kopparty and give a new class of q-ary positioning sequences robust to a constant fraction of errors. We first review Berkowitz and Kopparty's work.

THEOREM 6.1 (Berkowitz and Kopparty [2]). *Fix a generator g of $\mathbb{F}_q^*$. Let $\mathcal{G} = (\boldsymbol{\sigma}_0, \boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_{q^k-1})$ be a (k,q)-Gray code. For each $\boldsymbol{\sigma}_i$, let $f_i(x) \in \mathbb{F}_q[x]$ be the unique interpolating polynomial of degree $k+1$ so that*

1. $\mathrm{coeff}_x f_i = 1$, $\mathrm{coeff}_1 f_i = 0$;
2. $f_i(g^j) = \sigma[j]$ *for all* $0 \le j < k$.

Define a sequence

$$\boldsymbol{T} = \boldsymbol{t}_0 \boldsymbol{t}_1 \cdots \boldsymbol{t}_{q^k-1},$$

where

$$\boldsymbol{t}_i = (f_i(g^0), f_i(g^1), \ldots, f_i(g^{q-2})).$$

Then $\boldsymbol{T}$ is an $(n,d)_q$-RPS with $n = q-1$ and $d \ge \max\{\frac{n-k}{3} - 3, n-3k-9\}$.

COROLLARY 6.2 (Berkowitz and Kopparty [2]). *For any $0 < R < 1$ and $\delta < \max\{\frac{1-R}{3}, 1-3R\}$, for large enough q there exists a q-ary robust positioning sequence of strength n, rate R, and relative distance δ.*

Now, we use a simple strategy to improve on the relative distance δ: we map the symbols in some positions of $\boldsymbol{T}$ to another alphabet which is disjoint with $\mathbb{F}_q$.

Construction 4. Let E be a set of q elements which is disjoint from $\mathbb{F}_q$. Fix a one-to-one map χ from $\mathbb{F}_q$ to E. For a vector $\boldsymbol{v} = (v_0, v_1, \ldots, v_{\ell-1}) \in \mathbb{F}_q^\ell$, define $\chi(\boldsymbol{v}) = (\chi(v_0), \chi(v_1), \chi(v_2), \ldots, \chi(v_{\ell-1}))$. Now, let $\boldsymbol{t}_0, \boldsymbol{t}_1, \ldots, \boldsymbol{t}_{q^k-1}$ be the family of sequences defined in Theorem 6.1. Construct two sequences

$$\boldsymbol{T}_a = \boldsymbol{a}_0 \boldsymbol{a}_1 \cdots \boldsymbol{a}_{q^k-1} \quad \text{and} \quad \boldsymbol{T}_b = \boldsymbol{b}_0 \boldsymbol{b}_1 \cdots \boldsymbol{b}_{q^k-1},$$

where

$$\boldsymbol{a}_i = \boldsymbol{t}_i \text{ if } i \text{ is even or } \boldsymbol{a}_i = \chi(\boldsymbol{t}_i) \text{ if } i \text{ is odd, and}$$
$$\boldsymbol{b}_i = \boldsymbol{t}_i[0, k-1]\chi(\boldsymbol{t}_i[k, n-1]).$$

We have the following estimation on the distances of $\boldsymbol{T}_a$.

THEOREM 6.3. *The sequence $\boldsymbol{T}_a$ in Construction 4 is an $(n,d)_{2q}$-RPS with $n = q-1$ and $d \ge n-2k$.*

Proof. Let $\boldsymbol{w}_1, \boldsymbol{w}_2$ be two subwords of length n in $\boldsymbol{T}_a$, starting at positions m_1 and m_2, respectively. Let $m_1 = in + \bar{m}_1 \pmod n$ and $m_2 = jn + \bar{m}_2 \pmod n$, where $\bar{m}_1, \bar{m}_2 \in [\![n]\!]$. Assume that $\bar{m}_2 \le \bar{m}_1$, then we can partition the interval $[0, n-1]$ into three pieces by letting

$$I_1 = [0, n-\bar{m}_1-1], I_2 = [n-\bar{m}_1, n-\bar{m}_2-1], \text{ and } I_3 = [n-\bar{m}_2, n-1].$$

We consider the following cases.

Case 1. First assume that both i and j are even, then $\boldsymbol{w}_1[I_1] = \boldsymbol{t}_i[\bar{m}_1, n-1]$ and $\boldsymbol{w}_2[I_1] = \boldsymbol{t}_j[\bar{m}_2, \bar{m}_2 + n - \bar{m}_1 - 1]$. Noting that $\boldsymbol{t}_i$ and $\boldsymbol{t}_j$ are codewords of a Reed–Solomon code, we have $\mathrm{agree}(\boldsymbol{w}_1[I_1], \boldsymbol{w}_2[I_1]) \le k$. Similarly, $\mathrm{agree}(\boldsymbol{w}_1[I_3], \boldsymbol{w}_2[I_3]) \le k$.

Now for the interval I_2, we have $\boldsymbol{w}_1[I_2] = \chi(\boldsymbol{t}_{i+1}[0, \bar{m}_1 - \bar{m}_2 - 1])$ and $\boldsymbol{w}_2[I_2] = \boldsymbol{t}_j[\bar{m}_2 + n - \bar{m}_1, n-1]$, so the symbols of $\boldsymbol{w}_1[I_2]$ come from E and the symbols of $\boldsymbol{w}_2[I_2]$ come from $\mathbb{F}_q$. Since $E \cap \mathbb{F}_q = \emptyset$, $\mathrm{agree}(\boldsymbol{w}_1[I_2], \boldsymbol{w}_2[I_2]) = 0$. Hence $\mathrm{agree}(\boldsymbol{w}_1, \boldsymbol{w}_2) \le 2k$ and $d_H(\boldsymbol{w}_1, \boldsymbol{w}_2) \ge n - 2k$.

Case 2. Here assume that i is even and j is odd, then it is easy to see that the symbols of $\boldsymbol{w}_1[I_1]$ and $\boldsymbol{w}_2[I_3]$ are from $\mathbb{F}_q$ and the symbols of $\boldsymbol{w}_1[I_3]$ and $\boldsymbol{w}_2[I_1]$ are from E. Then $\mathrm{agree}(\boldsymbol{w}_1[I_1], \boldsymbol{w}_2[I_1]) = \mathrm{agree}(\boldsymbol{w}_1[I_3], \boldsymbol{w}_2[I_3]) = 0$. Noting that

$\boldsymbol{w}_1[I_2] = \chi(\boldsymbol{t}_{i+1}[0, \bar{m}_1 - \bar{m}_2 - 1])$ and $\boldsymbol{w}_2[I_2] = \chi(\boldsymbol{t}_j[\bar{m}_2 + n - \bar{m}_1, n-1])$, we have $\text{agree}(\boldsymbol{w}_1[I_2], \boldsymbol{w}_2[I_2]) \le k$. Hence $d_H(\boldsymbol{w}_1, \boldsymbol{w}_2) \ge n - k$.

Case 3. The final case is when i is odd. With the same argument as in Cases 1 and 2, we still can show that $d_H(\boldsymbol{w}_1, \boldsymbol{w}_2) \ge n - 2k$. ∎

For the sequence $\boldsymbol{T}_b$, we have the following result, the proof of which is similar to that of [2, Theorem 6], and we omit here.

THEOREM 6.4. *The sequence $\boldsymbol{T}_b$ in Construction* 3 *is an $(n,d)_{2q}$-RPS with $n = q-1$ and $d \ge \frac{n-k-9}{2}$.*

COROLLARY 6.5. *For any $0 < R < 1$ and $\delta < \max\{\frac{1-R}{2}, 1-2R\}$, for large enough q there exists a q-ary robust positioning sequence of strength n, rate R, and relative distance δ.*

Proof. $\boldsymbol{T}_a$ and $\boldsymbol{T}_b$ have the same rate:

$$R = \frac{\log_{2q}(nq^k)}{n} = \frac{\log_q(nq^k)}{n\log_q(2q)} = \frac{\log_q(nq^k)}{n}\frac{1}{1+\frac{1}{\log q}}$$
$$\ge \left(\frac{k+1}{n} - o(1)\right)\left(1 - O\left(\frac{1}{\log q}\right)\right) = \frac{k+1}{n} - o(1).$$

The relative distance of $\boldsymbol{T}_a$ is $\delta_a \ge \frac{n-2k}{n} = 1 - 2R - o(1)$, and the relative distance of $\boldsymbol{T}_b$ is $\delta_b \ge \frac{n-k-9}{2n} = \frac{1-R}{2} - o(1)$. ∎

Recall that the relative distance of $\boldsymbol{T}$ constructed in Corollary 6.2 is less than $\max\{\frac{1-R}{3}, 1-3R\}$. So, the constructed arrays $\boldsymbol{T}_a$ and $\boldsymbol{T}_b$ have larger relative distance, i.e., $\max\{\frac{1-R}{2}, 1-2R\}$. In contrast, using the Singleton bound, it is easy to see that the relative distance should be no more that $1 - R + o(1)$.

7. The maximum length of a binary robust positioning sequence. In this section, we determine the exact value of $P(n,d)$ for $d \ge \lfloor 2n/3 \rfloor$. We require the following upper bound on $P(n,d)$.

PROPOSITION 7.1 (Plotkin bound). *If d is even and $2d > n$, then $P(n,d) \le 2\lfloor \frac{d}{2d-n} \rfloor + n - 1$; if d is odd and $2d+1 > n$, then $P(n,d) \le 2\lfloor \frac{d+1}{2d+1-n} \rfloor + n - 1$.*

THEOREM 7.2. *If $\lfloor 2n/3 \rfloor + 1 \le d \le n$, we have $P(n,d) = n+1$.*

Proof. According to the Plotkin bound, if $\lfloor 2n/3 \rfloor + 1 \le d \le n$, we have $P(n,d) \le n+1$. It is easy to see that the sequence $(01)^{\lceil n/2 \rceil}0^{n+1-2\lceil n/2 \rceil}$ is an (n,d)-RPS of length $n+1$. ∎

THEOREM 7.3. *Let $n \equiv 0 \pmod 3$, then $P(n, 2n/3) = n+2$.*

Proof. Let $\boldsymbol{s}$ be an $(n, 2n/3)$-RPS of length N. According to the Plotkin bound, we have that $N \le n+3$. Suppose that $N = n+3$, then there are four subwords $\boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{c}_3$, and $\boldsymbol{c}_4$, which are listed in a $4 \times n$ matrix.

$$\boldsymbol{c}_1 = x_1x_2\cdots x_n$$
$$\boldsymbol{c}_2 = x_2x_3\cdots x_{n+1}$$
$$\boldsymbol{c}_3 = x_3x_4\cdots x_{n+2}$$
$$\boldsymbol{c}_4 = x_4x_5\cdots x_{n+3}$$

Since the Plotkin bound is attained in this case, the number of zeros and ones in each column of the matrix is equal. Hence, $x_i = x_{i+4}$, where $1 \le i \le n-1$. We consider the following cases.

Case 1. If $x_1 = x_2$, then $x_3 = x_4$. It follows that $d_H(\boldsymbol{c}_1, \boldsymbol{c}_2) = \lfloor \frac{n}{2} \rfloor < \frac{2n}{3}$, which is a contradiction.

Case 2. If $x_1 = x_3$, then $x_2 = x_4$ and hence $d_H(\boldsymbol{c}_1, \boldsymbol{c}_3) = 0 < \frac{2n}{3}$, which is a contradiction.

Case 3. If $x_1 = x_4$, then $x_2 = x_3$ and hence $d_H(\boldsymbol{c}_2, \boldsymbol{c}_3) = \lfloor \frac{n}{2} \rfloor < \frac{2n}{3}$, which is a contradiction.

Therefore, we have that $N \leq n + 2$. It is easy to see that the binary sequence $(100)^{n/3}(10)$ of length $n + 2$ is an $(n, 2n/3)$-RPS of length $n + 2$. It follows that $P(n, 2n/3) = n + 2$. ∎

THEOREM 7.4. *Let m be a positive integer, then we have that*

$$P(3m+1, 2m) = \begin{cases} 7 & \text{when } m = 1; \\ 13 & \text{when } m = 2; \\ 14 & \text{when } m = 3; \\ 3m+4 & \text{when } m \geq 4. \end{cases}$$

Proof. An exhaustive search shows that $P(4, 2) = 7$, $P(7, 4) = 13$, and $P(10, 6) = 14$. The corresponding optimal RPSs can be found in the appendix.

For $m \geq 4$, the Plotkin bound suggests that $P(3m + 1, 2m) \leq 3m + 4$. In the following, we give a recursive construction of RPSs with length achieving this bound.

Let

$$\boldsymbol{S}_1 \triangleq 0001000.$$

For $m \geq 1$, let

$$\boldsymbol{S}_{m+1} \triangleq \boldsymbol{S}_m[0, m+3]\overline{\boldsymbol{S}_m[m+4]}\boldsymbol{S}_m[m+2, 3m+3],$$

where $\overline{\boldsymbol{S}_m[m+4]}$ is the complement of $\boldsymbol{S}_m[m+4]$.

Obviously, each $\boldsymbol{S}_m$ has length $3m + 4$. By using inductive arguments, one can see that for any $m \geq 1$,

$$\boldsymbol{S}_m[m+1] = \boldsymbol{S}_m[m+4]$$

and

$$(\boldsymbol{S}_m[m+2], \boldsymbol{S}_m[m+3], \overline{\boldsymbol{S}_m[m+4]}) = (1, 0, 1) \text{ or } (0, 1, 0).$$

Now, we use mathematical induction to show that the sequences constructed above are $(3m + 1, 2m)$-RPSs. It is easy to check that $\boldsymbol{S}_1$ is a $(4, 2)$-RPS. Assume that $\boldsymbol{S}_m$ is a $(3m + 1, 2m)$-RPS. Let s_i be the $(1 + i)$th symbol of $\boldsymbol{S}_m$. Consider the following $4 \times (3m + 1)$ matrix:

$$\begin{pmatrix} s_0 & \cdots & s_m & s_{m+1} & s_{m+2} & \cdots & s_{3m} \\ s_1 & \cdots & s_{m+1} & s_{m+2} & s_{m+3} & \cdots & s_{3m+1} \\ s_2 & \cdots & s_{m+2} & s_{m+3} & s_{m+4} & \cdots & s_{3m+2} \\ s_3 & \cdots & s_{m+3} & s_{m+4} & s_{m+5} & \cdots & s_{3m+3} \end{pmatrix}.$$

According to our assumption, any two rows of the matrix above have distance at least $2m$. Now, for $\boldsymbol{S}_{m+1}$, since

$$\boldsymbol{S}_{m+1} = s_0 s_1 \cdots s_{m+1} s_{m+2} s_{m+3} \overline{s_{m+4}} s_{m+2} s_{m+3} \cdots s_{3m+3},$$

the four subwords of length $3m + 4$ form the following matrix:

$$\begin{pmatrix} s_0 & \cdots & s_m & s_{m+1} & s_{m+2} & s_{m+3} & \overline{s_{m+4}} & s_{m+2} & \cdots & s_{3m} \\ s_1 & \cdots & s_{m+1} & s_{m+2} & s_{m+3} & \overline{s_{m+4}} & s_{m+2} & s_{m+3} & \cdots & s_{3m+1} \\ s_2 & \cdots & s_{m+2} & s_{m+3} & \overline{s_{m+4}} & s_{m+2} & s_{m+3} & s_{m+4} & \cdots & s_{3m+2} \\ s_3 & \cdots & s_{m+3} & \overline{s_{m+4}} & s_{m+2} & s_{m+3} & s_{m+4} & s_{m+5} & \cdots & s_{3m+3} \end{pmatrix}.$$

So the second matrix can be obtained from the first matrix by replacing the column $(s_{m+1}, s_{m+2}, s_{m+3}, s_{m+4})^T$ with

$$\begin{pmatrix} s_{m+1} & s_{m+2} & s_{m+3} & \overline{s_{m+4}} \\ s_{m+2} & s_{m+3} & \overline{s_{m+4}} & s_{m+2} \\ s_{m+3} & \overline{s_{m+4}} & s_{m+2} & s_{m+3} \\ \overline{s_{m+4}} & s_{m+2} & s_{m+3} & s_{m+4} \end{pmatrix}.$$

We look at the first row and the second row. Since $(s_{m+2}, s_{m+3}, \overline{s_{m+4}}) = (0,1,0)$ or $(1,0,1)$, the replacement increases the distance by two. Similarly, for the other pairs of rows, except the first and fourth ones, we can see that the distances are increased by two; for the first row and the fourth row, since $s_{m+1} = s_{m+4}$, the distance is increased again by two. Thus, according to our assumption, the distance between any two rows in the second matrix is at least $2m+2$. The proof is completed. ∎

Now, we look at the case of $n \equiv 2 \pmod 3$. A binary vector is called *balanced* if the number of ones and the number of zeros are equal. Let $\mathbb{E}_1$ and $\mathbb{E}_2$ be the following infinite matrices.

$$\mathbb{E}_1 \triangleq \begin{pmatrix} \cdots & 0 & 0 & 1 & 1 & 0 & 0 & \cdots \\ \cdots & 0 & 1 & 1 & 0 & 0 & 1 & \cdots \\ \cdots & 1 & 1 & 0 & 0 & 1 & 1 & \cdots \\ \cdots & 1 & 0 & 0 & 1 & 1 & 0 & \cdots \end{pmatrix},$$

and

$$\mathbb{E}_2 \triangleq \begin{pmatrix} \cdots & 0 & 1 & 0 & 1 & 0 & 1 & \cdots \\ \cdots & 1 & 0 & 1 & 0 & 1 & 0 & \cdots \\ \cdots & 0 & 1 & 0 & 1 & 0 & 1 & \cdots \\ \cdots & 1 & 0 & 1 & 0 & 1 & 0 & \cdots \end{pmatrix}.$$

Let $\mathbb{E}_1(\ell)$ be a $4 \times \ell$ submatrix of $\mathbb{E}_1$ with ℓ consecutive columns. Similarly, let $\mathbb{E}_2(\ell)$ be a $4 \times \ell$ contiguous submatrix of $\mathbb{E}_2$.

THEOREM 7.5. *Let m be a positive integer, then we have that*

$$P(3m+2, 2m+1) = 3m+4.$$

Proof. Let $\boldsymbol{s}$ be an $(3m+2, 2m+1)$-RPS of length N. According to the Plotkin bound, we have that $N \le 3m+5$. Suppose that $N = 3m+5$, then there are four subwords of length $3m+2$, say, $\boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{c}_3$, and $\boldsymbol{c}_4$, which can be listed in the following $4 \times (3m+2)$ matrix:

$$\begin{pmatrix} \boldsymbol{c}_1 \\ \boldsymbol{c}_2 \\ \boldsymbol{c}_3 \\ \boldsymbol{c}_4 \end{pmatrix} = \begin{pmatrix} x_1 & x_2 & \cdots & x_{3m+2} \\ x_2 & x_3 & \cdots & x_{3m+3} \\ x_3 & x_4 & \cdots & x_{3m+4} \\ x_4 & x_5 & \cdots & x_{3m+5} \end{pmatrix}.$$

We consider the sum of the distances between $\boldsymbol{c}_i$ and $\boldsymbol{c}_j$, where $1 \le i \ne j \le 4$. Since $d_H(\boldsymbol{c}_i, \boldsymbol{c}_j) \ge 2m+1$, this sum is at least $24m+12$. In addition, every column contributes at most 8. So, the sum is at most $24m+16$. Therefore, there are at most two unbalanced columns and each of these unbalanced columns should has three identical symbols, i.e, it should be of one of the following forms:

$$\begin{pmatrix} a \\ b \\ b \\ b \end{pmatrix}, \begin{pmatrix} b \\ a \\ b \\ b \end{pmatrix}, \begin{pmatrix} b \\ b \\ a \\ b \end{pmatrix}, \text{ and } \begin{pmatrix} b \\ b \\ b \\ a \end{pmatrix}, \text{ where } a \ne b.$$

Denote these forms as $\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_2$, and $\boldsymbol{u}_4$, respectively. We consider the following cases.

Case 1. The sum of the distances is $24m + 16$. Then all columns are balanced. The same argument as that in the proof of Theorem 7.3 leads to a contradiction.

Case 2. The sum is equal to $24m + 14$. There is only one unbalanced column. Hence, the matrix should be one of the following forms:

$$\mathbb{E}_1(3m+1)\boldsymbol{u}_1,\ \mathbb{E}_2(3m+1-x)\boldsymbol{u}_2\mathbb{E}_1(x),\ \mathbb{E}_1(x)\boldsymbol{u}_3\mathbb{E}_2(3m+1-x),\ \text{ or } \boldsymbol{u}_4\mathbb{E}_1(3m+1).$$

If the matrix has form $\mathbb{E}_1(3m+1)\boldsymbol{u}_1$, then $d_H(\boldsymbol{c}_3, \boldsymbol{c}_4) = \lfloor (3m+1)/2 \rfloor < 2m+1$, a contradiction. If the matrix has form $\mathbb{E}_2(3m+1-x)\boldsymbol{u}_2\mathbb{E}_1(x)$, we have that $d_H(\boldsymbol{c}_1, \boldsymbol{c}_4) = 3m + 1 - x + \lfloor x/2 \rfloor$ and $d_H(\boldsymbol{c}_1, \boldsymbol{c}_3) = x$, both of which should be at least $2m+1$. That's impossible. For the other two cases, we may consider the reverse of $\boldsymbol{s}$ to get the contradiction.

Case 3. The sum of the distances is $24m + 12$. There are two unbalanced columns as we discuss in the following subcases, depending on the possible pair of the unbalanced columns.

1. If the pair has form $(\boldsymbol{u}_1, \boldsymbol{u}_4)$, then the matrix is of form $\mathbb{E}_1(x)\boldsymbol{u}_1\boldsymbol{u}_4\mathbb{E}_1(3m-x)$. In this case, we have $2m+1 \le d_H(\boldsymbol{c}_1, \boldsymbol{c}_2) \le 3m/2 + 1$, a contradiction.
2. If the pair has form $(\boldsymbol{u}_2, \boldsymbol{u})$ with $\boldsymbol{u} = (b, b, a, b)^T$ or $(a, a, b, a)^T$, then the matrix is of form $\mathbb{E}_2(x)\boldsymbol{u}_2\mathbb{E}_1(3m - x - z)\boldsymbol{u}\mathbb{E}_2(z)$. We have the following system:
$$\begin{cases} 2m+1 \le d_H(\boldsymbol{c}_1, \boldsymbol{c}_4) \le x + \lceil (3m - x - z)/2 \rceil + z, \\ 2m+1 \le d_H(\boldsymbol{c}_1, \boldsymbol{c}_3) = 3m - x - z + 1. \end{cases}$$
However, there are no solutions to this system.
3. If the pair has form $(\boldsymbol{u}_2, \boldsymbol{u})$ with $\boldsymbol{u} = (a, b, b, b)^T$ or $(b, a, a, a)^T$, then the matrix has form $\mathbb{E}_2(3m-x)\boldsymbol{u}_2\mathbb{E}_1(x)\boldsymbol{u}$ and x is even. In this case, we have the following system, which has no solutions:
$$\begin{cases} 2m+1 \le d_H(\boldsymbol{c}_1, \boldsymbol{c}_3) = x + 1, \\ 2m+1 \le d_H(\boldsymbol{c}_3, \boldsymbol{c}_4) = 3m - x + x/2. \end{cases}$$
4. If the pair has form $(\boldsymbol{u}_3, \boldsymbol{u})$ with with $\boldsymbol{u} = (b, a, b, b)^T$ or $(a, b, a, a)^T$, then the matrix has form $\mathbb{E}_1(x)\boldsymbol{u}_3\mathbb{E}_2(3m - x - z)\boldsymbol{u}\mathbb{E}_1(z)$. In this case, we have the following system, which has no solutions:
$$\begin{cases} 2m+1 \le d_H(\boldsymbol{c}_1, \boldsymbol{c}_4) = \lfloor x/2 \rfloor + 3m - x - z + \lfloor z/2 \rfloor, \\ 2m+1 \le d_H(\boldsymbol{c}_1, \boldsymbol{c}_3) = x + z + 1. \end{cases}$$
5. If the pair has form $(\boldsymbol{u}_4, \boldsymbol{u})$ with $\boldsymbol{u} = (a, b, b, b)^T$ or $(b, a, a, a)^T$, then the matrix has form $\boldsymbol{u}_4\mathbb{E}_1(3m)\boldsymbol{u}$ and we have that $2m+1 \le d_H(\boldsymbol{c}_1, \boldsymbol{c}_2) \le 3m/2$, a contradiction.
6. If the pair has form $(\boldsymbol{u}_4, \boldsymbol{u})$ with $\boldsymbol{u} = (b, b, a, b)^T$ or $(a, a, b, a)^T$, then the matrix has form $\boldsymbol{u}_4\mathbb{E}_1(x)\boldsymbol{u}\mathbb{E}_2(3m-x)$ and x is even. In this case, we have the following system:
$$\begin{cases} 2m+1 \le d_H(\boldsymbol{c}_1, \boldsymbol{c}_2) = x/2 + 3m - x, \\ 2m+1 \le d_H(\boldsymbol{c}_1, \boldsymbol{c}_3) = x + 1. \end{cases}$$
However, there is no solutions to this system.

TABLE 2
Some values of $P(n,d)$ for $2 \le d \le n \le 13$.

n \ d	2	3	4	5	6	7	8	9	10	11	12	13
2	3											
3	5	4										
4	7	5	5									
5	14	7	6	6								
6	23	12	8	7	7							
7	41	20	13	8	8	8						
8	74	25	15	10	9	9	9					
9	≥ 137	39	19	13	11	10	10	10				
10	≥ 220	71	31	20	14	11	11	11	11			
11	≥ 324	≥ 137	41	32	21	13	12	12	12	12		
12	≥ 598	≥ 141	73	37	23	15	14	13	13	13	13	
13				43	38	19	16	14	14	14	14	14

So far we have shown that $P(3m+2, 2m+1) \le 3m+4$. Note that $(100)^m(1001)$ is a $(3m+2, 2m+1)$-RPS of length $3m+4$. The conclusion follows. ∎

By exhaustive search, some values of $P(n,d)$ for $2 \le d \le n \le 13$ are determined (see Table 2). The corresponding optimal RPSs can be found in the appendix.

8. Asymptotically optimal positioning sequences of distance $n-1$. In this section, we study positioning sequences with large distance. Here, we require the Singleton bound.

PROPOSITION 8.1 (Singleton bound). *For all n, d, and q, we have that $P_q(n,d) \le q^{n-d+1} + n - 1$.*

We aim to construct sequences with length close to the bound above. Letting $n = qt+s$ with $s \in [\![q]\!]$, it is easy to check that the sequence $(012\cdots(q-1))^{t+1}01\cdots(s-1)$ is an $(n,n)_q$-RPS of length $q+n-1$. So for all n and q, we have that

$$P_q(n,n) = q + n - 1.$$

Now we turn to the case of $d = n-1$. We focus on cyclic sequences with the robust positioning ability. Formally, a cyclic sequence $\boldsymbol{s}$ is a *q-ary cyclic positioning sequence* of *strength* n and *distance* d if for any $0 \le i < j < N$, $d_H(\boldsymbol{s}[i, i+n-1], \boldsymbol{s}[j, j+n-1]) \ge d$. We denote such a sequence as $(n,d)_q$-CRPS. The maximum length of an $(n,d)_q$-CRPS is denoted by $P_q^\circ(n,d)$. Obviously, an $(n,d)_q$-CRPS is also an $(n,d)_q$-RPS; furthermore, we can obtain a slightly longer $(n,d)_q$-RPS from the $(n,d)_q$-CRPS.

PROPOSITION 8.2. $P_q(n,d) \ge P_q^\circ(n,d) + n - 1$.

Proof. Let $\boldsymbol{s}$ be an $(n,d)_q$-CRPS. Then the concatenation $\boldsymbol{ss}[0, n-2]$ is an $(n,d)_q$-RPS. ∎

Let $\boldsymbol{s}$ be a sequence of length N over Σ. We say an ordered pair $(a,b) \in \Sigma^2$ *appears in* $\boldsymbol{s}$ if $\boldsymbol{s}[i] = a$ and $\boldsymbol{s}[j] = b$ for some $i, j \in [\![N]\!]$; furthermore, if $j \equiv i + \delta \pmod N$ for some $\delta \in [\![N]\!]$, we say the pair (a,b) *appears in $\boldsymbol{s}$ with distance δ*. We have the following characterization for the $(n, n-1)_q$-CRPS, the proof of which is straightforward and we omit here.

PROPOSITION 8.3. *A sequence $\boldsymbol{s}$ with alphabet Σ is an $(n, n-1)_{|\Sigma|}$-CRPS if and only if for each $\delta \in [1, n-1]$, every ordered pair $(a,b) \in \Sigma^2$ appears in $\boldsymbol{s}$ with distance δ at most once.*

We borrow the idea of [12] to construct $(n, n-1)_q$-CRPSs.

Construction 5. Let p and r be two primes such that $p, r > n$ and $r^2 \geq p-1$. For each $d \in \mathbb{F}_p \backslash \{0\}$, construct a sequence $\boldsymbol{c}_d$ over $\mathbb{F}_p$ as

$$\boldsymbol{c}_d = (d, 2d, \ldots, (p-1)d).$$

Denote $E = [\![n]\!] \times \mathbb{F}_r$. For each $(a,b) \in \mathbb{F}_r^2$, construct a sequence $\boldsymbol{s}_{a,b}$ over E as

$$\boldsymbol{s}_{a,b} = ((0,b), (1, a+b), \ldots, (n-1, (n-1)a+b)).$$

Since $r^2 \geq p-1$, take $p-1$ sequences from the collection of $\boldsymbol{s}_{a,b}$'s and relabel them as $\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_{p-1}$.

Let $\Sigma = \mathbb{F}_p \cup E$ and construct a sequence $\boldsymbol{C}$ over Σ as

$$\boldsymbol{C} = \boldsymbol{c}_1 \boldsymbol{s}_1 \boldsymbol{c}_2 \boldsymbol{s}_2 \cdots \boldsymbol{c}_{p-1} \boldsymbol{s}_{p-1}.$$

THEOREM 8.4. *The sequence $\boldsymbol{C}$ is an $(n, n-1)_q$-CRPS with $q = p + nr$.*

Proof. According to Proposition 8.3, we only need to show that for any $1 \leq \delta \leq n-1$, every ordered pair in Σ^2 appears in $\boldsymbol{C}$ with distance δ at most once.

Case 1. We first consider the pairs in $\mathbb{F}_p^2$. Suppose that the pair $(\alpha, \beta) \in \mathbb{F}_p^2$ appears in $\boldsymbol{C}$ with distance δ twice. Since each $\boldsymbol{s}_{a,b}$ is a length-n sequence over E and $E \cap \mathbb{F}_p = \emptyset$, (α, β) must appear in the subsequences $\boldsymbol{c}_d$ and $\boldsymbol{c}_{d'}$ for some $d, d' \in [1, p-1]$. Then we may assume that $id = i'd' = \alpha$ and $(i+\delta)d = (i'+\delta)d' = \beta$, where $i, i' \in [1, p-1]$ and $i \neq i'$ if $d = d'$. It follows that $\beta - \alpha = \delta d = \delta d'$. Noting that $0 < \delta < n < p$, we have $d = d'$. This in turn implies that $i = i'$ since $id = i'd'$, a contradiction. Thus for each $1 \leq \delta \leq n-1$, every pair in $\mathbb{F}_p^2$ appears in $\boldsymbol{C}$ with distance δ at most once.

Case 2. Then we consider the pairs in E^2. Suppose that the pair $((i,\alpha),(j,\beta)) \in E^2$ appears in $\boldsymbol{C}$ with distance δ twice. Since each $\boldsymbol{c}_d$ is a length-$(p-1)$ sequence over $\mathbb{F}_p$ with $p-1 \geq n$ and $E \cap F_p = \emptyset$, we may assume that $j = i + \delta$, $ia + b = ia' + b' = \alpha$, and $ja + b = ja' + b' = \beta$, where $a, a', b, b' \in \mathbb{F}_r$ and $(a,b) \neq (a', b')$. It follows that $\beta - \alpha = \delta a = \delta a'$ and then $a = a'$. This leads to $b = b'$ as $ia + b = ia' + b' = \alpha$, a contradiction. Thus for each $1 \leq \delta \leq n-1$, every pair in E^2 appears in $\boldsymbol{C}$ with distance δ at most once.

Case 3. For the pair $(\alpha, (i,\beta))$ in $F_p \times E$, suppose that it appears in $\boldsymbol{C}$ with distance δ twice for some $\delta < n$, then it appears in the subsequences $\boldsymbol{c}_d \boldsymbol{s}_d$ and $\boldsymbol{c}_{d'} \boldsymbol{s}_{d'}$ for some $d, d' \in [1, p-1]$. Since the symbol (i, β) can only appear at the ith position of $\boldsymbol{s}_d$ and $\boldsymbol{s}_{d'}$, the two sequences $\boldsymbol{c}_d$ and $\boldsymbol{c}_{d'}$ must have the same symbol α at some position ℓ, i.e., $\ell d = \ell d'$, for some $\ell \in [1, p-1]$. It follows that $d = d'$, a contradiction.

Case 4. For the pair $((i,\alpha), \beta)$ in $E \times F_p$, with the same argument as in Case 3, we can show that it appears in $\boldsymbol{C}$ with distance δ at most once. ∎

COROLLARY 8.5. *Let $n = \lfloor cq^\alpha \rfloor$, where c and α are real numbers such that $c > 0$ and $0 \leq \alpha < \frac{1}{2}$. Then*

$$P_q^\circ(n, n-1) \geq q^2 - o(q^2).$$

Proof. Set $x = (\sqrt{q+n^2} - n)^2$. Then $x = q - 2n(\sqrt{q+n^2} - n) = q - O(q^{\frac{1}{2}+\alpha})$. According to Lemma 4.1, for sufficiently large q, we can choose a prime p such that $x - x^\theta \leq p \leq x$. Furthermore, according to Bertrand's postulate, we can choose a prime r such that $p - 1 \leq r^2 \leq 4p$. Then $p = \Theta(q)$ and $r = \Theta(p^{1/2}) = \Theta(q^{1/2})$. So we have $p, r^2 > n$ and we can apply Construction 5 to obtain an $(n, n-1)_{p+nr}$-CRPS, where

$$p + nr \leq p + 2n\sqrt{p} \leq \left(\sqrt{q+n^2} - n\right)^2 + 2n\left(\sqrt{q+n^2} - n\right) = q. \tag{8.1}$$

The length of $\boldsymbol{C}$ is

$$\begin{aligned}(p-1)(p-1+n) &\geq p^2-2p=(x-x^\theta)^2-2x\geq x^2-O(x^{1+\theta})\\ &\geq q^2-O(q^{\frac{3}{2}+\alpha})-O(q^{1+\theta})\geq q^2-o(q^2).\end{aligned}$$ ∎

COROLLARY 8.6. *Let $q=\Omega(n^{2+\epsilon})$ for some $\epsilon>0$. Then*

$$q^2+n-1-o(q^2)\leq P_q(n,n-1)\leq q^2+n-1.$$

9. Conclusion. We construct binary positioning patterns, equipped with efficient locating algorithms, that are robust to a constant number of errors. Our strategy is based on d-auto-cyclic vectors, Reed–Solomon codes, and Gray codes, and we reduce the number of redundancies as compared to previous constructions. In the locating algorithms, the d-auto-cyclic vectors are used as markers to locate the relative position. This information, together with the property of Gray codes, allows one to leverage the well-known fast decoding of Reed–Solomon codes to quickly identify the location.

Appendix. Here we list optimal (n,d)-RPSs for $n\leq 13$ and $2\leq d<\lfloor 2n/3\rfloor$, as well as $(n,d)\in\{(4,2),(7,4),(10,6)\}$.

$(4,2)$-RPS:
0001000
$(5,2)$-RPS:
00010111010001
$(6,2)$-RPS:
010011100100000101101010
$(6,3)$-RPS:
000101100010
$(7,2)$-RPS:
00101001101011001000111011100001011111010
$(7,3)$-RPS:
000010011111011000010
$(7,4)$-RPS:
0001011000101
$(8,2)$-RPS:
00100101010010011001000000100011010001011110101110110111000011100111110010
$(8,3)$-RPS:
000100011110111010010100
$(8,4)$-RPS:
000010110000101
$(9,3)$-RPS:
000001000111010100101111001101100000100
$(9,4)$-RPS:
000100101110001001 0
$(9,5)$-RPS:
0001101001110
$(10,3)$-RPS:
000000100011011010110011110001011111101110010010100110000111010000000100
$(10,4)$-RPS:
000010010001111011011000010010
$(10,5)$-RPS:
0001001011100010 0101
$(10,6)$-RPS:

00011010110001
$(11, 4)$-RPS:
00000100110001111001010110111010000010011
$(11, 5)$-RPS:
00001001000111101101110000100100
$(11, 6)$-RPS:
00010010111000 1001011
$(12, 4)$-RPS:
000000100011011010110011110001011111101110010010100110000111010000001000 1
$(12, 5)$-RPS:
000001010110011111010100110000010 1011
$(12, 6)$-RPS:
000001101011000000110101
$(12, 7)$-RPS:
000101001100111
$(13, 5)$-RPS:
0000010011000111100101011011101000001001100
$(13, 6)$-RPS:
000001010110011111010100110000010 10110
$(13, 7)$-RPS:
0001011000101100010

REFERENCES

[1] R. C. Baker, G. Harman, and J. Pintz, *The difference between consecutive primes,* II, Proc. Lond. Math. Soc., 83 (2001), pp. 532–562.

[2] R. Berkowitz and S. Kopparty, *Robust positioning patterns*, in Proceedings of the ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, 2016, pp. 1937–1951.

[3] A. M. Bruckstein, T. Etzion, R. Giryes, N. Gordon, R. J. Holt, and D. Shuldiner, *Simple and robust binary self-location patterns*, IEEE Trans. Inform. Theory, 58 (2012), pp. 4884–4889.

[4] J. Dai and C.-K. R. Chung, *Touchscreen everywhere: On transferring a normal planar surface to a touch-sensitive display*, IEEE Trans. Cybernet., 44 (2014), pp. 1383–1396.

[5] T. Etzion, *Constructions for perfect maps and pseudorandom arrays*, IEEE Trans. Inform. Theory, 34 (1988), pp. 1308–1316.

[6] E. M. Gabidulin, *Theory of codes with maximum rank distance*, Problemy Peredachi Informatsii, 21 (1985), pp. 3–16.

[7] J. Geng, *Structured-light* 3*D surface imaging: A tutorial*, Adv. Optics Photonics, 3 (2011), pp. 128–160.

[8] D.-J. Guan, *Generalized Gray Codes with Applications*, in Proc. Natl. Sci. Counc. Repub. China A, 22 (1998), pp. 841–848.

[9] M. Hagita, M. Matsumoto, F. Natsu, and Y. Ohtsuka, *Error correcting sequence and projective de Bruijn graph*, Graphs Combin., 24 (2008), pp. 185–194.

[10] P. V. Kumar and V. K. Wei, *Minimum distance of logarithmic and fractional partial m-sequences*, IEEE Trans. Inform. Theory, 38 (1992), pp. 1474–1482.

[11] M. Levy and E. Yaakobi, *Mutually uncorrelated codes for DNA storage*, IEEE Trans. Inform. Theory, 65 (2019), pp. 3671–3691, https://doi.org/10.1109/TIT.2018.2873138.

[12] Z. Lonc and M. Truszczyński, *Packing analogue of k-radius sequences*, European J. Combin., 57 (2016), pp. 57–70.

[13] F. J. MacWilliams and N. J. Sloane, *Pseudo-random sequences and arrays*, Proc. IEEE, 64 (1976), pp. 1715–1729.

[14] C. J. Mitchell, T. Etzion, and K. G. Paterson, *A method for constructing decodable de Bruijn sequences*, IEEE Trans. Inform. Theory, 42 (1996), pp. 1472–1478.

[15] K. G. Paterson, *Perfect maps*, IEEE Trans. Inform. Theory, 40 (1994), pp. 743–753.

[16] R. M. Roth, *Maximum-rank array codes and their application to crisscross error correction*, IEEE Trans. Inform. Theory, 37 (1991), pp. 328–336.

[17] E. R. Scheinerman, *Determining planar location via complement-free de Brujin sequences using discrete optical sensors*, IEEE Trans. Robot. Automat., 17 (2001), pp. 883–889.

[18] T. Simonite, *Microsoft Mulls a Stylus for Any Screen*, MIT Technology Review, 2012, http://www.technologyreview.com/news/428521/microsoftmulls-a-stylus-for-any-screen/.

[19] I. Szentandrasi, M. Zachariáš, J. Havel, A. Herout, M. Dubska, and R. Kajan, *Uniform marker fields: Camera localization by orientable de Bruijn Tori*, in Proceedings of the IEEE International Symposium on Mixed and Augmented Reality, 2012, pp. 319–320.

[20] L. R. Welch and E. R. Berlekamp, *Error Correction for Algebraic Block Codes*, US Patent 4,633,470, 1986.

[21] S. H. T. Yazdi, H. M. Kiah, R. Gabrys, and O. Milenkovic, *Mutually uncorrelated primers for DNA-based data storage*, IEEE Trans. Inform. Theory, 66 (2018), pp. 6283–6296.