Generalizations of All-or-Nothing Transforms and their Application in Secure Distributed Storage

by

Navid Nasr Esfahani

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2021

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:	Dr. Lucia Moura Professor, School of Electrical Engineering and Computer Science, University of Ottawa
Supervisor:	Dr. Douglas R. Stinson Professor, Cheriton School of Computer Science, University of Waterloo
Internal Member:	Dr. Ian Goldberg Professor, Cheriton School of Computer Science, University of Waterloo
Internal Member:	Dr. Alfred Menezes Professor, Cheriton School of Computer Science and Department of Combinatorics and Optimization, University of Waterloo
Internal-External Member:	Dr. David Jao Professor, Department of Combinatorics and Optimization, University of Waterloo

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

This thesis was researched and written under the supervision of Professor Douglas Stinson. Content from Chapters 2, 3 is from joint work with Professor Douglas Stinson, Professor Ian Goldberg, Dr. Paolo D'Arco, and Masoumeh Shafieinejad, the results of which are published in four papers [14, 29, 31, 39] and a technical report [30].

Abstract

An all-or-nothing transform is an invertible function that maps s inputs to s outputs such that, in the calculation of the inverse, the absence of only one output makes it impossible for an adversary to obtain any information about any single input. In this thesis, we generalize this structure in several ways motivated by different applications, and for each generalization, we provide some constructions. For a particular generalization, where we consider the security of t input blocks in the absence of t output blocks, namely, t-all-or-nothing transforms, we provide two applications. We also define a closeness measure and study structures that are close to t-all-or-nothing transforms. Other generalizations consider the situations where:

i) t covers a range of values and the structure maintains its t-all-or-nothingness property for all values of t in that range;

ii) the transform provides security for a smaller, yet fixed, number of inputs than the number of absent outputs;

iii) the missing output blocks are only from a fixed subset of the output blocks; and

iv) the transform generates n outputs so that it can still reconstruct the inputs as long as s outputs are available.

In the last case, the absence of $n - s + t$ outputs can protect the security of any t inputs. For each of these transforms, various existence and non-existence results, as well as bounds and equivalence results are presented. We finish with proposing an application of generalization (*iv*) in secure distributed storage.

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my supervisor, Professor Douglas Stinson, for his continuous support, guidance, patience, and generosity. Never have I left a meeting with Doug without a smile of positivity and determination on my face, and I cannot thank him enough just for that, let alone for his teachings and training. It is a great honour and was a unique opportunity for me to be his PhD student.

I would like to extend my gratitude to Professor Ian Goldberg for his guidance through constructive feedback and stimulating discussions throughout the past few years, and also for using every opportunity to teach us something interesting. I would also like to thank Professor Alfred Menezes for his thought-provoking comments and questions.

I would like to sincerely thank Professor Lucia Moura and Professor David Jao for their time, invaluable comments, and suggestions.

I wish to thank all members of CrySP Lab for the great conversations and for the amazing times. The reading groups, joint work, and discussions with Cecylia, Justin, Bailey, Masi, Shannon, Erinn, and Tao will stay with me as some of the most interesting and joyful memories from my years at CrySP.

Special thanks to Shervin, Parisa, Nazanin, and Keyvan, who have been my family in Canada, if one could choose their family; for they laughed with me when I was happy and supported me I was down. I would also like to thank my dear friends Ala, Shima, Kaveh, and Mehdi for the great times we had together learning and discussing interesting topics, and to thank Matin, Peyman, and Hengameh for the wonderful times.

Finally, and most importantly, I am deeply grateful to my parents, Parvaneh and Abbasali; thank you for giving me the opportunity to follow my dreams and supporting me all the way through my endeavors! My lovely brother and sister, Farid and Parinaz, talking to you has always brought me joy. And thank you all, including Shadi, Erfan, and Reihaneh for your endless love and for your support in every decision I made. Without your support, this would not be possible.

Dedication

To family, and to friends,

00001010 00001001 00001001 01110100 01101111 00100000 01110100 01101000 01101111
01110011 01100101 00100000 01110111 01101001 01110100 01101000 00100000 01110111
01101000 01101111 01101101 00100000 01110100 01101000 01101001 01110011 00100000
01110010 01101111 01100001 01100100 00100000 01100101 01101110 01100100 01110011
00001010 00001001 00001001 00001001 00001001 01110100 01101111 00100000 01100110
01110010 01100101 01100101 01100100 01101111 01101101 00100000 01100001 01101110
01100100 00100000 01101010 01101111 01111001 00101100 00100000 01100001 01101110
01100100 00100000 01110100 01101111 00100000 01110100 01101000 01101111 01110011
01100101 00100000 01110100 01101111 00100000 01110111 01101000 01101111 01101101
00101100 00001010 00001001 00001001 00001001 00001001 00001001 00001001 01110100
01101000 01101001 01110011 00100000 01110111 01101111 01110010 01100100 00100000
01100001 00100000 01100100 01100101 01110011 01101001 01110010 01100001 01100010
01101100 01100101 00100000 01101101 01100101 01110011 01110011 01100001 01100111
01100101 00100000 01110011 01100101 01101110 01100100 01110011 00101110

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Definitions

All-or-nothing transforms (AONTs) and their applications are the main focus of this thesis. Hence, it is appropriate to start with a definition of an AONT; however, some terminology needs to be established first.

Block ciphers are a family of ciphers that operate only on blocks of a fixed length, ℓ. To use a block cipher on a message M of length m, the message needs to be formatted into a sequence, called the *message sequence*, of s blocks of size ℓ, i.e., $m_1, m_2, \ldots, m_s$. If m is not a multiple of ℓ, the message will be padded with symbols, to the next multiple of ℓ. Each of the m_i's, $i \in \{1, \ldots, s\}$, is called a *message block*. To prevent identical blocks either in one message or in two different messages being encrypted to identical ciphertext blocks, a message sequence could be transformed to an intermediate sequence of blocks, called the *pseudo-message sequence*, prior to encryption. A pseudo-message consists of s' *pseudo-message blocks* of length ℓ', i.e., $m'_1, m'_2, \ldots, m'_{s'}$. In this setting, Rivest [38] defines all-or-nothing transforms as follows.

Definition 1.1.1. *[38] A mapping f from message sequences of s blocks to pseudo-message sequences of s' blocks, say from $m_1, m_2, \ldots, m_s$ to $m'_1, m'_2, \ldots, m'_{s'}$, is an* all-or-nothing transform *if it satisfies the following conditions:*

- *f is invertible, i.e., f^{-1} exists and maps the pseudo-message sequence to the message sequence,*
- *both f and f^{-1} are efficiently computable, and*

- *if any of the pseudo-message blocks is missing, computing any function of any message block is computationally infeasible.*

An AONT can be *deterministic*, i.e., a message sequence is always mapped to the same pseudo-message sequence, or *randomized*, i.e., each time the AONT is applied, the message sequence is randomly mapped to a pseudo-message sequence from a set of pseudo-message sequences.

Example 1.1.1 presents a variation of the *package transform*, introduced by Rivest [38], as an AONT.

Example 1.1.1. *Suppose $M = (m_1, m_2, \ldots, m_s)$ is the message, K_{PT} is a random encryption key, $E_K(\cdot)$ is a block cipher with key K, and $h(\cdot)$ is a cryptographic hash function. For every message block m_i, a pseudo-message block m'_i, $i \in \{1, 2, \ldots, s\}$ is calculated as*

$$m'_i = m_i \oplus E_{K_{PT}}(i).$$

Then the hash x_i of each pseudo-message block is computed:

$$x_i = h(m'_i \oplus i), \ i \in \{1, 2, \ldots, s\}.$$

After computing the x_i values, the last pseudo-message block, m'_{s+1}, is computed by XORing all s of x_i values and the key, K_{PT}:

$$m'_{s+1} = K_{PT} \oplus \bigoplus_{i=1}^{s} x_i.$$

It is easy to informally verify the AONT conditions for the package transform:

- Given the first s pseudo-message blocks, it is possible to compute x_i's, and then XOR all the x_i values with the last pseudo-message block to get the key, and finally, use the key to obtain the message blocks.
- Both the package transform and its inverse are efficiently computable.
- If any of the m'_i's is missing, $x_i = h(m'_i \oplus i)$ cannot be calculated. Therefore, the key cannot be extracted, and it is impossible to compute any function of any message block.

After applying the AONT, all the intermediate values, i.e., the pseudo-message blocks, are encrypted using the original encryption key K. Each block of the ciphertext is the encryption of a pseudo-message block, i.e., $y_i = E_K(m'_i)$, $i \in \{1, 2, \ldots, s+1\}$, and $y_1, y_2, y_3, \ldots, y_s, y_{s+1}$ are sent to the receiver.

It should be noted that the unkeyed hash function could be substituted by any one-way function.

As we previously mentioned, a block cipher can encrypt only plaintexts of a fixed length, denoted by ℓ. Thus, a message needs to be broken into blocks of length ℓ. A mode of operation is the method of linking these blocks. In *electronic codebook (ECB)* mode of operation, each block is encrypted independent of the other blocks. Hence, all identical blocks will be encrypted to identical ciphertexts if a deterministic encryption method is used. This property may be helpful to the adversary if they have observed the plaintext for some of the ciphertext blocks or if the plaintext has a low diversity of plaintext blocks. Therefore, encrypting each block individually can be an unsafe practice, and other modes of operation are used to link the message blocks in order to avoid such issues. The problem with some modes of operation, e.g., *cipher block chaining (CBC)* and *counter (CTR)* mode, is that decrypting one block of the ciphertext results in access to one block of the message, which in most cases is undesirable and is considered unwanted information leakage. AONTs were defined as a mode of operation by Rivest [38], originally in the computational security setting, i.e., an attacker is assumed to be limited by computational resources available and is not able to solve some instances of problems in a feasible time period. At the time of AONTs' introduction, brute-force attacks were mostly a big issue when there was a restriction on the key length due to either hardware limitations or US exporting regulations on cryptographic systems [15]. This constraint made it possible for an adversary to deploy a brute-force attack to find the encryption key. In other words, the original purpose of AONT was to provide a strong non-separable mode of operation such that to learn about "even one message block", one has to decrypt all the ciphertext blocks [38]. Rivest defines a strong non-separable mode of operation as the following.

Definition 1.1.2. *[38] A mode of encryption that transforms a sequence of message blocks*

$$m_1, m_2, \ldots, m_s$$

into a sequence of ciphertext blocks

$$c_1, c_2, \ldots, c_{s'},$$

for $s' \geq s$, is strongly non-separable *if obtaining even one message block requires decrypting*

all ciphertext blocks.

AONTs are instances of strongly non-separable modes of operation because the third AONT property is the condition for a mode of operation to be strongly non-separable, as mentioned above. Such property guarantees the necessity of decrypting all ciphertext blocks, which slows down brute-force attacks by a factor of the number of blocks.

Rivest [38] also provides various constructions with the AONT property. His work is followed up in different directions by different researchers [6, 8, 42]. The early instances of AONT will be discussed in detail along with the early extensions of Rivest's work on AONT in Section 1.4.

In addition to a mode of operation, an AONT can be considered as a secret-sharing scheme, i.e., a scheme that distributes shares of a secret among a group of participants such that only certain pre-defined subsets of shareholders are able to recover the secret. Examples of such schemes include Shamir's secret-sharing scheme [40], which maximizes security of the secret at the cost of storage efficiency, as well as Rabin's Information Dispersal Algorithm (IDA) [36], which sacrifices security to reduce the storage overhead. These schemes will be discussed in detail in Section 1.4.

Although some of the original motivations of using AONTs, e.g., government-enforced short keys, do not hold anymore, AONTs have been used by researchers for different purposes: anti-jamming techniques, location anonymization, network coding, secure distributed storage [9,17,35,48,52], to name a few. This wide range of applications motivated our research on AONTs in different levels, namely, defining AONTs with further properties, searching for instances of the new generalizations, analyzing their security, and using them in potential applications.

The focus of this research is solely on *unconditionally secure AONTs* (in which the adversary is given unlimited computational power), introduced by Stinson [42]. Hence, it is necessary to define unconditionally secure AONTs, prior to discussing our contribution to the topic. We first provide an informal introduction to unconditionally secure AONTs, and present the formal definition in Section 1.6.

Let Σ be a finite set, let s be a positive integer, and let ϕ be a mapping from Σ^s to Σ^s, i.e., $(y_1, y_2, \ldots, y_s) = \phi(x_1, x_2, \ldots, x_s)$ for $(x_1, x_2, \ldots, x_s) \in \Sigma^s$ and $(y_1, y_2, \ldots, y_s) \in \Sigma^s$. Then ϕ is an unconditionally secure AONT if it satisfies the following conditions:

1. ϕ is a bijection.
2. If any $s - 1$ of the output elements are fixed, then any one input element can take on all possible values in Σ.

Table 1.1: An example of an AONT from two input blocks x_1 and x_2 to two output blocks y_1 and y_2.

x_1	x_2	y_1	y_2
a	a	c	b
a	b	b	c
a	c	a	a
b	a	a	c
b	b	c	a
b	c	b	b
c	a	b	a
c	b	a	b
c	c	c	c

Note that, Σ can be equal to a finite field $\mathbb{F}_q$, in particular $\mathbb{F}_{2^\ell}$, where $\ell \geq 1$, in the binary setting.

Based on this definition, from this point on, we will assume each message has s blocks, unless it is stated otherwise. In other words, we will assume that the number of input blocks is equal to the number of output blocks.

Example 1.1.2. *The Table 1.1 shows an AONT over the alphabet $\Sigma = \{a, b, c\}$ and for $s = 2$, in table format. This AONT maps every pair of input elements to a pair of output elements, and it can be observed that for any fixed value of either y_1 or y_2, all the symbols for either x_1 or x_2 are distributed uniformly.*

1.2 Contributions

In this thesis, we study generalizations of unconditionally secure AONTs, in terms of existence, constructions, bounds, and their security properties, we show how these structures relate to other schemes, and we discuss the applications of some of these structures in secure distributed storage, in order to demonstrate the accuracy of the following thesis statement:

Generalizations of unconditionally secure all-or-nothing transforms and structures close to them that facilitate more flexible parameters in diverse applications exist.

1.3 Organization

The rest of this thesis is organized as follows. In the remaining sections of Chapter 1, background information on early instances, extensions and analysis of AONTs, and the applications of the AONTs will be discussed. In Chapter 2, we present the results of our research about theoretical aspects of AONTs. These results consist of the generalization of unconditionally secure AONTs to t-AONTs, as well as existence conditions and constructions for specific values of t, i.e., $t = 2, 3$. Chapter 3 covers the study of close to AONT structures, known as *almost AONTs*. Next, five more generalizations of AONTs, i.e., *range AONTs*, *strong AONTs*, *asymmetric AONTs*, *restricted AONTs*, and *rectangular AONTs* are presented in Chapter 4. Finally, Chapter 5 provides a summary of the previous chapters as well as some open problems for future research.

1.4 Mathematical Background

In this section, we will cover the basic concepts needed for the following chapters. The section begins with a brief overview of some cryptographic schemes and combinatorial structures that are used in relation to AONTs in this research: Shamir's secret sharing scheme, ramp schemes, information dispersal schemes, resilient functions, orthogonal arrays, split orthogonal arrays, mutually orthogonal Latin squares, transversal designs, balanced incomplete block designs, and codes. The rest of this chapter will then discuss some examples of AONTs provided by Rivest [38], the early theoretical studies of AONTs, and some applications of AONTs.

1.4.1 Cryptography

1.4.1.1 Shamir's Secret Sharing Scheme

A (t, n)-*threshold scheme* is a method of breaking a secret into n shares, such that:

1. any t-subset of the shares can be used to recover the secret,
2. any subset of fewer than t shares cannot reveal any information about the secret.

In particular, if any subset of shares, from a threshold scheme, that cannot recover the secret does not yield any information about the secret, then the threshold scheme is

perfect [7]. For a perfect threshold scheme, if the set of possible shares and the set of possible secrets are of the same cardinality, then the threshold scheme is *ideal* [7].

In this chapter, "secret" is used in a broad context. Based on the capabilities of the scheme and the application it can be a password, decryption key, a document, a photo, or any other type of file.

Shamir's secret sharing scheme (SSSS) [40] is an instance of a such scheme. This scheme relies on polynomial interpolation and the fact that a polynomial of degree $t-1$ over a finite field can be determined by any t distinct points on it, and any fewer number of points are on the same number of polynomials. The following is the construction of a (t, n)-secret sharing scheme given by Shamir [40].

Construction 1.4.1. *Let $\sigma \in \mathbb{F}_q$ be the secret, and let $a_1, a_2, \ldots, a_{t-1}$ be $t-1$ random elements from a finite field $\mathbb{F}_q$, where $q \geq n+1$. Form a polynomial $\mathcal{P}(x)$ as follows:*

$$\mathcal{P}(x) = \sigma + \sum_{i=1}^{t-1} a_i x^i.$$

Now evaluate $\mathcal{P}(x)$ at n distinct non-zero points, $x_1, x_2, \ldots, x_n$, and provide the shareholder i with share $(x_i, \mathcal{P}(x_i))$.

To recover the secret, t shares are gathered, and using polynomial interpolation, the polynomial $\mathcal{P}(x)$ is reconstructed and evaluated at $x = 0$.

1.4.1.2 Rabin's Information Dispersal Algorithm

Rabin [36] pointed out that Shamir's secret sharing scheme requires all the shares to be at least the same length as the secret. This overhead can be negligible for small secrets, but for larger ones, it imposes a great storage cost on the shareholders. Rabin [36] then introduced information dispersal algorithm (IDA) as a method of efficiently dispersing a file (not necessarily a secret) over multiple servers. In this construction, the file is represented as a vector of length m and with elements from $\mathbb{F}_q$. The message is then divided into blocks of length s_1. To share each block of length s_1, it is multiplied to an s_1 by s_2 matrix C, where $s_1 \leq s_2$, which is defined over $\mathbb{F}_q$, where any s_1 columns of C are linearly independent. To guarantee this property in the matrix, Rabin [36] used Cauchy matrices. While the given definition does not provide any level of security, in Chapter 2 we will discuss the security of Cauchy matrices as an AONT.

Based on the number of servers and the number of shares needed to reconstruct the message, the overhead of this scheme can vary, but the ratio of total storage to original document size remains close to 1 [36].

1.4.1.3 Ramp Schemes

Sometimes it is desired to have two different thresholds: one to protect the confidentiality of the secret and another to guarantee its availability. For this purpose, an (ℓ, t, n)-ramp scheme is defined as follows.

Definition 1.4.1. *An* (ℓ, t, n)-ramp scheme *breaks the secret into* n *shares, such that availability of* u *shares leads to:*

1. *no information regarding the secret, if* $u \leq \ell$,
2. *complete reconstruction of the secret, if* $u \geq t$.

Figure 1.1 illustrates these properties. For any ℓ or fewer columns of the output, fixing the values on those coordinates does not reveal any information about the inputs; on the other hand, fixing any t or more of the output columns identifies the input uniquely .

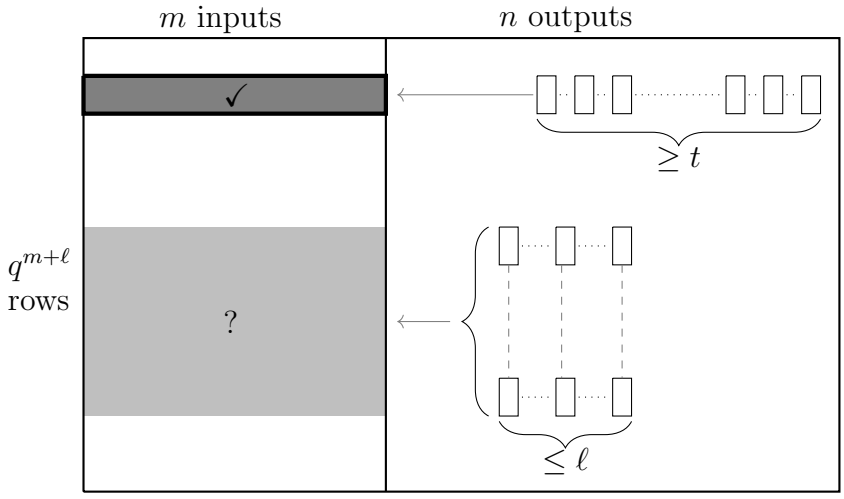


Figure 1.1: An (n, t, ℓ)-ramp scheme applied on secret of length m

Jackson and Martin [19] pointed out the fact that Rabin's IDA can be considered as a $(0, t, n)$-ramp scheme, where t is the number of shares needed to recover a secret in the IDA. Hence, having access to t shares is guaranteed to provide the secret, but there are no restrictions on the amount of information obtained from less than t blocks. Therefore, to protect the security of the secret against groups of users with fewer than ℓ shares, one can use an (ℓ, t, n)-ramp scheme to distribute the document among the users. In order to discuss the security of documents with respect to u shares, where $u < t$, it is useful to define *ideal ramp schemes*.

In an (ℓ, t, n)-ramp scheme with shares from $\mathbb{F}_q$, the number of possible secrets can be at most $q^{t-\ell}$ [44]. If the number of possible secrets in an (ℓ, t, n)-ramp scheme is equal to $q^{t-\ell}$ then the ramp scheme is an *ideal* ramp scheme [44]. This restriction in the definition of an ideal ramp-scheme translates to the following behavior of such a scheme. In an ideal (ℓ, t, n)-ramp scheme, the number of possible secrets given u shares is:

1. $q^{t-\ell}$ if $u \leq \ell$,
2. q^{t-u} if $\ell < u < t$, and
3. 1, if $t \leq u$.

1.4.1.4 Resilient Functions

Resilient functions are another type of cryptographic scheme that we need to introduce in this section. The properties of resilient functions have mostly been studied in the context of their relation to some combinatorial structures, for example, codes and orthogonal arrays. Therefore, we only define them here and leave additional discussion to later sections, after we learn about orthogonal arrays. Resilient functions were originally defined over a binary alphabet; however, a generalized definition over an alphabet $\mathcal{X}$ of size v is more appropriate in this thesis.

Definition 1.4.2. *Let s and n be positive integers and let $\mathcal{X}$ be a finite set of size $v \geq 2$. A function $f : \mathcal{X}^n \to \mathcal{X}^s$ is an* (n, s, t, v)-resilient function *(*(n, s, t, v)-RF *for short) if fixing any t of the n input symbols does not reveal any information about the output whenever the remaining inputs are chosen independently and uniformly at random.*

1.4.2 Combinatorics

1.4.2.1 Orthogonal Arrays

Let A be an N by k array with entries from an alphabet $\mathcal{X}$ of size v. We will refer to A as an (N, k, v)-*array*. Suppose the columns of A are labeled by the elements in the set $C = \{1, \ldots, k\}$. Let $D \subseteq C$, and define A_D to be the array obtained from A by deleting all the columns $c \notin D$. We say that A is *unbiased* with respect to D if the rows of A_D contain every $|D|$-tuple of elements of $\mathcal{X}$ exactly $N/v^{|D|}$ times.

An *orthogonal array*, denoted by $\mathrm{OA}_\lambda(s, k, v)$, is a $(\lambda v^s, k, v)$-array that is unbiased with respect to any subset of s columns. Therefore, any subset of s columns contains every s-tuple over $\mathcal{X}$ exactly λ times. Conventionally, the subscript λ is not written if $\lambda = 1$.

Example 1.4.1. *The array presented below is an orthogonal array* $OA(2, 3, 2)$.

a	a	a
a	b	b
b	a	b
b	b	a

In particular, if $\mathcal{X} = \mathbb{F}_q$ for a prime power q and the rows of A form a subspace of $(\mathbb{F}_q)^k$, the orthogonal array is a *linear orthogonal array* [43, p. 225].

Theorem 1.4.2. *The existence of an* $OA_\lambda(s, k, v)$ *implies the existence of an* $OA_{v\lambda}(s - 1, k, v)$.

Proof. For any $s - 1$ columns in the $\mathrm{OA}_\lambda(s, k, v)$, choose an arbitrary column from the remaining $k - s + 1$ columns to form a set of s columns. The $\mathrm{OA}_\lambda(s, k, v)$ is unbiased with respect to these s columns, and each s-tuple appears λ times. Thus, for each symbol in the added column, each $(s - 1)$-tuple appears λ times in the original $s - 1$ columns, so the $\mathrm{OA}_\lambda(s, k, v)$ is unbiased with respect to those $s - 1$ columns. Finally, since each s-tuple appeared λ times in those s columns and there are v symbols in the alphabet, each $(s - 1)$-tuple appears $v\lambda$ times in the $s - 1$ columns. □

Suppose $\lambda = v^r$ for some integer r. A *large set of* $OA_{v^r}(t, n, v)$ consists of v^{n-r-t} distinct $\mathrm{OA}_{v^r}(t, n, v)$'s, which together contain all v^n possible n-tuples exactly once. The following theorem by Stinson [41, Theorem 2.1] states that a resilient function is equivalent to a "large set" of orthogonal arrays.

Theorem 1.4.3. *[41] The existence of an (n, m, t, v)-resilient function is equivalent to the existence of a large set of $OA_{q^{n-m-t}}(t, n, v)$.*

Although the original theorem was stated for $v = 2$, the same argument can be used for any positive integer value of v.

For $s = 1, 2$, it is easy to find constructions of orthogonal arrays for different values of k [13, p. 113]; however, for $s \geq 3$, most existing constructions are derived from Reed-Solomon Codes (See Section 1.4.2.6).

1.4.2.2 Split Orthogonal Arrays

Levenshtein [25] defined *split orthogonal arrays* (SOAs) as follows. An $SOA(t_1, t_2; n_1, n_2; v)$ is a $(v^{t_1+t_2}, n_1 + n_2, v)$ array, say A, that satisfies the following properties:

1. the columns of A are partitioned into two sets, of sizes n_1 and n_2, respectively, and
2. A is unbiased with respect to any t_1+t_2 columns in which t_1 columns are chosen from the first set of columns and t_2 columns are chosen from the second set of columns.

Example 1.4.2. *The array presented below is an orthogonal array $SOA(1, 1; 1, 2; 2)$.*

a	a	a
a	b	b
b	a	b
b	b	a

1.4.2.3 Mutually Orthogonal Latin Squares

To discuss *mutually orthogonal Latin squares (MOLS)*, we first need to have the definition of *Latin squares* and *orthogonal Latin squares*.

Definition 1.4.3. *[43, p. 123] A* Latin square *of order v over a v-set $\mathcal{X}$ is a $v \times v$ array L in which every cell contains an element of $\mathcal{X}$, and each element of $\mathcal{X}$ appears exactly once in each row and each column of L.*

Definition 1.4.4. *[43, p. 131] Suppose L_1 and L_2 are two Latin squares of order v with entries from v-sets $\mathcal{X}$ and Y, respectively. L_1 and L_2 are* orthogonal Latin squares *if for every $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, there is a unique cell (i, j) such that $L_1(i, j) = x$ and $L_2(i, j) = y$.*

Example 1.4.3. *[43, p. 132] Consider the following Latin squares of order* 3, L_1 *and* L_2.

$$L_1 = \begin{array}{|c|c|c|}\hline a & b & c \\ \hline b & c & a \\ \hline c & a & b \\ \hline \end{array} \text{ and } L_2 = \begin{array}{|c|c|c|}\hline a & b & c \\ \hline c & a & b \\ \hline b & c & a \\ \hline \end{array}$$

To verify that they are orthogonal, we can construct a table with pairs of symbols as entries and check that each pair appears exactly once. In each pair, the first symbol is from the corresponding entry in L_1 *and the second one is from the corresponding entry in* L_2.

a, a	b, b	c, c
b, c	c, a	a, b
c, b	a, c	b, a

Definition 1.4.5. *[43, p. 136] Suppose* $L_1, L_2, \ldots, L_s$ *are Latin squares of order* v. *They are* mutually orthogonal Latin squares *if any pair of Latin squares* L_i *and* L_j, $1 \leq i < j \leq s$, *are orthogonal.*

A set of s mutually orthogonal Latin squares of order v is denoted by $s\,\mathrm{MOLS}(v)$.

Theorem 1.4.4. *[43, p. 140] The existence of* $s\,\mathrm{MOLS}(v)$ *is equivalent to the existence of an* $OA(2, s+2, v)$.

Many results, including constructions and existence conditions, on MOLS can be found in the Handbook of Combinatorial Designs [13] and Combinatorial Designs: Construction and Analysis [43]. One of these results is MacNeish's Theorem, stated below.

Theorem 1.4.5. *[43]* ***(MacNeish's Theorem.)*** *Suppose* $v = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_\ell^{\alpha_\ell}$, *where the* p_i*'s are distinct primes and* $\alpha_i \geq 1$ *for* $1 \leq i \leq \ell$. *Then there exist* $s\,\mathrm{MOLS}(v)$, *where*

$$s = \min_{i=1}^{\ell} \{p_i^{\alpha_i} - 1\}.$$

From Theorem 1.4.4 and MacNeish's Theorem, the following corollary can be concluded.

Corollary 1.4.6. *Suppose* $v = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_\ell^{\alpha_\ell}$, *where the* p_i*'s are distinct primes and* $\alpha_i \geq 1$ *for* $1 \leq i \leq \ell$. *Then there exists an* $OA(2, s, v)$, *where*

$$s = \min_{i=1}^{\ell} \{p_i^{\alpha_i} + 1\}.$$

1.4.2.4 Transversal Designs

In Chapter 2, we will discuss an application which uses a certain type of combinatorial structure, namely *transversal designs*. These combinatorial designs are closely related to orthogonal arrays and mutually orthogonal Latin squares. We will continue with the definition of transversal designs and then state the relationship between these three types of designs in a theorem.

Definition 1.4.6. *Let $k \geq s \geq 2$ and $v \geq 1$. A* transversal design s-TD(k, v) *is a triple $(\mathcal{X}, \mathcal{G}, \mathcal{B})$ such that the following properties are satisfied:*

- *$\mathcal{X}$ is a set of kv elements called* points*,*
- *$\mathcal{G}$ is a partition of $\mathcal{X}$ into k subsets of size v called* design groups*,*
- *$\mathcal{B}$ is a set of k-subsets of $\mathcal{X}$ called* blocks*,*
- *any design group and any block contain exactly one common point, and*
- *every s-tuple of points from s distinct design groups is contained in exactly one block.*

Theorem 1.4.7. *[43, 146] Suppose that $n \geq 2$ and $k \geq 3$. Then the existence of any one of the following designs implies the existence of the other two designs:*

1. *$k - 2$* MOLS(v)*,*
2. *an* $OA(2, k, v)$*,*
3. *a* 2-TD(k, v)*.*

Also, using the same formulation as the one used by Stinson [43], it can be shown that an s-TD(k, v) is equivalent to an OA(s, k, v).

1.4.2.5 Balanced Incomplete Block Designs

A combinatorial design consists of a set $\mathcal{X}$ and one or more collections of its subsets that satisfy certain "balance" conditions. *Balanced incomplete block designs* (BIBDs) are defined as follows.

Definition 1.4.7. *Let v, b, r, k, and λ be positive integers. Consider a set of points, called $\mathcal{X}$, of size v and a family of b k-subsets of $\mathcal{X}$, called* blocks*, denoted as $\mathcal{B} = \{B_i : i = 1, 2, \ldots, b\}$. The ordered pair $(\mathcal{X}, \mathcal{B})$ is a* balanced incomplete block design (BIBD) *if*

- *each element $x \in \mathcal{X}$ appears in exactly r blocks:*

$$\text{for all } x \in \mathcal{X}, |\{B_i : x \in B_i\}| = r,$$

- *each pair of distinct elements $x_1, x_2 \in \mathcal{X}$ occur together in exactly λ blocks:*

$$\text{for all } x_1, x_2 \in \mathcal{X}, x_1 \neq x_2, |\{B_i : \{x_1, x_2\} \subseteq B_i\}| = \lambda.$$

*A design with such parameters is called a (v, b, r, k, λ)-*BIBD*, or more concisely, a (v, k, λ)-*BIBD*, since b and r are determined from the values of v, k, and λ.*

Example 1.4.4. *Let $\mathcal{X} = \{a, b, c, d, e, f, g\}$ and $\mathcal{B} = \{B_1, B_2, B_3, B_4, B_5, B_6, B_7\}$, such that $B_1 = \{a, b, c\}, B_2 = \{a, d, e\}, B_3 = \{a, f, g\}, B_4 = \{b, d, g\}, B_5 = \{b, e, f\}, B_6 = \{c, e, g\}$,and $B_7 = \{c, d, f\}$. Then $(\mathcal{X}, \mathcal{B})$ is a $(7, 7, 3, 3, 1)$-BIBD, also known as the* Fano plane*. Figure 1.2 depicts a graphical representation of the design, where each black disc on a line (or circle) represents a point on a block.*

Besides explicitly listing the elements of $\mathcal{X}$ and $\mathcal{B}$, and the graphical representation, a BIBD can be represented using an *incidence matrix*.

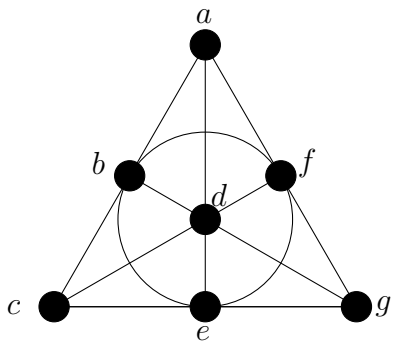


Figure 1.2: Fano plane

Definition 1.4.8. *The* incidence matrix $M = \{m_{ij}\}$ *of a* (v, k, λ)*-BIBD is a* $v \times b$ *matrix, where each entry* m_{ij} *is defined as follows:*

$$m_{ij} = \begin{cases} 1 & \text{if } x_i \in B_j \\ 0 & \text{if } x_i \notin B_j. \end{cases}$$

If M is the incidence matrix of a (v, b, r, k, λ)-BIBD, then there are exactly k 1's in each column of M and exactly r 1's in each row of M. Also, for each pair of rows r_i and $r_j, i \neq j$ in M, $r_i \cdot r_j = \lambda$, where "$r_i \cdot r_j$" denotes the inner product of r_i and r_j.

Example 1.4.5. *Let* $x_1 = a, x_2 = b, \cdots, x_7 = g$*. Then the following matrix* M *is the incidence matrix of the Fano plane in Example 1.4.4.*

$$M = \begin{pmatrix} 1&1&1&0&0&0&0 \\ 1&0&0&1&1&0&0 \\ 1&0&0&0&0&1&1 \\ 0&1&0&1&0&0&1 \\ 0&1&0&0&1&1&0 \\ 0&0&1&1&0&1&0 \\ 0&0&1&0&1&0&1 \end{pmatrix}$$

As mentioned above, each row of M *has three 1's and there are three 1's in each column of* M*. Also, for each distinct pair of rows, their inner product is equal to 1.*

The following lemma can be proven by counting the number of 1's in the incidence matrix of a (v, b, r, k, λ)-BIBD: once by counting the 1's in all the rows and again by counting the 1's in the columns.

Lemma 1.4.8. *In a* (v, b, r, k, λ)*-BIBD, the following equation holds:*

$$vr = bk.$$

Note that a point forms λ pairs with each of the other $v - 1$ points. This number can also be calculated by counting the number of pairs it forms in each of the k blocks where it appears. Therefore, we have the following.

Lemma 1.4.9. *In a* (v, b, r, k, λ)*-BIBD, the following equation holds:*

$$(v - 1)\lambda = r(k - 1).$$

Definition 1.4.9. *A (v, k, λ)-BIBD is a* symmetric balanced incomplete block design *(SBIBD) if $v = b$.*

Example 1.4.6. *In a Fano plane, we have $v = b = 7$, and therefore it is a $(7, 3, 1)$-SBIBD.*

Remark 1.4.1. *An SBIBD does not necessarily have a symmetric incidence matrix. Also, from Lemma 1.4.8, it can be concluded that $r = k$ in an SBIBD.*

Theorem 1.4.10. *Let M be the incidence matrix of a (v, k, λ)-SBIBD. Then, its complement, M^c, is the incidence matrix of a $(v, v - k, v - 2k + \lambda)$-SBIBD.*

For more instances and properties of BIBDs, see the Handbook of Combinatorial Designs [13] and Combinatorial Designs: Constructions and Analysis [43].

1.4.2.6 Coding Theory

Suppose $V_q(n)$ is the set of all n-tuples over an alphabet of size q, say Σ. Then an $[n, M]$ *q-ary code* is an M-subset of $V_q(n)$. Any n-tuple in $V_q(n)$ is a *word*, and any word that is a member of the code is a *codeword*.

Definition 1.4.10. *The* Hamming distance *of any two words is the number of coordinates in which they differ.*

In this thesis, *distance* between two codewords will refer to their Hamming distance, unless stated otherwise.

Example 1.4.7. *Consider the following words over $V_2(7)$:*

$$\mathbf{a} = (0000000), \mathbf{b} = (1000111), \mathbf{c} = (1010101), \mathbf{d} = (0010010).$$

The distances between these words are:

$$d(\mathbf{a}, \mathbf{a}) = 0, d(\mathbf{a}, \mathbf{b}) = 4, d(\mathbf{a}, \mathbf{c}) = 4, d(\mathbf{a}, \mathbf{d}) = 2, d(\mathbf{b}, \mathbf{b}) = 0,$$

$$d(\mathbf{b}, \mathbf{c}) = 2, d(\mathbf{b}, \mathbf{d}) = 4, d(\mathbf{c}, \mathbf{c}) = 0, d(\mathbf{c}, \mathbf{d}) = 4, d(\mathbf{d}, \mathbf{d}) = 0.$$

Definition 1.4.11. *The distance of a code is the minimum distance between any distinct pair of codewords, and it is denoted by d.*

Example 1.4.8. *Consider the $[7, 4]$ code with codewords as given in Example 1.4.7. The distance of this code is 2, as the minimum distance between any two distinct codewords is 2, e.g., between $\mathbf{b}$ and $\mathbf{c}$.*

Definition 1.4.12. *For positive integers n and k with $n \geq k$, and a prime power q, a code $\mathcal{C}$ is a* linear (n, k) code *over $\mathbb{F}_q$ if it forms a k-dimensional subspace of $V_q(n)$. Hence, $\mathcal{C}$ contains q^k codewords and for each two codewords $\mathbf{u}, \mathbf{v} \in \mathcal{C}$ their sum $\mathbf{u} + \mathbf{v} \in \mathcal{C}$, where the operations are done in $\mathbb{F}_q$.*

Definition 1.4.13. *A* generator matrix *of an (n, k) linear code $\mathcal{C}$ is a matrix of n columns and k linearly independent rows which form a basis of $\mathcal{C}$, i.e., any linear combination of rows of M is a codeword in $\mathcal{C}$ and any codeword in $\mathcal{C}$ is a linear combination of rows of M.*

Example 1.4.9. *Let*

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

M is a generator matrix for the linear code from Example 1.4.7. However, M is not the only generator matrix for that code. For example the matrix below is another matrix that generates the same code:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

If the probability of error or erasure is independent of a symbol's location in a codeword, a rearrangement of coordinates in the codewords does not change the properties of a code. Two codes are *equivalent* if the codewords of one code can be obtained through permutations of the coordinates of the other one.

In most cases, a linear (n, k) code can have different generator matrices, which generate the same code. For each linear code, $\mathcal{C}$, there is at least one code in the equivalence class of $\mathcal{C}$ that has a generator matrix of the form $[I_k \mid A_{(n-k)\times k}]$. It is standard for a code to be represented by a generator matrix of the form $[I_k \mid A_{(n-k)\times k}]$ that generates either the same code or an equivalent code. At the receiver's end, because of the noise from the environment the received words might not be equal to the transmitted codewords. To detect and possibly correct the errors, linear codes use the parity check matrix, denoted by H. The parity check matrix of an (n, k) linear code over $\mathbb{F}_q$ is an $(n - k) \times n$ matrix, with entries from $\mathbb{F}_q$, such that

$$GH^T = 0_{k\times(n-k)}$$

and

$$HG^T = 0_{(n-k)\times k}.$$

Any linear (n, k) q-ary code, $\mathcal{C}$, has a dual code that is a linear $(n, n - k)$ q-ary code denoted by $\mathcal{C}^\perp$, and its codewords are the linear combinations of rows of H, the parity check matrix of $\mathcal{C}$. In other words, H generates $\mathcal{C}^\perp$ and G is the parity check matrix of $\mathcal{C}^\perp$.

The following theorem shows how orthogonal arrays are related to linear codes and their dual codes.

Theorem 1.4.11. *[43, p. 231] $\mathcal{C}$ is a linear (n, k) q-ary code of distance d if and only if $\mathcal{C}^{\perp}$ is an $OA_{q^{n-k-d+1}}(d-1, n, q)$.*

In a linear code over $\mathbb{F}_q$, the following inequality, known as the Singleton bound [47], holds:

$$d \leq n - k + 1.$$

If $d = n - k + 1$, then the code is called a *maximum distance separable (MDS)* code. With respect to an (n, k) code, $\mathcal{C}$, with distance d the following statements are equivalent [27, p. 319]:

1. $\mathcal{C}$ is an MDS code.
2. any k columns of the generator matrix of $\mathcal{C}$ are linearly independent.
3. any $n - k$ columns of the parity check matrix of $\mathcal{C}$ are linearly independent.

Also, the equivalence of (n, k) MDS codes and linear $\mathrm{OA}(n, k, q)$'s has been proved [27, p. 329]. In an MDS code, even if up to $n - k$ symbols of a codeword are lost, the message can be still retrieved. The codes with the property that a message can be recovered despite the absence of some symbols have been studied under the name of *erasure codes* [47]. While error correcting codes can detect and correct up to $\lfloor \frac{d-1}{2} \rfloor$ errors at unknown coordinates, erasure codes can recover more lost symbols utilizing the fact that the coordinates of the lost symbols are known. This capability of erasure codes can be used in distributed storage systems, where some of the servers, and consequently some symbols, may not be available, yet the message (in this case, the file) should be recoverable. Reed-Solomon codes are examples of erasure codes. The remainder of this section will briefly introduce Reed-Solomon codes.

Reed-Solomon codes are used in various applications, such as digital audio recording, digital communication, and distributed storage [37, 47]. McEliece and Sarwate [28] demonstrated that a Shamir secret sharing scheme (mentioned in Section 1.4.1.1) can be formulated as a special case of Reed-Solomon codes. To create a Reed-Solomon code of length n and distance d, an element β of order n of a Galois field $\mathbb{F}_q$ is used to construct a generator polynomial $g(x)$ as follows:

$$g(x) = \prod_{i=1}^{d-1} (x - \beta^{i+a})$$

for some non-zero integer a. The $k = n - d + 1$ cyclic shifts of the coefficients of the polynomial $g(x)$ can be used to create the generator matrix G of the code. A Reed-Solomon code can also be constructed using polynomials up to a certain degree over a finite field, $\mathbb{F}_q$, where each codeword is constructed by evaluating a polynomial at n given field elements. This is basically the approach that was used in Construction 1.4.1 to construct a Shamir secret sharing scheme.

1.5 AONT Examples

As mentioned previously, Rivest [38] defined an AONT as a reversible transformation, from a set of message blocks to a set of pseudo-message blocks, such that both the transform and its inverse are "efficiently computable", while obtaining any information about any message block without the knowledge of all pseudo-message blocks is "computationally infeasible". AONTs were originally desirable as a strongly non-separable mode of operation, i.e., obtaining one block of plaintext is possible only if all ciphertext blocks are decrypted. In the same work [38], "package transform", "secret sharing schemes", "Bear and Lion schemes", and an "FFT-like scheme" are presented as instances of an AONT. We already explained the package transform in Example 1.1.1. Here we will briefly review secret sharing, Bear, and Lion schemes and their AONT properties in the following subsections. Finally, the error-propagation property of AONT is discussed and use of error-correcting codes after applying the AONT and before the encryption of the output blocks is suggested as a possible solution. Alternatively, the error-propagation property can be utilized with a redundancy block of random data to detect corrupted ciphertext.

1.5.1 Secret Sharing Scheme

Rivest [38] cites Krawczyk's work on secret sharing [23] as an instance of AONT. In his scheme [23], Krawczyk combines Shamir's secret sharing scheme [40] and Rabin's information dispersal scheme [36], i.e., first encrypting the data using a keyed encryption scheme, and then sharing the encryption key using the former scheme and the encrypted data using the latter scheme. The thresholds for the schemes can be set so that, without having all key shares, it is impossible to reconstruct the key to decrypt the encrypted data. Therefore, all participants need to contribute their key-shares and secret-shares.

Generally, if the threshold of a secret sharing scheme is equal to the number of shares, i.e., the number of participants, then the scheme satisfies the AONT conditions established

by Rivest [38]. For example, one needs all the shares of an (n, n)-Shamir secret sharing scheme to reconstruct the secret: without having even one output block (a share), it is impossible to obtain any information about the only input block (the secret); however, the total storage cost is n times the cost of storing the secret. However, usually due to the stronger conditions that the secret sharing schemes need to satisfy, they might not be efficient. For instance, Shamir secret sharing requires all shares to be of the same length as the secret.

1.5.2 Bear and Lion Schemes

Inspired by Luby and Rackoff's three round Feistel structure [26], Anderson and Biham [2] introduced BEAR and LION block ciphers. The BEAR scheme divides the message M of length m into two parts, namely L and R, such that $|L| = \ell$ and $|R| = m - \ell$. Suppose $S : \{0,1\}^{\ell} \rightarrow \{0,1\}^{*}$ is a stream cipher, i.e., a pseudo-random function which can generate outputs of arbitrary length, and $H : \{0,1\}^{m-\ell} \times \{0,1\}^{b} \rightarrow \{0,1\}^{\ell}$ is a keyed collision-free cryptographic hash function, where b is the length of the key. The process of encrypting each part is depicted in Figure 1.3. First, R is hashed using key K_2 and the result is XORed, depicted by $\oplus$, with L. Then, the outcome of the XOR operation is encrypted using S and the encrypted value is XORed with R to compute the value R'. R' is then hashed using key K_2 and the result is XORed with the XOR of L and the hashed value of R from the first hash computation, and the result is L'.

To decrypt the ciphertext, one needs to start from the bottom and follow the steps back to L and R. Regarding the AONT property of BEAR, it is clear that calculating either of L or R requires availability of both L' and R'.

The authors [2] then proved the security of the scheme against an adversary who has access to one *(plaintext, ciphertext)* pair. They proved that if an oracle can find the key of BEAR, using a plaintext-ciphertext pair, it can be used to efficiently and with high probability find the seed of the stream cipher S, as well as preimages and collisions for the hash function H. However, the scheme is not secure against an attacker who has aquired many instances of *(plaintext, ciphertext)* pairs.

In contrast to BEAR, LION, presented in Figure 1.4, does not require a keyed hash function, while providing the same level of security [2], so the hash function $H : \{0,1\}^{m-\ell} \rightarrow \{0,1\}^{\ell}$ is used, and $|K_1| = |K_2| = \ell$.

Finally, Anderson and Biham [2] introduce a four-round scheme, LIONESS, presented in Figure 1.4, that is secure against an adaptive combined chosen plaintext and ciphertext

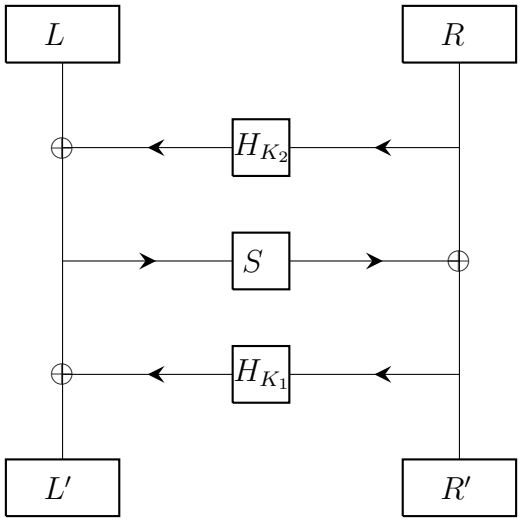


Figure 1.3: BEAR scheme

attack, to which BEAR and LION are susceptible; however, the authors do not provide any security proof.

1.6 Extended AONTs

The early theoretical work on AONTs was done by Stinson [42], Boyko [6], Canetti et al. [8], and Dodis et al. [16]. In the remainder of this section, we will discuss these studies in an increasing order of their relevance to this thesis.

To continue Rivest's work [38] in computational security and random oracle model (ROM), Boyko [6] defined semantic security and indistinguishability for AONT. The author also showed that optimal asymmetric encryption padding (OAEP), originally introduced by Bellare and Rogaway [5], satisfies the AONT properties. Boyko concluded by proving that OAEP provides close to optimal security in ROM.

In another work branching from the original AONT definition, Canetti et al. [8] studied AONTs to find a solution for the "partial key exposure" problem. This application will be discussed in Section 1.7. The authors define an ℓ-AONT as a randomized transform T, computable in polynomial time, that maps an m-bit string to an s-bit string, where $s = s_c + s_p$, the first s_c bits are kept secret and the other s_p bits are public, and T satisfies the following conditions:

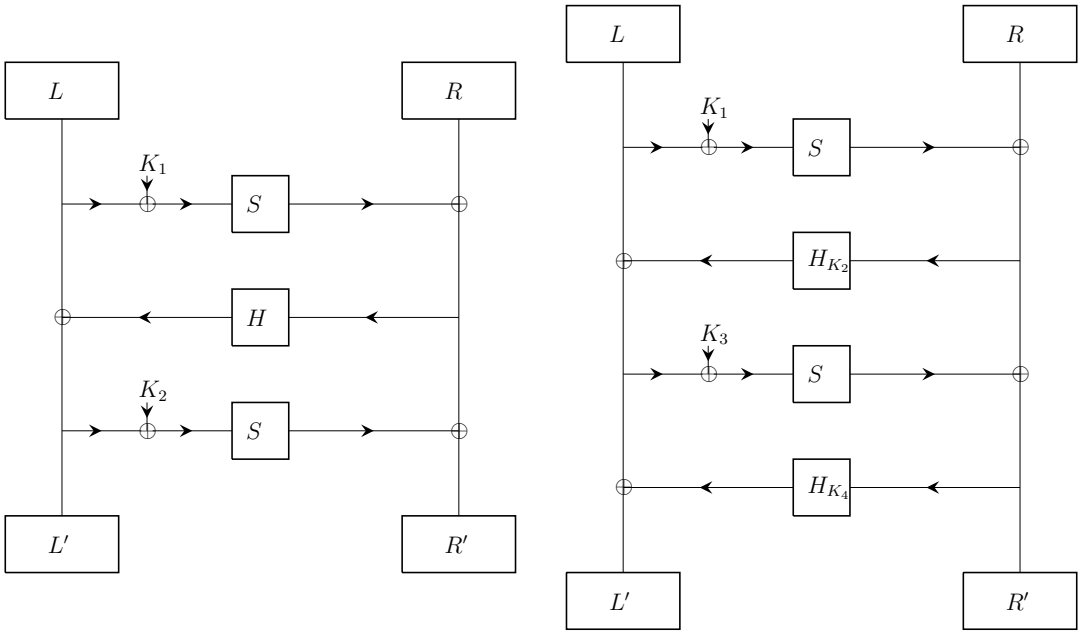


Figure 1.4: LION (on the left) and LIONESS schemes (on the right)

1. The inverse of T is also computable in polynomial time.

2. If an adversary is given all but any ℓ bits of the secret bits of the output along with all of the public portion, then the input is completely undetermined.

To construct such transforms, the authors use *exposure-resilient functions* (ERFs). An ℓ-ERF is a function computable in polynomial time that maps s bit inputs to m bit outputs, such that its output is indistinguishable from random, as long as the adversary does not have access to at least ℓ bits of the input. This definition of AONT was further studied and developed by Dodis et al. [16], where the authors considered the adaptive security of the aforementioned primitives: AONT and ERF.

Stinson [42] extended the definition of AONT and introduced unconditionally secure AONTs. This thesis generalizes AONTs presented in that work [42] and in the unconditional security setting. Therefore, in the following chapters, by AONT we mean an unconditionally secure AONT, unless otherwise noted, and we will use the following definition of an AONT due to Stinson [42], which defines AONTs using the entropy function H. For a random variable X with possible outcomes $\sigma_i, i \in \{1, 2, \ldots, n\}$, $\mathsf{H}(X) = -\sum_{i=1}^{n} P_{\sigma_i} \log P_{\sigma_i}$, where P_{σ_i} is the probability of the random variable X taking on the value σ_i. Let

$X_1, X_2, \ldots, X_s, Y_1, Y_2, \ldots, Y_s$ be $2s$ random variables with values from a finite set Σ. Then $X_1, X_2, \ldots, X_s, Y_1, Y_2, \ldots, Y_s$ form an AONT if they satisfy the following conditions:

1. $\mathsf{H}(X_1, X_2, \ldots, X_s \mid Y_1, Y_2, \ldots, Y_s) = 0$,
2. $\mathsf{H}(Y_1, Y_2, \ldots, Y_s \mid X_1, X_2, \ldots, X_s) = 0$,
3. $\mathsf{H}(X_i \mid Y_1, Y_2, \ldots, Y_{j-1}, Y_{j+1}, \ldots, Y_s) = \mathsf{H}(X_i)$, for all $i, j \in \{1, 2, \ldots, s\}$.

Remark 1.6.1. *We note that the security achieved by the AONT depends on the probability distribution defined on the input s-tuples. This will discussed in detail in Section 2.1 in a more general setting.*

Stinson [42] then defines a *linear* (s, q)*-AONT* over a finite field $\mathbb{F}_q$ to be an AONT, where each of the s output elements is an $\mathbb{F}_q$-linear combination of the s input elements. Stinson [42] shows that the transform is given by $\mathbf{y} = \mathbf{x}M^{-1}$, where M is an s by s matrix with entries from $\mathbb{F}_q$ that only consists of non-zero elements, and $\mathbf{x}$ and $\mathbf{y}$ are row vectors of length s. Then the inverses of $s \times s$ Hadamard matrices modulo p are introduced as instances of linear (s, p)-AONTs, where $p > 2$ is prime and s is a multiple of 4. Also, a construction for linear (s, q)-AONTs is given for prime powers $q > 2$, and any positive integer s. Stinson [42] then proves the non-existence of linear $(s, 2)$-AONTs for $s \geq 2$, and presents $J - I$ as the closest approximation to a linear AONT for even values of s (I is the identity matrix, and J is the matrix with all entries equal to 1). The study of structures close to AONT was continued in later work [14,31] and will be discussed further in Chapter 3. Finally, the equivalence between AONT and certain families of arrays and orthogonal arrays is proven, and some existence results are provided.

We finish this section with an example of an unconditionally secure linear AONT.

Example 1.6.1. *The matrix M^{-1} below is a $(4, 3)$-AONT. Checking the inverse matrix, it can be verified that M does not have any singular 1×1 submatrices, i.e., zero entries. Therefore, as proven by Stinson [42], fixing any $3 (= s - 1)$ output elements does not yield any information about any one input element.*

$$M^{-1} = \begin{pmatrix} 1 & 2 & 2 & 2 \\ 2 & 1 & 0 & 0 \\ 2 & 0 & 1 & 0 \\ 2 & 0 & 0 & 1 \end{pmatrix}, \quad M = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 2 \end{pmatrix}.$$

1.7 Applications

Besides the theoretical study of AONTs, the all-or-nothing property of these structures has been employed in various setups and to achieve different goals. In this section, we will briefly review some of these applications of AONTs in different categories. This review serves as a motivation for the work being presented in the following chapters, but it is not a prerequisite for those chapters, nor it is a comprehensive survey on the applications of AONTs.

1.7.1 Access Control

In remote storage systems, it is common for more than one user to have access to the content; however, there is only one data provider. Suppose that the set of users who require access to the data is dynamic, where some users will lose their access privileges, while new users can join the group as well. In this setting, revoking a user's access to older and newer versions of the documents is a concern. To achieve this goal, Bacis et al. [3], Chen et al. [11], Cheng et al. [12], and Wu et al. [51] have used AONTs in their schemes.

Chen et al. [11] present a hierarchical key assignment scheme, *CloudHKA*, for cloud storage that provides a security model with different access levels and read operations for the users and various access management controls for the data provider, including access revocation. In particular, Chen et al. [11] utilize the AONT as a mode of operation. In their model, the blocks of data, i.e., input elements, are preprocessed using an AONT prior to being encrypted using a key that is derived from the access level keys. This preprocessing solely helps to defend against the attacks on the encryption scheme, which is the same purpose that Rivest [38] originally proposed AONTs.

Cheng et al. [12] and Bacis et al. [3] utilize the all-or-nothing property to reduce the cost of revoking a user's access to the data. Their schemes first apply an AONT on the data so that all the blocks are interdependent. Then they encrypt the output blocks. To revoke a user's access to the data, they need to re-encrypt only one output block using a new encryption key that is unknown to the user whose access is being revoked. The interdependency between the blocks requires the data provider to recalculate all the blocks after each revocation. To reduce this cost, Bacis et al. [3] introduce a hierarchical structures on the blocks, and apply the AONT on groups of blocks, namely *macro blocks*.

In a slightly different setting, Wu et al. [51] proposed an access control scheme for *named data networking* (NDN), a content-oriented architecture. The main components of the scheme are AONTs and network coding. Each file on the network is segmented into

shares stored on different nodes. To gain access to a file, a node receives the segments from the nodes storing those segments and recovers the file. The proposed scheme applies an AONT on the segments so that a user requires all the output blocks to learn about a file. To restrict access to a file, the owner of the file should encrypt one output block and also keeps the used AONT a secret. The encryption key and the AONTs are then provided to the authorized users.

1.7.2 Secure Data Transfer

Guo et al. [17] studied the application of AONTs in providing security for wireless networks. In particular, they suggested the use of linear AONTs for defending against wiretapping attacks and detecting Byzantine attacks, i.e., passively wiretapping the network and altering the messages on the network, respectively. They applied AONTs on encrypted blocks to defend against the attacks utilizing linearity of the AONTs. In their model, the source node sends each output block to a neighboring node. Each node other than the source and the sink forwards the sum of the blocks it has received. If all the nodes follow the protocol honestly and the adversary can only control one communication line, then the AONT and the encryption defend the network against wiretapping attacks and the error propagation property of AONTs helps the receiver detect modifications to the blocks.

In a more recent work, Pham et al. [33] utilized unconditionally secure AONTs to share the security of an unconditionally secure channel that uses optical encryption with a regular channel that guarantees security in computational setting. In their model, they apply an AONT on the message and then send one output block through the unconditionally secure channel and the rest of it through the other channel. Even if an adversary learns about all the output blocks sent over the regular channel, since they do not have access to the output block that is sent via the unconditionally secure channel, they do not learn anything about any of the message blocks. Although the authors analyze their scheme with an optical encryption scheme as the unconditionally secure channel, their scheme can utilize any unconditionally secure encryption method, such as one-time pad, and provide the same level of security.

1.7.3 Secure Distributed Storage

With the advent of cloud computing/storage and popularity of outsourcing storage, the security of the remotely stored data has become a concern. Hence, experts with different backgrounds and research fields have studied a myriad of approaches and techniques to

address various aspects of this concern. Many of the schemes designed for secure distributed storage are based on earlier definitions of AONT. These schemes use AONTs as a pre-processing step followed by a coding algorithm that encodes the output of the AONT to shares, which will be distributed among servers.

Resch and Plank in [37] used AONT-RS, a combination of Reed-Solomon codes and package transforms, to securely disperse a document over different servers. First, the AONT is applied to the document, so that all of the k output blocks are needed to reconstruct the document. Then an erasure code, namely, a Reed-Solomon code, is used to encode those k blocks into n blocks such that any k of these blocks will reconstruct the output blocks of the AONT. Erasure codes help recovering the original message, even if parts of the encoded data are lost. Subsequently, Chen et al. [10] introduced two adversarial games and proved the computational security and privacy of AONT-RS in the random oracle model.

Based on AONT-RS, Baldi et al. [4] developed AONT-LT. Their objective was to make it possible to have different sizes of blocks, and more storage devices. To achieve this they used a type of fountain code, namely, Luby transforms, in conjunction with a package transform, as AONT. Fountain codes are a family of codes that encode data into a very large number of codewords; for decoding, a receiver will collect codewords until the collected codewords are sufficient to recover the message. Fountain codes can be considered to be a type of erasure code because a lost codeword will not be collected by the receiver, but the receiver will eventually collect enough codewords to decode the message.

In an extension of Stinson's unconditionally secure AONTs, Karame et al. [21] introduced bastion AONTs. Bastion AONT can be presented as an $s \times s$ matrix $J - I$, where J is an all-one square matrix and I is the identity matrix, to hide information about any block in the absence of any two blocks. This construction can be implemented only by XORs, so it is fast and efficient. Suppose $X = (x_1, x_2, \ldots, x_s)$ is the input, and $Y = (y_1, y_2, \ldots, y_s)$ is the output. Based on the XOR formulation of this construction by Stinson [42] the bastion AONT can be computed as follows:

$$t = \bigoplus_{i=1}^{s} x_i$$

$$y_i = t \oplus x_i$$

Karame et al. [21] then used this construction to distribute encrypted blocks on multiple servers to prevent an adversary who has access to the long term key from obtaining information regarding the encrypted data, as long as there are at least two servers to which the

adversary does not have access.

Based on the bastion AONTs, Kapusta and Memmi [20] devised the selective-AONT scheme to store data confidentially on a remote server and a personal device. This scheme keeps a small part of the data on the personal device, and the rest of the data is stored on the remote server. This approach is interesting in that it uses the minimal number of remote servers. Selective-AONT uses a combination of CBC mode of encryption, a block cipher, and bastion AONT. The blocks of data are divided into private and public shares and encrypted and interlinked using CBC mode of operation. Then the bastion AONT is applied on the private shares and a portion of the public private shares.

In a more relevant work, Oliveira et al. [32] study the use of super-regular matrices[1] in information dispersal with regard to "fault-tolerance", "recovery efficiency", and computational complexity of recovery. Their constructions are equivalent to t-AONT [32], as they use $k \times n$ matrices of rank k, where $k < n$, such that for any $t < k$, all $t \times t$ submatrices of the original matrices are invertible.

1.7.4 Anti-jamming Techniques

In wireless networks, since the signals are transmitted through the air and the space around us, it is usually infeasible to restrict adversaries' access to the medium. Although cryptographic tools can be used to achieve confidentiality, integrity, and authenticity, these networks are susceptible to jamming attacks, through impacting the signals received by the receiver by generating stronger signals or noises on the communication channel used between the sender and receiver. Since the adversary requires a stronger signal to overshadow the sender's signal, it is in the adversary's interest to jam the minimum number of packets possible that are required to prevent meaningful reconstruction of the packets at the receiver's end. Such attacks are known as selective jamming attacks.

Proaño and Lazos [34,35] introduced and developed AONT-based packet-hiding schemes to protect the communication on a wireless network against an internal active attacker who tries to selectively jam the network. In a related work, Lazos and Krunz [24] studied this problem in a wireless mesh network setting. The proposed schemes apply an AONT on the packets so that the adversary cannot identify and target certain packets in real time. To selectively jam a packet, the attacker needs to collect all the sent packets, reconstruct the original packets, and then decide whether or not to jam the signal for that packet; however, the receiver has received the message if the adversary has not attacked the packet

[1] A matrix is super-regular if all its square submatrices are invertible.

already. These schemes [24, 34, 35] consider the adversary to be incapable of jamming all the packets due to insufficient power. However, the system as proposed is susceptible to jamming attacks because the adversary needs to jam only one block out of each message and the error-propagation property of AONTs, as mentioned by Rivest [38], will amplify the attack to all blocks. If we assume that the adversary is reluctant to jam the whole network, for example, to keep the attack undetected, then this model might be useful.

1.7.5 Defending Against Partial Key Exposure

For a cryptosystem, it is usual to study its security under the assumption that the keys are completely unknown to the adversary, and whenever the adversary learns about the key, it is considered to be full knowledge, thus, the system is compromised. In real-life however, it is possible for an adversary to obtain partial knowledge of the key. Suppose an adversary knows t bits of the encryption key. This knowledge might help the adversary to decrypt the message or to obtain information about the corresponding plaintext, without full knowledge of the key. It is undeniable that the partial knowledge of the key reduces the key space, however, our concern here is shortcuts that do not involve key exhaustion techniques and partial information regarding the plaintext.

Canetti et al. [8] studied the case where it is possible for an adversary to learn a portion of the key. To protect the system against an adversary with partial knowledge of the secret key, the authors propose the use of an AONT on the key. In their model, even if the adversary knows up to t bits of the key, since a t-AONT is applied on the key, then the adversary cannot learn anything about any bit of the derived key that is used for the encryption. The only advantage the attacker has gained is that if they decide to conduct an exhaustive search on the key domain, they know those fixed t bits, but they still need to apply AONT on every single guess, and then try to decrypt the message using the derived key.

Chapter 2

t-AONTs

Generalizations of all-or-nothing transforms (AONTs) are desirable from two different aspects: theoretical, where generalizations open doors to new problems and present us with better understanding of these structures; and applications, where they provide us with more options and flexibility with the parameters. Based on Stinson's [42] definition of unconditionally secure AONTs and the definition of 2-AONTs from a collaboration with Paolo D'Arco and Doug Stinson [14], Ian Goldberg and Doug Stinson and I [29] studied a generalization of AONTs. This chapter mostly reports results from that work. This generalization of AONTs considers the security of t input elements in the absence of t output elements, for $t \geq 1$. The second part of this chapter focuses on results for $t = 2$. At the end, two applications of t-AONTs are presented.

2.1 From 1 to t

As mentioned above, the first generalization of all-or-nothing-transforms presented in this thesis concerns the size of input-element sets about which the adversary should not be able to learn any information. We define a (t, s, v)-AONT to be a bijection from s input elements to s output elements, such that in the absence of any t output elements, no information can be obtained about any t input elements. Based on two different interpretations of this definition, we define (t, s, v)-AONTs with *perfect security* and with *weak security*.

In general, we will assume that every input s-tuple occurs with a non-zero probability, and since an AONT is a bijection, every output s-tuple occurs with a non-zero probability. In the stronger security interpretation the probability that t inputs take on any t specified

values, given the values of any $s-t$ outputs, is the same as the a priori probability that they take on the same values. In weak security any t inputs can take on any possible values with a non-zero probability, given the values of any $s-t$ outputs. More formally, we have the following definitions.

Definition 2.1.1. *Let*

$$\mathbf{X_1}, \ldots, \mathbf{X_s}, \mathbf{Y_1}, \ldots, \mathbf{Y_s}$$

*be random variables taking on values in the finite set Σ of size v. These $2s$ random variables define a (t, s, v)-*AONT *provided that the following conditions are satisfied:*

1. $\mathsf{H}(\mathbf{Y_1}, \ldots, \mathbf{Y_s} \mid \mathbf{X_1}, \ldots, \mathbf{X_s}) = 0.$
2. $\mathsf{H}(\mathbf{X_1}, \ldots, \mathbf{X_s} \mid \mathbf{Y_1}, \ldots, \mathbf{Y_s}) = 0.$
3. *For all $\mathcal{X} \subseteq \{\mathbf{X_1}, \ldots, \mathbf{X_s}\}$ with $|\mathcal{X}| = t$, and for all $\mathcal{Y} \subseteq \{\mathbf{Y_1}, \ldots, \mathbf{Y_s}\}$ with $|\mathcal{Y}| = t$, it holds that*
$$\mathsf{H}(\mathcal{X} \mid \{\mathbf{Y_1}, \ldots, \mathbf{Y_s}\} \setminus \mathcal{Y}) = \mathsf{H}(\mathcal{X}). \tag{2.1}$$

Condition 2.1 of Definition 2.1.1 can be satisfied if the input s-tuples occur with uniform probability.on the set of all possible input s-tuples. Theorem 2.2.2 will prove this property.

The parameters t, s, and v are called the *strength*, *size*, and *alphabet size* of the AONT, respectively.

Definition 2.1.2. *Let*

$$\mathbf{X_1}, \ldots, \mathbf{X_s}, \mathbf{Y_1}, \ldots, \mathbf{Y_s}$$

be random variables taking on values in the finite set Σ of size v. These $2s$ random variables define a weakly secure (t, s, v)-AONT *provided that the following conditions are satisfied:*

1. $\mathsf{H}(\mathbf{Y_1}, \ldots, \mathbf{Y_s} \mid \mathbf{X_1}, \ldots, \mathbf{X_s}) = 0.$
2. $\mathsf{H}(\mathbf{X_1}, \ldots, \mathbf{X_s} \mid \mathbf{Y_1}, \ldots, \mathbf{Y_s}) = 0.$
3. *Given the values of any $s-t$ outputs, any t inputs take on any possible values with a non-zero probability.*

Figure 2.1 illustrates the AONT property. Let all the v^s possible inputs be listed on the left half of the array, and the output corresponding to each input is listed in the right half of the same row. In this setting, fixing any $s-t$ coordinates of an output does not yield any information, in the sense of weak security, about any t input coordinates.

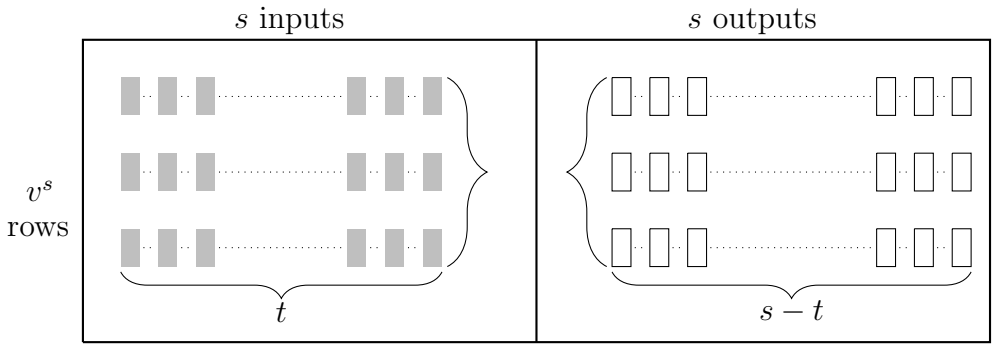


Figure 2.1: (t, s, v)-AONT: in array format

Example 2.1.1 presents a $(2, 3, 3)$-AONT and demonstrates how missing any pairs (from y_1, y_2, and y_3) of its output elements does not yield any information, in the sense of weak security, about any pairs (from x_1, x_2, and x_3) of its input elements.

Example 2.1.1. *The Table 2.1 presents a $(2, 3, 3)$-AONT, over the alphabet $\{a, b, c\}$, by listing the outputs y_1, y_2, and y_3 for all possible values of input elements x_1, x_2, and x_3. Suppose an adversary learns that $y_2 = a$. This knowledge allows them to exclude 18 possible inputs, e.g., $(a, a, b), (a, a, c)$, and (a, b, c); however, all combinations of values for any pair of input elements are still possible. For example, all of the 9 possible values for the pair (x_2, x_3) are highlighted in the table, and it can be verified that each combination of values for that pair occurs exactly once.*

Figure 2.2 depicts the behavior of a t-AONT. The area hatched in blue presents the number of protected input blocks are protected upon the availability of that many output blocks to the adversary.

As Stinson [42] and D'Arco et al. [14] mentioned, linear AONTs are desirable due to their simplicity and efficiency; however, the study of general AONTs is still interesting and useful, and some of the results apply to linear AONTs too. Therefore, we continue with results on general AONTs, in Section 2.2, followed by the definition and properties of linear AONTs in Section 2.3.

Table 2.1: A $(2, 3, 3)$-AONT over the alphabet $\{a, b, c\}$

x_1	x_2	x_3	y_1	y_2	y_3
a	a	a	a	a	a
a	a	b	c	c	b
a	a	c	b	b	c
a	b	a	c	b	c
a	b	b	b	a	a
a	b	c	a	c	b
a	c	a	b	c	b
a	c	b	a	b	c
a	c	c	c	a	a
b	a	a	b	c	c
b	a	b	a	b	a
b	a	c	c	a	b
b	b	a	a	a	b
b	b	b	c	c	c
b	b	c	b	b	a
b	c	a	c	b	a
b	c	b	b	a	b
b	c	c	a	c	c
c	a	a	c	b	b
c	a	b	b	a	c
c	a	c	a	c	c
c	b	a	b	c	a
c	b	b	a	b	b
c	b	c	c	a	c
c	c	a	a	a	c
c	c	b	c	c	a
c	c	c	b	b	b

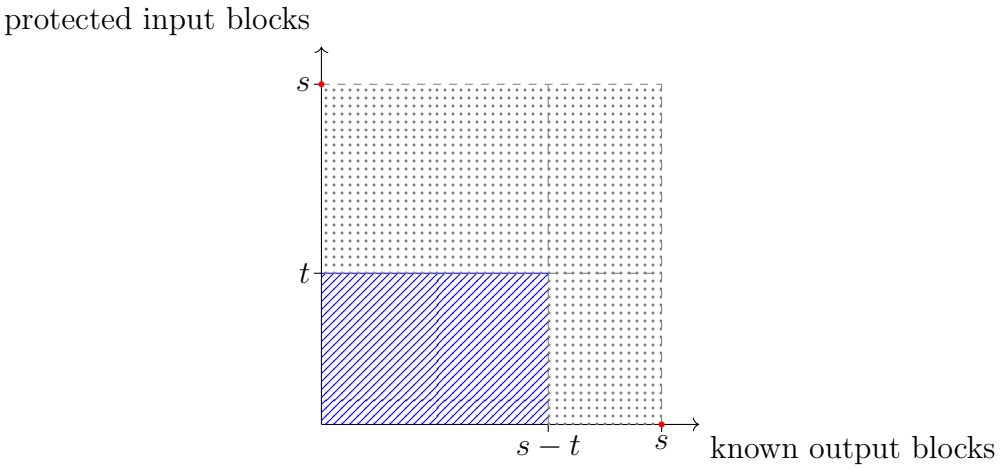


Figure 2.2: The behavior of a (t, s, v)-AONT for different numbers of available output blocks.

2.2 Results on General AONTs

In this section, we present results on the relationships between "general" AONTs, that is linear and non-linear AONTs, and three types of combinatorial structures: unbiased arrays, orthogonal arrays (OAs), and resilient functions (RFs). See Section 1.4 for definitions. Knowing these relations allows us to apply currently known properties of one structure to the other related structures.

We start with a result that characterizes (t, s, v)-AONT in terms of unbiased arrays.

Theorem 2.2.1. *A weakly secure (t, s, v)-AONT is equivalent to a $(v^s, 2s, v)$-array that is unbiased with respect to the following subsets of columns:*

1. $\{1, \ldots, s\}$,
2. $\{s+1, \ldots, 2s\}$, *and*
3. $I \cup \{s+1, \ldots, 2s\} \setminus J$, *for all* $I \subseteq \{1, \ldots, s\}$ *with* $|I| = t$ *and all* $J \subseteq \{s+1, \ldots, 2s\}$ *with* $|J| = t$.

Proof. Let A be the hypothesized $(v^s, 2s, v)$-array on alphabet $\mathcal{X}$, $|\mathcal{X}| = v$. We construct

$\phi : \mathcal{X}^s \to \mathcal{X}^s$ as follows: for each row $(x_1, \ldots, x_{2s})$ of A, define

$$\phi(x_1, \ldots, x_s) = (x_{s+1}, \ldots, x_{2s}).$$

Being unbiased with respect to the first two subsets of columns indicates that ϕ is a bijection, and being unbiased with respect to the third subset of columns is equivalent to Condition (3) of Definition 2.1.2. Hence, the function ϕ is a weakly secure (t, s, v)-AONT.

Conversely, suppose ϕ is a weakly secure (t, s, v)-AONT. Let A be the array presentation of the AONT as depicted in Figure 2.1. The array's rows consist of all v^s $2s$-tuples $(x_1, \ldots, x_{2s})$, where $\phi(x_1, \ldots, x_s) = (x_{s+1}, \ldots, x_{2s})$. Then A is the desired $(v^s, 2s, v)$-array. □

Theorem 2.2.2. *If all input s-tuples are equally probable, then the unbiased array is a perfectly secure AONT.*

Proof. We prove this theorem by showing that values of any t inputs are independent of any $(s - t)$ outputs. Consider any arbitrary sub-list of t inputs, A, and any sub-list of $(s-t)$ outputs, B. Since there are v^s rows, given any input t-tuple, $\sigma_i = (\sigma_{i_1}, \sigma_{i_2}, \ldots, \sigma_{i_t})$, there are v^{s-t} rows where $A = \sigma_i$. Since all the v^s rows are equiprobable, the probability of any specified input t-tuple can be calculated as follows:

$$Pr[A = \sigma_i] = \frac{v^{s-t}}{v^s} = v^{-t}.$$

Similarly, given any output $(s - t)$-tuple, $\sigma_j = (\sigma_{j_1}, \sigma_{j_2}, \ldots, \sigma_{j_{s-t}})$, there are v^t rows with $B = \sigma_j$, so the probability of any specified output $(s - t)$-tuple is

$$Pr[B = \sigma_j] = \frac{v^t}{v^s} = v^{t-s}.$$

We also know that any input t-tuple and output $(s-t)$-tuple only appears together in one row. Therefore,

$$\Pr[A = \sigma_i \wedge B = \sigma_j] = \frac{1}{v^s} = v^{-s}.$$

Hence, for any given input t-tuple and any given output $(s - t)$-tuple,

$$Pr[A = \sigma_i]Pr[B = \sigma_j] = v^{-t}v^{s-t} = v^{-s} = \Pr[A = \sigma_i \wedge B = \sigma_j].$$

Therefore, any specified t inputs and any specified $s - t$ outputs are independent, which is what we intended to prove.

□

In the rest of the thesis, the term (t, s, v)-AONT will be used in reference to an unbiased array. Hence, it automatically provides weak security when every input s-tuple occurs with positive probability, and if all the input s-tuples are equiprobable, then the array provides perfect(or strong) security.

Theorem 2.2.1 immediately implies the following corollary by Wang et al. [50].

Corollary 2.2.3. *A mapping $\phi : \mathcal{X}^s \to \mathcal{X}^s$ is a (t, s, v)-AONT if and only if ϕ^{-1} is an $(s - t, s, v)$-AONT.*

Proof. Interchange the first s columns with the last s columns in the array representation of the AONT. □

An $\mathrm{OA}(s, 2s, v)$ is an array that is unbiased with respect to the three groups of column-sets that were listed in Theorem 2.2.1, as well as many other groups. Therefore, the following corollary immediately follows from Theorem 2.2.1.

Corollary 2.2.4. *If there exists an $OA(s, 2s, v)$, then there exists a (t, s, v)-AONT for all t such that $1 \leq t \leq s$.*

We note that orthogonal arrays are equivalent to maximum distance separable (MDS) codes. Hence, it is easy to construct (t, s, v)-AONTs whenever v is a prime power and $2s \leq v$. Cauchy matrices result in the same bound for linear AONTs, as will be discussed in Theorem 2.4.1. In the case of $v = 2$, it has been previously shown by Stinson [42, Theorem 3.5] that there is no $(1, s, 2)$-AONT (linear or nonlinear) if $s \geq 2$.

Corollary 2.2.4 presents a construction of AONTs from certain orthogonal arrays. The next theorem addresses the construction of an orthogonal array from an AONT.

Theorem 2.2.5. *Suppose there is a (t, s, v)-AONT. Then there is an $OA(t, s, v)$.*

Proof. We represent the (t, s, v)-AONT in its array presentation, which is a $(v^s, 2s, v)$-array denoted by A. Let R denote the rows of A that contain a fixed $(s - t)$-tuple in the last $s - t$ columns of A. Then $|R| = v^t$. Delete all the rows of A not in R and delete the last s columns of A and call the resulting array A'. Within any t columns of A', we see that every t-tuple of symbols occurs exactly once, since the rows of A' are determined by fixing $s - t$ outputs of the AONT. Hence, A' is an $\mathrm{OA}(t, s, v)$. □

a	a	c	b
a	b	b	c
a	c	a	a
b	a	a	c
b	b	c	a
b	c	b	b
c	a	b	a
c	b	a	b
c	c	c	c

a	a	c	b
b	c	b	b
c	b	a	b

$\longrightarrow$

a	a
b	c
c	b

Figure 2.3: *Top:* a $(9, 4, 3)$-array that is unbiased with respect to the following set of columns: $I = \{1, 2\}, O = \{3, 4\}$, and $\{a, b : a \in I, b \in O\}$. *Bottom:* extracting an $\mathrm{OA}(2, 1, 3)$ from the $(9, 4, 3)$-array.

Example 2.2.1. *In Example 1.1.2, if the columns are not divided based on inputs and outputs, it can be verified that the $(1, 2, 3)$-AONT introduced in Chapter 1 is equivalent to an array with alphabet size three, four columns, and nine rows, i.e., a $(9, 4, 3)$-array, that is unbiased with respect to any two columns. These are all the column combinations specified in Theorem 2.2.1.*

It should be noted that the array shown in Figure 2.3 is an $OA(2, 4, 3)$ as well. However, to illustrate the result of Theorem 2.2.5 on this example, we need to select the rows with a fixed value, e.g., b, in the last column. It is easy to verify that the left two columns in those three rows form an OA of strength 1, i.e., the left half of the array is unbiased with respect to any single column. The process of obtaining the $OA(1, 2, 3)$ is depicted in Figure 2.3.

Now consider the following classical bound on orthogonal arrays, which can be found in the Handbook of Combinatorial Designs [13].

Theorem 2.2.6 (Bush Bound)**.** *If there is an $OA(t, s, v)$, then*

$$s \leq \begin{cases} v + t - 1 & \text{if } t = 2, \text{ or if } v \text{ is even and } 3 \leq t \leq v \\ v + t - 2 & \text{if } v \text{ is odd and } 3 \leq t \leq v \\ t + 1 & \text{if } t \geq v. \end{cases}$$

The following corollaries are the direct results of fixing t to be 2 and 3 in Theorem 2.2.6.

Corollary 2.2.7. *If there is a $(2, s, v)$-AONT, then $s \leq v + 1$.*

Corollary 2.2.8. *If there is a $(3, s, v)$-AONT, then $s \leq v+2$ if $v \geq 4$ is even, and $s \leq v+1$ if $v \geq 3$ is odd.*

Since the existence of an $\mathrm{OA}(t, s, v)$ is a necessary condition for the existence of a (t, s, v)-AONT, the same inequality holds between the size of the AONT and its alphabet size and strength.

Next, we show that any AONT, linear or nonlinear, gives rise to a resilient function. This result is based on a characterization of resilient functions by Stinson [41], which was stated in Theorem 1.4.3.

Theorem 2.2.9. *Suppose there is a (t, s, v)-AONT. Then there is an $(s, s-t, t, v)$-resilient function.*

Proof. We use the same technique that was used in the proof of Theorem 2.2.5. Let A be the $(v^s, 2s, v)$-array representing the AONT. For any $(s-t)$-tuple $\mathbf{x}$, let $R_{\mathbf{x}}$ be the rows of A that contain $\mathbf{x}$ in the last $s-t$ columns of A. Let $A_{\mathbf{x}}$ denote the array formed by the rows in $R_{\mathbf{x}}$ and the first s columns of A. Theorem 2.2.5 showed that $A_{\mathbf{x}}$ is an $\mathrm{OA}(t, s, v)$.

Now, consider all v^{s-t} possible $(s-t)$-tuples $\mathbf{x}$. For each choice of $\mathbf{x}$, we get an $\mathrm{OA}(t, s, v)$. These v^{s-t} orthogonal arrays together contain all v^s s-tuples, since the array A is unbiased with respect to the first s columns. Thus we have a large set of $\mathrm{OA}_1(t, s, v)$. Applying Theorem 1.4.3, this large set of OAs is equivalent to an $(s, s-t, t, v)$-resilient function (note that $m = s - t$ because $v^{s-m-t} = 1$). □

2.3 Linear t-AONTs

Let q be a prime power. An AONT with alphabet $\mathbb{F}_q$ is *linear* if each output element y_i is an $\mathbb{F}_q$-linear function of the input elements $x_1, \ldots, x_s$. For a linear AONT, we can write

$$\mathbf{y} = \phi(\mathbf{x}) = \mathbf{x}M^{-1} \quad \text{and} \quad \mathbf{x} = \phi^{-1}(\mathbf{y}) = \mathbf{y}M, \tag{2.2}$$

where M is an invertible s by s matrix with entries from $\mathbb{F}_q$. Subsequently, when we refer to a "linear AONT", we mean the matrix M that transforms $\mathbf{y}$ to $\mathbf{x}$, as specified in (2.2).

The following lemma by D'Arco et al. [14] presents the properties of linear all-or-nothing transforms in terms of the properties of the matrix M.

For $I, J \subseteq \{1, \ldots, s\}$, define $M(I, J)$ to be the $|I|$ by $|J|$ submatrix of M induced by the *columns* in I and the *rows* in J. The following lemma characterizes linear all-or-nothing transforms in terms of properties of the matrix M. This lemma can be considered to be a generalization of a similar result for linear 1-AONTs by Stinson [42, Theorem 2.1].

Lemma 2.3.1. *[14, Lemma 1] Suppose that q is a prime power and M is an invertible s by s matrix with entries from $\mathbb{F}_q$. Let $\mathcal{X} \subseteq \{X_1, \ldots, X_s\}$, $|\mathcal{X}| = t$, and let $\mathcal{Y} \subseteq \{Y_1, \ldots, Y_s\}$, $|\mathcal{Y}| = t$. Then the function $\phi(\mathbf{x}) = \mathbf{x}M^{-1}$ satisfies Condition (3) with respect to $\mathcal{X}$ and $\mathcal{Y}$ if and only if the submatrix $M(I, J)$ is invertible, where $I = \{i : X_i \in \mathcal{X}\}$ and $J = \{j : Y_j \in \mathcal{Y}\}$.*

Proof. Let $\mathbf{x}' = (x_i : i \in I)$. We have $\mathbf{x}' = \mathbf{y}M(I, \{1, \ldots, s\})$. Now assume that y_j is fixed for all $j \notin J$. Then we can write $\mathbf{x}' = \mathbf{y}'M(I, J) + \mathbf{c}$, where $\mathbf{y}' = (y_j : j \in J)$ and $\mathbf{c}$ is a vector of constants.

If $M(I, J)$ is invertible, then $\mathbf{x}'$ is completely undetermined, in the sense that $\mathbf{x}'$ takes on all values in $(\mathbb{F}_q)^t$ as $\mathbf{y}'$ varies over $(\mathbb{F}_q)^t$. On the other hand, if $M(I, J)$ is not invertible, then $\mathbf{x}'$ can take on only $(\mathbb{F}_q)^{t'}$ possible values, where $\mathsf{rank}(M(I, J)) = t' < t$. □

Corollary 2.3.2. *Suppose that q is a prime power and M is an invertible s by s matrix with entries from $\mathbb{F}_q$. Then $\mathbf{y} = \mathbf{x}M^{-1}$ defines a (t, s, q)-AONT if and only if all t by t submatrices of M are invertible.*

Example 2.3.1. *Consider a linear AONT with matrix M as follows:*

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 2 \\ 1 & 0 & 1 \end{pmatrix}.$$

Note that the top right 2×2 submatrix is singular.

If M^{-1} is used as a $(2, 3, 3)$-AONT to map the message (x_1, x_2, x_3) to (y_1, y_2, y_3), then the following equations hold:

$$y_1 + 2y_2 = x_2$$

$$y_1 + 2y_2 + y_3 = x_3$$

$$\Rightarrow x_3 - x_2 = y_3.$$

Therefore, it is possible to calculate $f(x_2, x_3) = x_3 - x_2$, a function of $t = 2$ input elements, if y_3 is known, that is, in the absence of $t = 2$ output elements, y_1 and y_2, which contradicts an AONT property.

Remark 2.3.1. *Any invertible s by s matrix with entries from $\mathbb{F}_q$ defines a linear (s, s, q)-AONT.*

Corollary 2.3.3. *Suppose that $\mathbf{y} = \mathbf{x}M^{-1}$ defines a linear (t, s, q)-AONT. Then $\mathbf{x} = \mathbf{y}M$ defines a linear $(s - t, s, q)$-AONT.*

The proof of Corollary 2.3.3 follows from Corollary 2.2.3.

Corollary 2.3.4. *Suppose M is an invertible s by s matrix with entries from $\mathbb{F}_q$. Then $\mathbf{y} = \mathbf{x}M^{-1}$ defines a linear (t, s, q)-AONT if and only if every $(s - t)$ by $(s - t)$ submatrix of M is invertible.*

Proof. From Corollary 2.3.3, $\mathbf{y} = \mathbf{x}M$ is a linear (t, s, q)-AONT if and only if $\mathbf{y} = \mathbf{x}M^{-1}$ is a linear $(s-t, s, q)$-AONT. Therefore, from Lemma 2.3.1, $\mathbf{y} = \mathbf{x}M$ is a linear (t, s, q)-AONT if and only if every $(s - t)$ by $(s - t)$ submatrix of $(M^{-1})^{-1} = M$ is invertible. □

To summarize, we have proven the following.

Theorem 2.3.5. *Suppose M is an invertible s by s matrix with entries from $\mathbb{F}_q$. Then the following are equivalent.*

1. *$\mathbf{y} = \mathbf{x}M^{-1}$ is a linear (t, s, q)-AONT.*
2. *Every $(s - t)$ by $(s - t)$ submatrix of M^{-1} is invertible.*
3. *Every t by t submatrix of M is invertible.*

2.4 Existence of Linear t-AONTs

As Corollary 2.3.2 indicated, the existence of a linear (t, s, q)-AONT is equivalent to the existence of an invertible $s \times s$ matrix, with elements from $\mathbb{F}_q$, having only invertible $t \times t$ submatrices. Cauchy matrices are a family of matrices all of whose square submatrices, including itself, are invertible. An s by s Cauchy matrix can be defined over $\mathbb{F}_q$ if $q \geq 2s$. Let $a_1, \ldots, a_s, b_1, \ldots, b_s$ be distinct elements of $\mathbb{F}_q$. Let $c_{ij} = (a_i - b_j)^{-1}$, for $1 \leq i \leq s$ and $1 \leq j \leq s$. Then $C = (c_{ij})$ is the Cauchy matrix defined by the sequence $a_1, \ldots, a_s, b_1, \ldots, b_s$.

Example 2.4.1. *Let $s = 3$ and $q = 7$. A 3×3 Cauchy matrix over $\mathbb{F}_7$ can be constructed as follows.*

$$\begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 4 & \frac{1}{4-1} & \frac{1}{4-2} & \frac{1}{4-3} \\ 5 & \frac{1}{5-1} & \frac{1}{5-2} & \frac{1}{5-3} \\ 6 & \frac{1}{6-1} & \frac{1}{6-2} & \frac{1}{6-3} \end{array} \Rightarrow \begin{pmatrix} \frac{1}{3} & \frac{1}{2} & \frac{1}{1} \\ \frac{1}{4} & \frac{1}{3} & \frac{1}{2} \\ \frac{1}{5} & \frac{1}{4} & \frac{1}{3} \end{pmatrix} = \begin{pmatrix} 5 & 4 & 1 \\ 2 & 5 & 4 \\ 6 & 2 & 5 \end{pmatrix}.$$

Cauchy matrices were briefly mentioned by Stinson [42] as a possible method of constructing AONTs. However, they are particularly relevant in light of the stronger definitions we are now investigating. To be specific, Cauchy matrices immediately yield the strongest possible all-or-nothing transforms, as stated in the following theorem.

Theorem 2.4.1. *Suppose q is a prime power and $q \geq 2s$. Then there is a linear transform that is simultaneously a (t, s, q)-AONT for all t such that $1 \leq t \leq s$.*

The following theorem proves the existence of AONTs, provided that an AONT with larger value of s exists.

Theorem 2.4.2. *If there exists a linear (t, s, q)-AONT with $t < s$, then there exists a linear $(t, s-1, q)$-AONT.*

Proof. Let M be a matrix for a linear (t, s, q)-AONT. Consider all the s possible $s-1$ by $s-1$ submatrices formed by deleting the first column and a row of M. We claim that at least one of these s matrices is invertible. For, if they were all non-invertible, then M would be non-invertible, by considering the cofactor expansion with respect the first column of M. □

We finish this subsection by showing that the existence of linear AONTs implies the existence of certain linear resilient functions.

Suppose q is a prime power. An (n, m, t, q)-resilient function f is *linear* if $f(\mathbf{x}) = \mathbf{x}M^T$ for some m by n matrix M defined over $\mathbb{F}_q$. Now, applying Theorem 1.4.3 we result in the following theorem.

Theorem 2.4.3. *The existence of a linear (t, s, q)-AONT implies the existence of a linear $(s, s-t, t, q)$-resilient function.*

Proof. Suppose that the s by s matrix M over $\mathbb{F}_q$ gives rise to a linear (t, s, q)-AONT. Then, from Lemma 2.3.1, every t by t submatrix of M is invertible. Construct an s by

t matrix M^* by deleting any $s-t$ rows of M. Clearly any t columns of M^* are linearly independent. Let $\mathcal{C}$ be the code generated by the rows of M^* and let $\mathcal{C}'$ be the dual code (i.e., the orthogonal complement of $\mathcal{C}$). It is well-known from basic coding theory (e.g., see [27, Chapter 1, Theorem 10]) that the minimum distance of $\mathcal{C}'$ is at least $t+1$. Let N be a generating matrix for $\mathcal{C}'$. Then N is an $s-t$ by s matrix over $\mathbb{F}_q$. Since N generates a code having minimum distance at least $t+1$, the function $f(\mathbf{x}) = \mathbf{x}N^T$ is a a (linear) $(s, s-t, t, q)$-resilient function (for a short proof of this fact, see [45, Theorem 1]). □

2.5 2-AONT

This section is comprised of the results for $t = 2$. The results are reported in categories based on their type, i.e., theoretical or computational, while maintaining a chronological order within the types. The exceptions to this pattern are the results based on the published results of this section, by other researchers; these results will appear last.

We begin with an existence proof for $(2, q-1, q)$-AONTs, for special prime values $p = q-1$.

Theorem 2.5.1. *Suppose $q = 2^n$, $q-1$ is prime and $s \leq q-1$. Then there exists a linear $(2, s, q)$-AONT over $\mathbb{F}_q$.*

Proof. Let $\alpha \in \mathbb{F}_q$ be a primitive element and let $M = (m_{r,c})$ be the s by s Vandermonde matrix in which $m_{r,c} = \alpha^{rc}$, $0 \leq r, c \leq s-1$. Clearly M is invertible, so we only need to show that any 2 by 2 submatrix is invertible. Consider a submatrix M' defined by rows i, j and columns i', j', where $i \neq j$ and $i' \neq j'$. We have

$$\det(M') = \alpha^{ii'+jj'} - \alpha^{ij'+ji'},$$

so $\det(M') = 0$ if and only if $\alpha^{ii'+jj'} = \alpha^{ij'+ji'}$, which happens if and only if

$$ii' + jj' \equiv ij' + ji' \bmod (q-1).$$

This condition is equivalent to

$$(i-j)(i'-j') \equiv 0 \bmod (q-1).$$

Since $q-1$ is prime, this happens if and only if $i = j$ or $i' = j'$. We assumed $i \neq j$ and $i' \neq j'$, so we conclude that M' is invertible. □

The above result requires that $2^n - 1$ is a (Mersenne) prime. The first few Mersenne primes occur for

$$n = 2, 3, 5, 7, 13, 31, 61, 89, 107, 127.$$

At the time this thesis was written, there were 51 known Mersenne primes, the largest being $2^{82589933} - 1$, which was discovered in December 2018 [1].

Example 2.5.1. *Let $n = 3$, then $q - 1 = 2^3 - 1 = 7$, the following Vandermonde matrix is the inverse of a $(2, 7, 8)$-AONT (here $\mathbb{F}_8$ is constructed using $x^3 + x + 1$ as the irreducible polynomial).*

$$M^{-1} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & x & x^2 & x+1 & x^2+x & x^2+x+1 & x^2+1 \\ 1 & x^2 & x^2+x & x^2+1 & x & x+1 & x^2+x+1 \\ 1 & x+1 & x^2+1 & x^2 & x^2+x+1 & x & x^2+x \\ 1 & x^2+x & x & x^2+x+1 & x^2 & x^2+1 & x+1 \\ 1 & x^2+x+1 & x+1 & x & x^2+1 & x^2+x & x^2 \\ 1 & x^2+1 & x^2+x+1 & x^2+x & x+1 & x^2 & x \end{pmatrix}$$

For a prime power q, if we ignore the requirement that a linear AONT is an invertible matrix, then constructing q by q matrices with invertible 2×2 submatrices is easy.

Theorem 2.5.2. *For any prime power q, there is a q by q matrix defined over $\mathbb{F}_q$ such that any 2 by 2 submatrix is invertible.*

Proof. $M = (m_{r,c})$ be the q by q matrix of entries from $\mathbb{F}_q$ defined by the rule $m_{r,c} = r + c$, where the sum is computed in $\mathbb{F}_q$. Consider a submatrix M' defined by rows i, j and columns i', j', where $i \neq j$ and $i' < j'$. We have

$$\det(M') = ij' + ji' - (ii' + jj'),$$

so $\det(M') = 0$ if and only if $ii' + jj' = ij' + ji'$. This condition is equivalent to

$$(i - j)(i' - j') = 0,$$

which happens if and only if $i = j$ or $i' = j'$. We assumed $i \neq j$ and $i' \neq j'$, so we conclude that M' is invertible. □

We note that the above construction does not yield an AONT for $q > 2$, because the sum of all the rows of the constructed matrix M is the all-zero vector and hence M is not invertible.

Example 2.5.2. *Let $q = 5$, then using the construction from Theorem 2.5.2 we have*

$$M = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 & 0 \\ 2 & 3 & 4 & 0 & 1 \\ 3 & 4 & 0 & 1 & 2 \\ 4 & 0 & 1 & 2 & 3 \end{pmatrix}$$

It is easy to check that the the sum of all the rows is a row of zeros.

When enumerating (t, s, q)-AONTs, we would like to avoid counting different forms of one construction, or what we call *equivalent* AONTs. Hence, we now discuss how to determine if two linear AONT are "equivalent". We define this notion as follows. Suppose M and M' are linear (t, s, q)-AONT. We say that M and M' are *equivalent* if M can be transformed into M' by performing a sequence of operations of the following type:

- row and column permutations,
- multiplying a row or column by a nonzero constant, and
- transposing the matrix.

In the following sections, we use this process to find inequivalent linear AONTs we obtain from analytical or computational searches.

To restrict the computer search for $(2, s, q)$-AONTs to one equivalence class for each AONT, we next define a *standard form* for linear AONTs. Suppose M is a matrix for a linear $(2, s, q)$-AONT. There can be at most one zero in each row and column of M; otherwise, M has a 2 by 2 submatrix containing those zeros that is not invertible. Then we can permute the rows and columns so that the 0's comprise the first μ entries on the main diagonal of M. If $\mu = 0$, then we can multiply rows and columns by nonzero field elements so that all the entries in the first row and first column consist of 1's. If $\mu \neq 0$, we can multiply rows and columns by nonzero field elements so that all the entries in the first row and first column consist of 1's, except for the entry in the top left corner, which is a 0. Such a matrix M is said to be of *type μ standard form*.

The following theorem proves an upper bound on the size, s, of the AONT for linear AONTs with a prime power alphabet size.

Theorem 2.5.3. *There is no linear $(2, q + 1, q)$-AONT for any prime power $q > 2$.*

Proof. Suppose M is a matrix for a linear $(2, q+1, q)$-AONT defined over $\mathbb{F}_q$. We can assume that M is in standard form. Consider the $q+1$ ordered pairs occurring in any two fixed rows of the matrix M. There are q symbols, which result in q^2 possible ordered pairs. However, the pair consisting of two zeros is ruled out, leaving q^2-1 ordered pairs. For two such ordered pairs $(i,j)^T$ and $(i',j')^T$, define $(i,j)^T \sim (i',j')^T$ if there is a nonzero element $\alpha \in \mathbb{F}_q$ such that $(i,j)^T = \alpha(i',j')^T$. Clearly $\sim$ is an equivalence relation, and there are $q+1$ equivalence classes, each having size $q-1$. We can only have at most one ordered pair from each equivalence class, otherwise the 2 by 2 submatrix resulting from those ordered pairs is not invertible. Hence, there are only $q+1$ possible pairs that can occur. Since there are $q+1$ columns, it follows that, from each of these $q+1$ equivalence classes, exactly one will be chosen. Therefore, each row must contain exactly one 0 and thus M is of type $q+1$ standard form.

From the above discussion, we see that M has the following structure:

$$\begin{pmatrix} 0 & 1 & 1 & 1 & \dots & 1 & 1 \\ 1 & 0 & & & & & \\ 1 & & 0 & & & & \\ 1 & & & 0 & & & \\ \vdots & & & & \ddots & & \\ 1 & & & & & 0 & \\ 1 & & & & & & 0 \end{pmatrix}.$$

Now consider the lower right q by q submatrix M' of M. Any element of $\mathbb{F}_q$ can occur on each column of M' at most once, otherwise M would have a singular 2 by 2 submatrix. Since each column of M' has q entries, each element of $\mathbb{F}_q$ appears exactly once on each column of M'. The sum of all the elements of a finite field $\mathbb{F}_q$ is equal to 0, provided that $q > 2$. Hence, the sum of all the rows in this matrix is an all-zero row. Therefore, regardless of the configuration of the remaining entries, the sum of the last q rows of M is the all-zero vector. Therefore, the matrix M is singular, which contradicts its being an AONT. □

Remark 2.5.1. *[14, Example 16] Linear $(2,3,2)$-AONTs do not exist. This covers the exception $q=2$ in Theorem 2.5.3.*

Theorem 2.5.3 and Remark 2.5.1 improve the bounds from Corollary 2.2.7, for linear AONTs.

Theorem 2.5.4. *If there is a linear $(2,s,q)$-AONT, then $s \leq q$.*

2.6 Linear $(2, q, q)$-AONT

Theorem 2.5.3 and Remark 2.5.1 showed that, in linear AONTs, $s \leq v$. Therefore, we next obtain some structural conditions for linear $(2, q, q)$-AONT in standard form.

Lemma 2.6.1. *Suppose M is a matrix for a linear $(2, q, q)$-AONT in standard form. Then M is of type q or type $q - 1$.*

Proof. Suppose that M is of type μ standard form, where $\mu \leq q - 2$. Then the last two rows of M contain no zeroes. We proceed as in the proof of Theorem 2.5.3. The q ordered pairs in the last two rows must all be from different equivalence classes. However, there are only $q - 1$ equivalence classes that do not contain a 0, so we have a contradiction. □

Therefore the standard form of a linear $(2, q, q)$-AONT looks like

$$M = \begin{pmatrix} 0 & 1 & 1 & 1 & \dots & 1 & 1 \\ 1 & 0 & & & & & \\ 1 & & 0 & & & & \\ 1 & & & 0 & & & \\ \vdots & & & & \ddots & & \\ 1 & & & & & 0 & \\ 1 & & & & & & \chi \end{pmatrix},$$

where $\chi = 0$ if and only if M is of type q and $\chi \neq 0$ if and only if M is of type $q - 1$.

2.6.1 Computer Searches for Small Linear $(2, q, q)$-AONT

For the rest of this section, we will focus on linear $(2, q, q)$-AONTs in standard form. Suppose M is a matrix for such an AONT. Define a linear ordering on the elements in the alphabet $\mathbb{F}_q$. If M also has the additional property that the entries in columns $3, \dots, q$ of row 2 are in increasing order (with respect to this linear order), then we say that M is *reduced*. Thus, the term "reduced" means that M is a linear $(2, q, q)$-AONT that satisfies the following additional properties:

- the diagonal of M consists of zeroes,
- the remaining entries in the first row and first column of M are ones, and

Table 2.3: Number of reduced and inequivalent linear $(2, q, q)$-AONT, for prime powers $q \leq 11$

q	reduced $(2, q, q)$-AONT	inequivalent $(2, q, q)$-AONT
3	2	1
4	3	2
5	38	5
7	13	1
8	0	0
9	0	0
11	21	1

- the entries in columns $3, \ldots, q$ of row 2 of M are in increasing order.

We implemented and executed exhaustive search algorithms that searched for reduced $(2, q, q)$-AONT, for all prime powers $q \leq 11$. The results are presented in Table 2.3.

One perhaps surprising outcome of our computer searches is that there are no linear $(2, q, q)$-AONT in type q standard form for $q = 8, 9$; however, it is easy to find examples of linear $(2, q-1, q)$-AONT for $q = 8, 9$.

For the prime orders $3, 5, 7, 11$, it turns out that there exists a reduced $(2, q, q)$-AONT having a very interesting structure, which we define here. Let M be a matrix for a reduced $(2, q, q)$-AONT. Let $\tau \in \mathbb{F}_q$. We say that M is *τ-skew-symmetric* if, for any pair of cells (i, j) and (j, i) of M, where $2 \leq i, j \leq q$ and $i \neq j$, it holds that $m_{ij} + m_{ji} = \tau$. Notice that this property implies that the matrix M contains no entries equal to τ because the only zero entries are located on the main diagonal. Another way to define the τ-skew-symmetric property is to say that $M_1 + {M_1}^T = \tau(J - I)$, where M_1 is formed from M by deleting the first row and column, J is the all-ones matrix and I is the identity matrix.

Our computer searches show that there is a $(q-1)$-skew-symmetric reduced $(2, q, q)$-AONT for $q = 3, 5, 7, 11$, as well as τ-skew-symmetric examples with various other values of τ.

Example 2.6.1. *A 2-skew-symmetric reduced linear $(2, 3, 3)$-AONT:*

$$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}.$$

Example 2.6.2. *A linear* $(2,4,4)$*-AONT, defined over the finite field* $\mathbb{F}_4 = \mathbb{Z}_2[x]/(x^2 + x + 1)$*:*

$$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & x \\ 1 & x & 0 & x+1 \\ 1 & 1 & x & 0 \end{pmatrix}.$$

Example 2.6.3. *A* 4*-skew-symmetric reduced linear* $(2,5,5)$*-AONT:*

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 2 & 3 \\ 1 & 3 & 0 & 1 & 2 \\ 1 & 2 & 3 & 0 & 1 \\ 1 & 1 & 2 & 3 & 0 \end{pmatrix}$$

Example 2.6.4. *A* 6*-skew-symmetric reduced linear* $(2,7,7)$*-AONT:*

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 2 & 3 & 4 & 5 \\ 1 & 5 & 0 & 3 & 4 & 2 & 1 \\ 1 & 4 & 3 & 0 & 5 & 1 & 2 \\ 1 & 3 & 2 & 1 & 0 & 5 & 4 \\ 1 & 2 & 4 & 5 & 1 & 0 & 3 \\ 1 & 1 & 5 & 4 & 2 & 3 & 0 \end{pmatrix}.$$

Example 2.6.5. *A linear* $(2,8,9)$*-AONT, defined over the finite field* $\mathbb{F}_9 = \mathbb{Z}_3[x]/(x^2+1)$*:*

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 2 & x & x+1 & x+2 & 2x \\ 1 & 1 & 0 & 2x+1 & x+1 & x+2 & 2 & x \\ 1 & 2x & x & 0 & x+2 & 2 & 2x+1 & x+1 \\ 1 & x+2 & 2 & x & 0 & 1 & 2x & 2x+1 \\ 1 & x+1 & x+2 & 2x & 2x+1 & 0 & 1 & 2 \\ 1 & x & x+1 & 1 & 2 & 2x+1 & 0 & x+2 \\ 1 & 2 & 2x+1 & x+1 & 1 & 2x & x & 0 \end{pmatrix}$$

Example 2.6.6. *A 10-skew-symmetric reduced linear* $(2, 11, 11)$*-AONT:*

$$\begin{pmatrix}
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
1 & 9 & 0 & 7 & 8 & 1 & 3 & 2 & 5 & 4 & 6 \\
1 & 8 & 3 & 0 & 2 & 5 & 6 & 1 & 9 & 7 & 4 \\
1 & 7 & 2 & 8 & 0 & 6 & 1 & 3 & 4 & 9 & 5 \\
1 & 6 & 9 & 5 & 4 & 0 & 8 & 7 & 3 & 1 & 2 \\
1 & 5 & 7 & 4 & 9 & 2 & 0 & 8 & 1 & 6 & 3 \\
1 & 4 & 8 & 9 & 7 & 3 & 2 & 0 & 6 & 5 & 1 \\
1 & 3 & 5 & 1 & 6 & 7 & 9 & 4 & 0 & 2 & 8 \\
1 & 2 & 6 & 3 & 1 & 9 & 4 & 5 & 8 & 0 & 7 \\
1 & 1 & 4 & 6 & 5 & 8 & 7 & 9 & 2 & 3 & 0
\end{pmatrix}$$

2.6.2 Additional Results on Linear AONT

Lastly, we will consider the existence of (t, s, q)-AONTs for different values of s and q, for the particular case of $t = 2$.

Given a prime power q and a positive integer t, we define

$$\mathcal{S}_t(q) = \{s : \text{there exists a linear } (t, s, q)\text{-AONT}\}.$$

Accordingly,

$$\mathcal{S}_2(q) = \{s : \text{there exists a linear } (2, s, q)\text{-AONT}\}.$$

From Remark 2.3.1, we have that $2 \in \mathcal{S}_2(q)$, so $\mathcal{S}_2(q) \neq \emptyset$. Also, from Theorem 2.5.3, Remark 2.5.1 and Theorem 2.4.2, there exists a maximum element in $\mathcal{S}_2(q)$, which we will denote by $M_2(q)$. Hence, based on Theorem 2.4.2, we know that a linear $(2, s, q)$-AONT exists for all s such that $2 \leq s \leq M_2(q)$.

In order to find $M_2(q)$ for some small values of q, we used an exhaustive computer search. In this algorithm, we confined our attention to reduced $(2, q, q)$-AONTs, as defined in Section 2.6. We have already showed that any linear $(2, q, q)$-AONT of type q standard form is equivalent to a reduced $(2, q, q)$-AONT. But it is possible that two reduced $(2, q, q)$-AONT could be equivalent. We next describe a useful process to test for equivalency of reduced $(2, q, q)$-AONT.

1. Pick two distinct rows r_1, r_2. Interchange rows 1 and r_1 of M and interchange rows 2 and r_2 of M. Then interchange columns 1 and r_1 and interchange columns 2 and r_2 of the resulting matrix.
2. Multiply columns $2, \ldots, q$ by constants to get $(0\ 1\ 1\ \cdots\ 1)$ in the first row.
3. Multiply rows $2, \ldots, q$ by constants to get $(011 \cdots 1)^T$ in the first column.
4. Permute columns $3, \ldots, q$ so the entries in row 2 in these columns are in increasing order (there is a unique permutation π that does this).
5. Apply the same permutation π to rows $3, \ldots, q$.
6. Transpose M and apply the first five steps to the transposed matrix.

Figure 2.4: Generating the reduced $(2, q, q)$-AONT that are equivalent to a given reduced $(2, q, q)$-AONT, M

The idea is to start with a specific reduced $(2, q, q)$-AONT, say M. Given M, we can generate all the reduced $(2, q, q)$-AONT that are equivalent to M. After doing this, it is a simple matter to examine any other reduced $(2, q, q)$-AONT, say M' and see if it occurs in the list of reduced $(2, q, q)$-AONT that are equivalent to M.

The algorithm presented in Figure 2.4 generates all the reduced $(2, q, q)$-AONT that are equivalent to M. After executing the first five steps, we have a list of $q^2 - q$ reduced $(2, q, q)$-AONT, each of which is equivalent to M (this includes M itself). After transposing the original matrix, we repeat the same five steps, which gives $q^2 - q$ additional equivalent AONT. The result is a list of $2q^2 - 2q$ equivalent AONT, but of course there could be duplications in the list.

2.6.3 Updated Results

Based on the results mentioned in this chapter, Wang et al. [50] subsequently proved the existence of linear $(2, p, p)$-AONTs for all prime values of p, using the following construction.

Construction 2.6.2. *[50, Construction 2.8] For a prime p, the $p \times p$ matrix M^{-1} is*

constructed as follows.

$$m_{ij} = \begin{cases} 0, & \text{if } i = j \\ 1, & \text{if } i > 0, j = 0 \\ \frac{1}{i-j}, & \text{otherwise} \end{cases} \tag{2.3}$$

Wang et al. [50, Theorem 2.3] also proved the non-existence of linear $(2, q, q)$-AONTs of type $q - 1$, where q is a prime power. Remark 2.6.1 is a direct result of this non-existence result and Lemma 2.6.1.

Remark 2.6.1. *[50] For any prime power q, only linear $(2, q, q)$-AONTs of type q may exist.*

We finish this chapter by summarizing our knowledge of linear 2-AONTs as the following theorem.

Theorem 2.6.3. *These three statements regarding upper and lower bounds on $M_2(q)$ are true:*

1. *$\lfloor q/2 \rfloor \leq M_2(q) \leq q$ for all prime powers q.*
2. *$M_2(q) \geq q - 1$ if $q - 1$ is a Mersenne prime.*
3. *$M_2(p) = p$ if p is a prime.*

Proof. For the first statement, we first note that for any prime power q and positive integer $s \leq q/2$, Cauchy matrices are instances of linear $(2, s, q)$-AONTs, so $\lfloor q/2 \rfloor \leq M_2(q)$. Theorem 2.5.3, on the other hand, states that $M_2(q) \leq q$. Therefore, $\lfloor q/2 \rfloor \leq M_2(q) \leq q$.

As discussed in Theorem 2.5.1, Vandermonde matrices can be used to achieve $(2, q - 1, q)$-AONTs, for Mersenne prime values of $q - 1$.

The third statement is directly derived from Construction 2.6.2 by Wang et al. [50].

□

2.7 Applications

This section discusses two applications of t-AONTs, namely,

1. an extension of Rivest's package transform, and
2. a new hash-based group signature scheme.

2.7.1 Extended Package Transform

As we discussed in Chapter 1, Rivest's *package transform* [38] was the first instance of using AONTs as a block cipher mode of operation. Hence, it is interesting to consider the performance of extended AONTs in this application. To use an unconditionally secure t-AONT as a package transform, the following scheme, presented in Algorithm 1, can be used. Suppose E_K is a semantically secure secret-key encryption scheme with key K.

Algorithm 1 Extended Package Transform

1: Divide the message into $s-1$ blocks: $m_1, m_2, \ldots, m_{s-1}$
2: Choose a random key K
3: Set x_i to $E_K(m_i)$ for $1 \leq i \leq s-t$
4: Set x_i to m_i for $s-t+1 \leq i \leq s-1$
5: Set x_s to K
6: Apply a (t, s, v)-AONT to the s-tuple $(x_1, x_2, \ldots, x_s)$, to get the output blocks $m'_1, \ldots, m'_s$ (v is the size of an alphabet that includes all possible ciphertexts.)
7: Output m'_i for $1 \leq i \leq s$

Now we provide an informal justification of security for this scheme. Please note that this scheme is not information theoretically secure. Using the scheme above, we cannot learn any information about any t input blocks (i.e., the x_i's) if we are missing t or more m'_i's. Hence, we cannot learn the key K nor any groups of t of the x_i's and in particular we will not be able to learn anything about the key K and the last $t-1$ of the x_i's, which were not encrypted in Algorithm 1. The first $s-t$ of the x_i's are encrypted m_i's using a secure encryption scheme, so it is not a problem if the adversary can determine the values of these x_i's.

The extended package transform can be used as a mode of operation. In this process, $m'_1, m'_2, \ldots, m'_s$ are created using the extended package transform. Then each m'_i is encrypted using a secret key K' to obtain the ciphertext block y_i, for $1 \leq i \leq s$. Figure 2.5 shows the process of applying extended package transform on a message for $t = 1$. This is roughly equivalent to Rivest's package transform scheme.

If the extended package transform is used as a mode of encryption, then any $s-t$ of the m'_i's do not yield any information about any t x_i's. In particular, if t of the m'_i's are missing, no information can be obtained about $x_{s-t+1}, x_{s-t}, \ldots, x_s$. Note that all the other x_i blocks are encrypted using K, but K is stored in x_s and is therefore unknown. Therefore, if an attacker is using an exhaustive key search to learn the message, they need

to decrypt at least $s - t + 1$ blocks to check whether the tested key is equal to K'. In other words, their search is slowed down at least by a factor of $s - t + 1$. For $t = 1$, this scheme performs similarly to Rivest's package transform [38]. Since only $s - t$ of the m_i's need to be encrypted, increasing t reduces the number of encryptions required; however, the security of the scheme is weakened at the same time because it must be assumed that the adversary can access at most $s - t$ of the s output blocks.

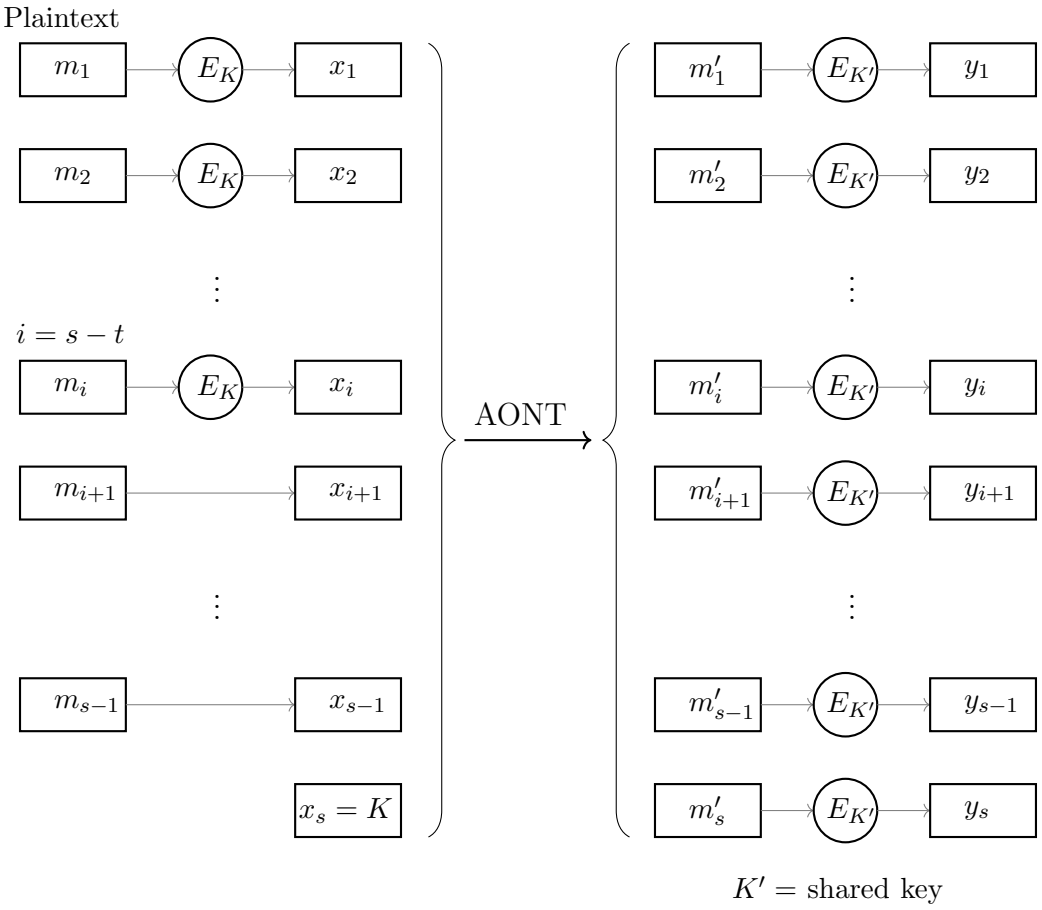


Figure 2.5: Different stages of extended package transform as a mode of encryption for a block cipher E

2.7.2 A New Hash-based Group Signature Scheme

In a joint work with Masoumeh Shafieinejad [39], we used AONTs in the context of hash-based group signature schemes. We will finish this chapter by briefly describing this application of t-AONTs.

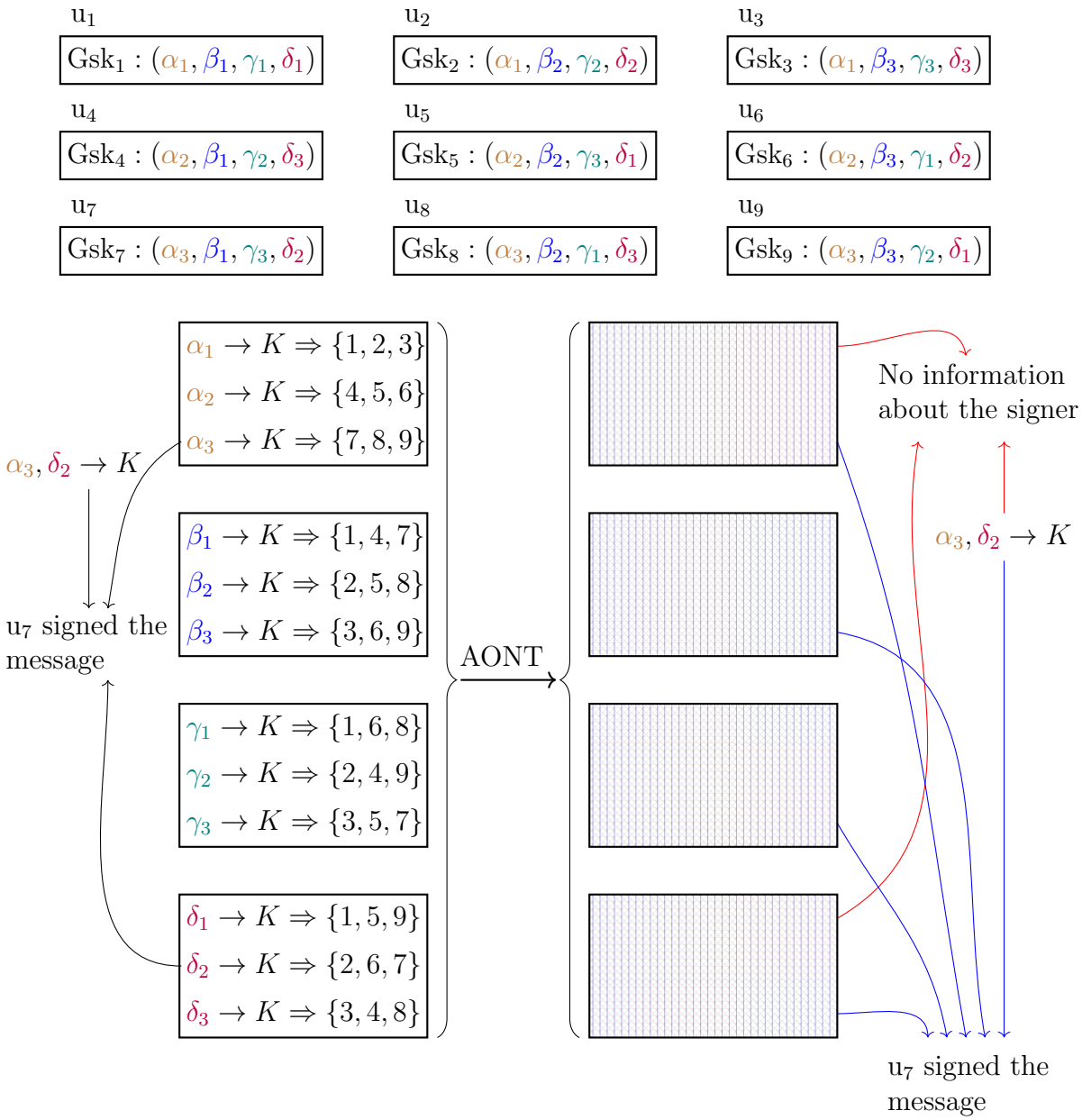
In hash-based signatures, the signer has a set of signing key elements, and the public verification keys are the hashes of those values. There is a bijection, ψ, from the set of possible messages to a family of subsets of these elements. To sign a message, the signer releases the message with its corresponding subset of signing key elements, using ψ. The verification process uses another bijection, ψ', from the set of possible messages to a family of subsets of hashed signing key elements. Both bijections have the same domain, and for any possible message m, $\psi'(m)$ is equal to the set that contains the hashed elements of $\psi(m)$. To verify a signature on message m, one needs to check whether the hash of the released elements matches $\psi'(m)$. A group signature allows any group member to anonymously sign a message on behalf of the group. A group signature scheme may allow an authorized group of openers to violate the anonymity property and identify the signer of signature.

This signature scheme uses a t-TD(s, n) (see Section 1.4.2 for the definition) to distribute secret signing keys among the n^t users and opening keys among the s openers. Each point in the design represents a signing-key element. The secret signing keys correspond to the blocks in the transversal designs and the opening keys are distributed according to the design groups. In this scheme, a signature requires at least t elements.

Recall that in a t-TD(s, n), any group and any block intersect at exactly one point, and every t points from t distinct design groups is contained in exactly one block. Therefore, each opener has the key corresponding to at most one signature element and each signature can be identified by one or more groups of t openers, depending on the number of signing key elements released for a signature. However, a group of t' openers, where $t' < t$, can use their opening keys to reduce the set of potential signers for a signature from n^t to $n^{t-t'}$. Shamir secret sharing scheme and AONTs were considered to prevent openers from obtaining this information [39].

Consider each opener's key as an input block and use a (t, s, v)-AONT on opening keys. The result is s output blocks, such that any subset of $s - t$ of these blocks does not reveal any information about any t-subset of the group designs, yielding no information about the signer. For $t < n/2$, this method can be used to increase the number of openers required to identify the signer from t to at least $n - t$ and at most n openers. It also prevents any coalition of fewer than $n - t$ openers to obtain any information about the signer. The

advantage of this method to secret sharing is that it does not require any extra storage, and the total storage cost remains the same.



The diagram above presents the application of a 2-AONT on the group opening keys of the signature scheme. As the diagram shows, each user of the signature scheme is given a set of secret values, Gsk_i. The signer can derive the signing key from these secret values. $x \rightarrow K$ denotes that secret value x is used in the derivation of a key K. Note that without an AONT, in the presented setting, only two openers are required to identify the signer, but if the group opening keys, Gok's, are transformed using a 2-AONT, openers cannot learn anything about the signer, if they are missing at least 2 output blocks; however, if they have all the output blocks, they can correctly identify the signer.

Chapter 3

Almost AONT

In this chapter, we will focus on linear transforms that satisfy the condition (3) for some, but not necessarily all, pairs of $\mathcal{X}, \mathcal{Y}$. In particular, we will explore the parameter sets for which the t by t submatrices of invertible matrices cannot all be invertible. For example, there is no s by s Cauchy matrix over $\mathbb{F}_2$ if $s > 1$. In fact, Stinson [42] showed that there is no linear $(1, s, 2)$-AONT if $s > 1$. This is because every entry of M must equal 1 (in order that the 1 by 1 submatrices of M are invertible), but then M itself is not invertible. This motivates trying to determine how close we can get to a $(1, s, 2)$-AONT, or more generally, to a $(t, s, 2)$-AONT, for a given t, $1 \leq t \leq s$. This will be particularly relevant in the case where ϕ is a binary linear transform. More specifically, suppose $q = 2^r$ for some $r \geq 1$ and M is defined over the subfield $\mathbb{F}_2$ (so M is a 0 - 1 matrix). This could be desirable from an efficiency point of view, because the only operations required to compute the transform are exclusive-ors of bit-strings. Hence it is a reasonable and interesting problem to study how close we can get to an AONT with regards to the number of invertible t by t submatrices.

The content of this chapter are from collaborations with Paolo D'Arco [14] and Doug Stinson [14, 31]. To present our results, first, we introduce measures for evaluating the closeness of an invertible matrix to an AONT. Then, we study the linear transforms over $\mathbb{F}_2$ and then $\mathbb{F}_3$ as almost AONT structures. We will give optimum results for $t = 1$; much of the rest of this chapter will study the case where $t = 2$. Our study includes both theoretical and computational results.

3.1 Closeness to AONT

As we stated above, the focus of this chapter is on the linear transforms that satisfy the AONT condition for as many submatrices as possible. Therefore, to quantify the "closeness" of a linear transform M to an all-or-nothing transform we consider the ratio of the number of invertible t by t square submatrices to the total number of t by t square submatrices. Hence, for an s by s invertible matrix M with elements in $\mathbb{F}_q$ and for $1 \leq t \leq s$, we define

$$N_t(M) = \text{number of invertible } t \text{ by } t \text{ submatrices of } M$$

and

$$R_t(M) = \frac{N_t(M)}{\binom{s}{t}^2}.$$

We refer to $R_t(M)$ as the *t-density* of the matrix M. For example, if M is a linear t-AONT, then $R_t(M) = 1$. For $1 \leq t \leq s$, we also define

$$N_{t,q}(s) = \max\{N_t(M) : M \text{ is an } s \text{ by } s \text{ invertible matrix over } \mathbb{F}_q\}$$

$$R_{t,q}(s) = \max\{R_t(M) : M \text{ is an } s \text{ by } s \text{ invertible matrix over } \mathbb{F}_q\}.$$

$R_{t,q}(s)$ denotes the maximum t-density of any s by s invertible matrix with elements in $\mathbb{F}_q$.

The following example offers a more tangible understanding of almost AONT structures.

Example 3.1.1. *Consider the following* 3 *by* 3 *matrix over* $\mathbb{F}_2$*:*

$$M = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}.$$

M is an invertible matrix, and there are nine 1 *by* 1 *submatrices of M and seven of them are invertible, i.e., non-zero. Therefore,* $R_1(M) = \frac{7}{9}$.

Finally, for a fixed set of invertible matrices, we refer to the maximum number of invertible t by t submatrices in and maximum t-density of a matrix in that set by N_t and R_t, respectively.

3.2 Linear Transforms over $\mathbb{F}_2$

To study the linear almost AONTs over $\mathbb{F}_2$, we begin by reviewing the invertible 2×2 binary matrices.

Fact 3.2.1. *A 2 by 2 0 - 1 matrix is invertible if and only if it is one of the following six matrices:*

$$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

Example 3.2.1 shows how $R_{2,2}(s)$ is calculated for a 3 by 3 matrix from Example 3.1.1.

Example 3.2.1. *Consider the following 3 by 3 matrix:*

$$M = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}.$$

There are nine 2 by 2 submatrices of M and seven of them are seen to be invertible, from Fact 3.2.1. The only non-invertible 2 by 2 submatrices are $M(\{1,3\},\{1,2\})$ and $M(\{1,2\},\{1,3\})$. Finally, M itself is invertible. Hence, $R_2(M) = 7/9$.

In the next step, we will find the value of $R_{1,2}(s)$, using the following lemmas.

Lemma 3.2.2. *Suppose $M = J_s - I_s$, where I_s denotes the s by s identity matrix and J_s denotes the s by s matrix in which every entry is equal to one. Then M is invertible over $\mathbb{F}_2$ if and only if s is even.*

Proof. If s is even, then it is easy to check that $M^{-1} = M$. If s is odd, then observe that the sum of all the columns of M yields the zero-vector, so the columns of M are linearly dependent. □

Lemma 3.2.3. *Suppose M is an s by s 0 - 1 matrix with at most $s-1$ zero entries. Then M is invertible over $\mathbb{F}_2$ if and only if the zero entries occur in $s-1$ different rows and in $s-1$ different columns.*

Proof. If M has at most $s-2$ zero entries, then there must exist at least two columns of M that do not contain a zero entry. These two columns are identical, so they are linearly dependent.

Now, suppose that M has exactly $s-1$ zero entries. If there are at least two zero entries in a specific column of M, then there must exist at least two columns of M that do not contain a zero entry, and therefore, M is not invertible. A similar conclusion holds if there exist at least two zero entries in a specific row of M. Therefore, we can restrict our attention to the case where the zero entries occur in $s-1$ different rows and in $s-1$ different columns. We will show that M is invertible in this case.

By permuting rows and columns,which does not affect invertibility, if necessary, we can assume that $M = (m_{ij})$ has the form

$$M = \begin{pmatrix} 1 & 1 & 1 & 1 & \dots & 1 \\ 1 & 0 & 1 & 1 & \dots & 1 \\ 1 & 1 & 0 & 1 & \dots & 1 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & 1 & \dots & 0 \end{pmatrix}. \tag{3.1}$$

It is easy to see that the s by s matrix M is invertible by verifying the following formula for M^{-1}:

$$M^{-1} = \begin{pmatrix} a & \mathbf{1}^T \\ \mathbf{1} & I \end{pmatrix},$$

where $\mathbf{1}$ is a column vector consisting of $s-1$ ones, I is an $s-1$ by $s-1$ identity matrix, and $a = s \bmod 2$.

□

The following result is an immediate corollary of Lemma 3.2.3.

Theorem 3.2.4. *For all $s \geq 1$, we have $R_{1,2}(s) = 1 - (s-1)/s^2$.*

It was shown by Stinson [42] that $R_{1,2}(s) \geq 1 - (1/s)$ when s is even. This was based on using the matrix $J_s - I_s$ as a transform and Lemma 3.2.3. Theorem 3.2.4 is a slight improvement, and it holds for all values of s.

Example 3.2.2. *Consider the 4 by 4 matrix given by (3.1):*

$$M = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}.$$

Here, we can verify using Lemmas 3.2.2 and 3.2.3 that $R_{1,2}(M) = 13/16$, $R_{2,2}(M) = 24/36 = 2/3$ *and* $R_{3,2}(M) = 9/16$.

In fact, it is possible to compute all the values $R_t(M)$ for the s by s matrix M given in (3.1). There are $\binom{s}{t}^2$ submatrices N of M of dimensions t by t. From the structure of M, and from Lemmas 3.2.2 and 3.2.3, we see that a t by t submatrix N is invertible if and only if one of the following conditions holds:

1. N contains $t-1$ zero entries, or
2. t is even and N contains t zero entries.

If we can count the number of submatrices of this form, then we can compute $R_t(M)$. The following lemmas do this.

Lemma 3.2.5. *The s by s matrix M given in (3.1) has exactly $\binom{s-1}{t-1}(1+(s-t+1)(s-t))$ submatrices that contain exactly $t-1$ zero entries.*

Proof. We divide the desired submatrices into two sets and count the number of submatrices in each set separately. First, consider all the submatrices with $t-1$ zeros that intersect with the first row: there are $\binom{s-1}{t-1}$ ways to choose the other t rows, and for each one there are $s-t+1$ ways to choose the columns. Hence, there are $\binom{s-1}{t-1}(s-t+1)$ such submatrices that include elements from the first row.

Next, we count all the submatrices with $t-1$ zeros that do not intersect with the first row. There are $\binom{s-1}{t-1}$ ways to choose the zeros, fixing $t-1$ rows and $t-1$ columns. For each choice of $t-1$ zeros, there are $s-t$ ways to choose the other row and $s-t$ ways to choose the other column. Thus, there are $\binom{s-1}{t-1}(s-t)(s-t)$ submatrices with $t-1$ zeros that do not intersect with the first row.

Therefore, an s by s matrix of the form 3.1 in total has

$$\begin{aligned}\binom{s-1}{t-1}((s-t)(s-t)+(s-t+1)) &= \binom{s-1}{t-1}((s-t+1-1)(s-t)+(s-t+1)) \\ &= \binom{s-1}{t-1}((s-t+1)(s-t)-(s-t)+(s-t)+1) \\ &= \binom{s-1}{t-1}(1+(s-t)(s-t+1))\end{aligned}$$

t by t submatrices with $t-1$ zeros. □

Lemma 3.2.6. *The s by s matrix M given in (3.1) has exactly $\binom{s-1}{t}$ submatrices that contain exactly t zero entries.*

Proof. The submatrices should have t zeros, and each zero fixes a row and a column. Therefore, we only need to choose t zeros out of $s-1$ zeros. Hence, M has $\binom{s-1}{t}$ such submatrices. □

From Lemma 3.2.5 and Lemma 3.2.6, we obtain the following theorem.

Theorem 3.2.7. *Let M be the s by s matrix given in (3.1) and let $1 \leq t \leq s-1$. If t is odd, then*

$$N_t(M) = \binom{s-1}{t-1}(1 + (s-t+1)(s-t)).$$

If t is even, then

$$N_t(M) = \binom{s-1}{t} + \binom{s-1}{t-1}(1 + (s-t+1)(s-t)).$$

Theorem 3.2.7 also provides (constructive) lower bounds on $R_{t,2}(s)$ for all values of $t \leq s$. We do not claim that these bounds are necessarily good asymptotic bounds, however. Even for $t = 2$, we get $R_{2,2}(M) \to 0$ as $s \to \infty$, since $\binom{s-1}{t-1}(1 + (s-t+1)(s-t)) \in \theta(s^3)$ and $\binom{s}{t}^2 \in \theta(s^4)$. This suggests looking for constructions which will yield constant lower bounds on $R_{2,2}(s)$. On the other hand, good upper bounds on $R_{2,2}(s)$ can help evaluate the constructions. Therefore, we will continue with some upper bounds $R_{2,2}(s)$, followed by different construction methods and some results.

3.2.1 Upper Bounds for $R_{2,2}(s)$

We first establish an easy upper bound for $R_{2,2}(s)$. This bound is a consequence of the following lemma.

Lemma 3.2.8. *Any 2 by s 0 - 1 matrix contains at most $s^2/3$ invertible 2 by 2 submatrices.*

Proof. Let N be any 2 by s 0 - 1 matrix. Consider the 2 by 1 submatrices of N. Suppose there are a_0 occurrences of $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$, a_1 occurrences of $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$, a_2 occurrences of $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$, and a_3 occurrences of $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$. Of course $a_0 + a_1 + a_2 + a_3 = s$. From Fact 3.2.1, the number of

invertible 2 by 2 submatrices in N is easily seen to be $a_1a_2 + a_1a_3 + a_2a_3$. This expression is maximized when $a_0 = 0$, $a_1 = a_2 = a_3 = s/3$, yielding $3(s/3)^2 = s^2/3$ invertible 2 by 2 submatrices if we allow a_i's to be rational numbers. □

Theorem 3.2.9. *For any $s \geq 2$, it holds that*

$$R_{2,2}(s) \leq \frac{2s}{3(s-1)}.$$

Proof. From Lemma 3.2.8, in any two rows of M there are at most $s^2/3$ invertible 2 by 2 submatrices. Now, in the entire matrix M, there are $\binom{s}{2}$ ways to choose two rows, and there are $\binom{s}{2}^2$ submatrices of order 2. This immediately yields

$$R_{2,2}(s) \leq \frac{\binom{s}{2}\left(\frac{s^2}{3}\right)}{\binom{s}{2}^2} = \frac{2s}{3(s-1)}.$$

□

Example 3.2.3. *When $s = 3$, we only get the trivial upper bound $R_{2,2}(3) \leq 1$ from Theorem 3.2.9. Consider the matrix*

$$M_0 = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}.$$

It is clear from the proof of Theorem 3.2.9 that all nine 2 by 2 submatrices of M_0 are invertible, and M_0 is the only 3 by 3 matrix with this property. However, M_0 is not itself invertible, so we can conclude that $R_{2,2}(3) \leq 8/9$. Example 3.2.1 shows that $R_{2,2}(3) \geq 7/9$.

In fact, we can show that $R_{2,2}(3) = 7/9$. Suppose that $R_{2,2}(3) = 8/9$. Let M be a 3 by 3 matrix such that $R_{2,2}(M) = 8/9$. Then we can assume that the first two rows of M contain three invertible 2 by 2 submatrices, the first and third rows of M contain three invertible 2 by 2 submatrices, and the last two rows of M contain two invertible 2 by 2 submatrices. By permuting columns, the first two rows of M look like:

$$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}.$$

In order that the first and third rows contain three invertible 2 by 2 submatrices, the third row must be 101 or 110. In the first case, the last two rows of M contain no invertible 2 by

2 *submatrices, and in the second case, the last two rows of* M *contain three invertible* 2 *by* 2 *submatrices, but in this case* $M = M_0$*. We conclude that* $R_{2,2}(3) < 8/9$*, so* $R_{2,2}(3) = 7/9$*.*

Example 3.2.4. *When* $s = 4$*, we get* $R_{2,2}(4) \leq 8/9$ *from Theorem 3.2.9. Consider the matrix* $M = J_4 - I_4$*.* M *is invertible from Lemma 3.2.2. It is easy to check that* 30 *of the* 2 *by* 2 *submatrices of* M *are invertible. Therefore,* $R_{2,2}(4) \geq 5/6$*.*

We can in fact show that $R_{2,2}(4) = 5/6$*, as follows. Suppose* $R_{2,2}(4) > 5/6$*. Then there is a* 4 *by* 4 0 - 1 *matrix* M *having at least* 31 *invertible* 2 *by* 2 *submatrices. There are six pairs of rows in* M*, and* $31 > 6 \times 5$*, so there is at least one pair of rows that contains six invertible* 2 *by* 2 *submatrices. But this contradicts Lemma 3.2.8, where it is shown that the maximum number of* 2 *by* 2 *submatrices in two given rows is at most* $4^2/3 = 16/3 < 6$*.*

We next present a generalization of Theorem 3.2.9 that leads to an improved upper bound on $R_{2,2}(s)$. The proof of Theorem 3.2.9 was based on upper-bounding the number of invertible 2 by 2 submatrices in any two rows of an s by s matrix M. Here we instead determine an upper bound on the number of invertible 2 by 2 submatrices in any four rows of M. (It turns out that considering three rows at a time yields the same bound as Theorem 3.2.9, so we skip directly to an analysis of four rows at a time.)

Label the vectors in $\{0,1\}^4$ in lexicographic order as follows: $b_0 = (0,0,0,0)$, $b_1 = (0,0,0,1)$, $b_2 = (0,0,1,0)$, $b_3 = (0,0,1,1)$, $\ldots$, $b_{15} = (1,1,1,1)$. For $1 \leq i,j \leq 15$, define c_{ij} to be the number of invertible 2 by 2 submatrices in the 4 by 2 matrix $\left(\begin{array}{c|c} b_i^T & b_j^T \end{array} \right)$. Let $C = (c_{ij})$; note that C is a 15 by 15 symmetric matrix with zero diagonal such that every off-diagonal element is a positive integer. This matrix C is straightforward to compute and it is presented in Figure 3.1.

Now define $\mathbf{z} = (z_1, \ldots, z_{15})$ and consider the following quadratic program $\mathcal{Q}$:

Maximize	$\frac{1}{2}\mathbf{z}C\mathbf{z}^T$
subject to	$\sum_{i=1}^{15} z_i \leq 1$ and $z_i \geq 0$, for all i, $1 \leq i \leq 15$.

We have the following result.

Theorem 3.2.10. *For any integer* $s \geq 4$*, it holds that*

$$R_{2,2}(s) \leq \frac{\gamma s}{3(s-1)},$$

where γ *denotes the optimal solution to* $\mathcal{Q}$*.*

$$C = \begin{pmatrix}
0 & 1 & 1 & 1 & 1 & 2 & 2 & 1 & 1 & 2 & 2 & 2 & 2 & 3 & 3 \\
1 & 0 & 1 & 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 2 & 3 & 2 & 3 \\
1 & 1 & 0 & 2 & 3 & 3 & 2 & 2 & 3 & 3 & 2 & 4 & 5 & 5 & 4 \\
1 & 1 & 2 & 0 & 1 & 1 & 2 & 1 & 2 & 2 & 3 & 1 & 2 & 2 & 3 \\
1 & 2 & 3 & 1 & 0 & 3 & 2 & 2 & 3 & 4 & 5 & 3 & 2 & 5 & 4 \\
2 & 1 & 3 & 1 & 3 & 0 & 2 & 2 & 4 & 3 & 5 & 3 & 5 & 2 & 4 \\
2 & 2 & 2 & 2 & 2 & 2 & 0 & 3 & 5 & 5 & 5 & 5 & 5 & 5 & 3 \\
1 & 1 & 2 & 1 & 2 & 2 & 3 & 0 & 1 & 1 & 2 & 1 & 2 & 2 & 3 \\
1 & 2 & 3 & 2 & 3 & 4 & 5 & 1 & 0 & 3 & 2 & 3 & 2 & 5 & 4 \\
2 & 1 & 3 & 2 & 4 & 3 & 5 & 1 & 3 & 0 & 2 & 3 & 5 & 2 & 4 \\
2 & 2 & 2 & 3 & 5 & 5 & 5 & 2 & 2 & 2 & 0 & 5 & 5 & 5 & 3 \\
2 & 2 & 4 & 1 & 3 & 3 & 5 & 1 & 3 & 3 & 5 & 0 & 2 & 2 & 4 \\
2 & 3 & 5 & 2 & 2 & 5 & 5 & 2 & 2 & 5 & 5 & 2 & 0 & 5 & 3 \\
3 & 2 & 5 & 2 & 5 & 2 & 5 & 2 & 5 & 2 & 5 & 2 & 5 & 0 & 3 \\
3 & 3 & 4 & 3 & 4 & 4 & 3 & 3 & 4 & 4 & 3 & 4 & 3 & 3 & 0
\end{pmatrix}.$$

Figure 3.1: The objective function C for the quadratic program

Proof. Let M be any s by s 0 - 1 matrix. Consider any four rows of M, say the first four rows without loss of generality, and denote the resulting 4 by s submatrix by M'. For $0 \leq i \leq 15$, suppose there are a_i columns of M' that are equal to b_i^T. The number N of 2 by 2 invertible submatrices of M' is equal to $\frac{1}{2}\mathbf{a}C\mathbf{a}^T$, where $\mathbf{a} = (a_1, \ldots, a_{15})$ (we can ignore a_0 because a zero column does not give rise to any invertible submatrices). If we now define $z_i = a_i/s$ for all i, then we obtain

$$N = \frac{1}{2}\mathbf{a}C\mathbf{a}^T = \frac{s^2}{2}\mathbf{z}C\mathbf{z}^T \leq \gamma s^2.$$

There are $\binom{s}{4}$ ways to choose four rows from M. The total number of occurrences of invertible 2 by 2 submatrices obtained is at most $\binom{s}{4}\gamma s$. However, each invertible 2 by 2 submatrix is included in exactly $\binom{s-2}{2}$ sets of four rows, so the total number of invertible 2 by 2 submatrices is at most

$$\frac{\binom{s}{4}\gamma s^2}{\binom{s-2}{2}}.$$

The total number of 2 by 2 submatrices is $\binom{s}{2}^2$, so we obtain the upper bound

$$R_{2,2}(s) \leq \frac{\binom{s}{4}\gamma s^2}{\binom{s-2}{2}\binom{s}{2}^2} = \frac{\gamma s}{3(s-1)}. \tag{3.2}$$

□

3.2.1.1 Computational Results

In general, it can be difficult to find (globally) optimal solutions for quadratic programs. We were able to solve our quadratic program $\mathcal{Q}$ using the BARON software [46] on the NEOS server (`http://www.neos-server.org/neos/`). The result is that $\gamma = 15/8$ and an optimal solution is given by $z_7 = z_{11} = z_{13} = z_{14} = 1/4$, and $z_i = 0$ if $i \notin \{7, 11, 13, 14\}$. It is interesting to observe that this solution corresponds to the given set of four rows containing only columns consisting of three 1's and one 0. In fact, when $s = 4$, this provides an alternative proof of Example 3.2.4.

Applying Theorem 3.2.10, we immediately obtain the following improved upper bound.

Corollary 3.2.11. *For any $s \geq 4$, it holds that*

$$R_{2,2}(s) \leq \frac{5s}{8(s-1)}.$$

This upper bound is asymptotically equal to 5/8, which is an improvement over the asymptotic upper bound of 2/3 obtained from Theorem 3.2.9.

It is of course possible to generalize this approach, by considering ρ rows at a time. The coefficient matrix C will have $2^\rho - 1$ rows and columns. If γ_ρ denotes the solution to the related quadratic program, then we obtain the following generalization of Theorem 3.2.10.

Theorem 3.2.12. *For any integers $2 \leq \rho \leq s$, it holds that*

$$R_{2,2}(s) \leq \frac{4\gamma_\rho}{\rho(\rho-1)} \times \frac{s}{s-1}. \tag{3.3}$$

Proof. The equation (3.2) becomes the following:

$$R_{2,2}(s) \leq \frac{\binom{s}{\rho}\gamma_\rho s^2}{\binom{s-2}{\rho-2}\binom{s}{2}^2} = \frac{4\gamma_\rho}{\rho(\rho-1)} \times \frac{s}{s-1}.$$

□

The difficulty in obtaining improved bounds using this approach is that the optimal solutions γ_ρ of the quadratic programs are hard to compute.

3.2.2 Computational Constructions

This subsection covers the different computational methods we employed to search for invertible matrices over $\mathbb{F}_2$ with high R_2 values.

3.2.2.1 Exhaustive Search

The first search algorithm utilizes the quadratic programming formulation of 2-AONTs given by D'Arco et al. [14], presented earlier in Section 3.2.1.

The matrix C is defined such that c_{ij} is the number of invertible 2×2 submatrices formed by considering binary representations of i and j as two rows. Thus, the main diagonal of C is all zeros. Since the trace of C equals 0 and C is not an all-zero matrix, it has both positive and negative eigenvalues. Therefore, the matrix C is not positive/negative semi-definite for any value of s. According to Vavasis [49, p. 81], the QP problem for such matrices is NP-hard, in general. Consequently, an exhaustive search algorithm was used to search for an instance of invertible $s \times s$ matrices with the maximum possible number of invertible 2×2 submatrices, for $4 \leq s \leq 9$. The algorithm used a branch-and-bound technique that branches by iterating over different possible combinations for each row, in nested loops, and bounds the search as soon as the rows of the matrix become linearly dependent.

3.2.3 Exhaustive Search

In total, there are 2^{n^2} different binary $n \times n$ matrices; however many of them can be skipped because either they are not invertible, or a permutation of their rows and columns has already been considered. In the search algorithm, each `for` loop iterates over the possible values for a row. At each iteration, if a row is a linear combination of the rows above it, that row will not be considered, i.e., the search will be bounded by this linear dependency check. Besides, since any permutation of rows and columns does not affect either the singularity, or the number of invertible 2×2 submatrices of a matrix, we want

to enumerate as few matrices of each class as possible. Therefore, the search algorithm only generated matrices in which each row has at least as many 1's as the number of 1's in the row above it. Also, if two rows have the same number of 1's, the row, representing a smaller number in binary, should appear higher. These two rules enabled us to search only a $1/s!$ fraction of the search domain. Finally, we partially restricted column permutations by fixing all the 1's in the first row to be the right-most coordinates. This constraint helped the algorithm to skip repeated computations over different permutations of the first row, for a fixed number of 1's.

Example 3.2.5. *Let $s = 4$. Then following the bounding constraints in the backtrack algorithm search, the first row can only be chosen from:* $(0,0,0,1), (0,0,1,1)$, $(0,1,1,1)$, *and* $(1,1,1,1)$. *Suppose* $(0,0,1,1)$ *is the chosen as the first row, then neither* $(0,0,0,1)$, $(0,0,1,0), (0,1,0,0)$, *nor* $(1,0,0,0)$ *can be chosen as the second row because their weights are smaller than that of the first row. Now, if the second row is* $(1,0,0,1)$, *then the third row cannot be* $(0,1,1,0)$ *as it represents 6 in binary which is smaller than 9, which is represented by the second row; also, it cannot be* $(1,0,1,0)$ *because it is a linear combination of the first two rows.*

To be able to use the algorithm for the case when $s = 9$, another restriction was added: for the first 5 rows, if two coordinates in a row have different values, but in all the rows above them, those coordinates were identical, in this row the value 0 should appear on the left side of the value 1. Since the order of rows and columns does not impact the existence the existence of a linear AONT, and the other combination will present at another iteration, this restriction imposes no loss of generality.

The computations for $4 \leq s \leq 8$ were executed on one node on a server of the Cheriton School of Computer Science, `linux.cs.uwaterloo.ca`, with a 64 bit AMD CPU, having a 2.6 GHz clock rate. We also attempted to use the same algorithm distributed over 256 processors on `grex.westgrid.ca`. But 14 of those processes did not terminate by the end of the 96 hour time limit. The search domain, corresponding to those processes, was distributed again among 266 processes. In total, the whole computation took approximately 10000 CPU hours.

Some information about the resulting matrices is provided below. Also, the pseudocode in Algorithm 1 illustrates the general algorithm used in the processes. For the $s = 9$ case, different iterations of the second `for` loop were distributed among 256 processes on `grex.westgrid.ca`, two iterations per process.

Table 3.1 presents the number of 1's, their density, and minimum weight of a row in the resultant matrices, along with the values for s, $N_2(s)$, and $R_{2,2}(s)$.

Algorithm 2 Exhaustive search(d, i, x, comb)// Matrix C is available globally

```
1: soln: An array containing the best answer during the current function call.
2: N: Highest number of invertible 2 by 2 submatices at the current function call
3: b_j is the binary presentation of j in s bits
4: if d = 0 then
5:    x ← 0
6:    Initialize comb to an array of all 0's
7:    for i : 1 → 2^s − s do
8:       comb[1] = i
9:       (x, soln) ← Exhaustive search (d + 1, i, 0, comb)
10:      if x > N then
11:         N ← x
12:         soln ← comb
13:      end if
14:   end for
15: else
16:   for j : i + 1 → 2^s − s + d do
17:      comb[d] = j
18:      y ← x
19:      if weight(i) ≤ weight(j) then
20:         Add b_j to the matrix as a row.
21:         if all the rows are linearly independent then
22:            for k : 1 → d do
23:               y ← y + C[comb[k]][j]
24:            end for
25:            if d < s then
26:               (y, soln) ← Exhaustive search (d + 1, j + 1, y, comb)
27:               if y > N then
28:                  N ← y
29:                  soln ← comb
30:               end if
31:            end if
32:         end if
33:         Remove b_j from the matrix
34:      end if
35:   end for
36:   return (N, soln)
37: end if
```

Table 3.1: $N_{2,2}(s)$ and $R_{2,2}(s)$ submatrices for $s = 3, \ldots, 9$.

s	$N_{2,2}(s)$	$R_{2,2}(s)$	1's in the matrix	density of 1's	Min weight of a row
3	7	$0.\bar{7}$	7	$0.\bar{7}$	2
4	30	$0.8\bar{3}$	12	0.75	3
5	70	0.7	17	0.68	3
6	150	$0.\bar{6}$	25	$0.69\bar{4}$	4
7	287	≈ 0.651	35	≈ 0.714	5
8	485	≈ 0.618	47	≈ 0.734	5
9	783	≈ 0.604	55	≈ 0.679	6

3.2.4 Search for Cyclic Matrices

Based on the idea of constructing almost 2-AONTs using cyclotomy in [14] and [53], and also due to computational limitations of enumerating all possible matrices, we decided to limit the search to cyclic matrices. The algorithm for this search iterates over different possible values for the first row of the matrix, and each of the other rows will be generated by applying a cyclic shift to the row above.

In order to search all the invertible $s \times s$ cyclic matrices for the one with the greatest $R_2(s)$ value, it suffices to iterate through all possible permutations for the first row. This is because the order of rows does not affect the value of $R_2(s)$. It also guarantees that the first row can be substituted by any of its cyclic shifts. Hence, only one representative of each class of rows, resulting by shifting the first row, need to be examined. For each choice of the first row, the number of invertible 2×2 submatrices generated by that row and any of its shifts are counted and multiplied by $s/2$, in order to compute the total number of invertible 2×2 submatrices for that cyclic matrix. The algorithm keeps the row resulting in the best ratio found so far, and reports that row at the end of the search.

The cyclic search program, for $3 \leq s \leq 36$, was sequentially executed on a node on `linux.cs.uwaterloo.ca` in about 14 hours. Table 3.2 demonstrates the results of the cyclic search and the exhaustive search, as far as possible, for the sake of comparison.

As Table 3.2 shows, that $s = 2$, $s = 4$, and $s = 7$ are the only cases, as far as we can compare, where both algorithms generate similar results.

3.2.5 Search for Almost Cyclic Matrices

As previously mentioned, a cyclic search may fail to find some solutions near the optimal solution. This limitation can be attributed to the restriction on the matrices to be cyclic. Since being cyclic is not an intrinsic property of AONTs, the condition can be relaxed so that the search considers matrices that are almost cyclic and have large 2-density. To do so, we developed a modification of the adjusting step by Zhang et al. [53] (their algorithm will be presented in Subsection 3.2.8). The algorithm searches the matrices that are cyclic or off-by-one from being cyclic, i.e., a cyclic matrix with one entry altered from 0 to 1 or from 1 to 0, from the matrix with the maximum 2-density.

The search algorithm enumerates the matrices in the same method that the cyclic search does; however, it does not consider the independence of the rows as a necessary condition if the rank of the resultant matrix is $n - 1$. Instead, the algorithm tries flipping each of the entries in the last row of the matrix, one at a time, and checks the independence of the rows and the number of invertible 2×2 submatrices of each of the new matrices. The 2-densities of these matrices are then compared, first among themselves and then to best 2-density found so far. It should be noted that any entry can be moved to the last row without changing the values of the other entries through cyclic shifts of the rows[1] followed by cyclic shifts of the columns; therefore, it is sufficient to flip entries only in the last row of the matrix because the matrix is cyclic.

The computations were executed sequentially, on a node on `linux.cs.uwaterloo.ca`, in about 16 hours.

Table 3.3 compares the results of cyclic search to those of cyclic search with adjusting step. It can be seen from the table that the adjusting step improves the results in 15 out of 27 cases, and for the case $s = 8$ the algorithm performs as well as the exhaustive search. The rate of improvement is the most significant for $s = 3$, where the adjusting step improves the result by more than 130%. This rate decreases to less than 1% for $s = 28$, where flipping one entry increases the number of invertible 2×2 submatrices by 499.

[1] Rows and columns are shifted as a whole, not the entries in them.

Table 3.2: Performance of cyclic matrices as almost AONTs for $s = 2, \ldots, 36$ and $q = 2$.

s	1's / row	1 Frq	Max Cyc N_2	Max Cyc R_2	$N_2(s)$	$R_2(s)$
2	1	0.5	1	1	1	1
3	1	$0.\bar{3}$	3	$0.\bar{3}$	7	$0.\bar{7}$
4	3	0.75	30	$0.8\bar{3}$	30	$0.8\bar{3}$
5	3	0.6	65	0.65	70	0.7
6	5	$0.8\bar{3}$	135	0.6	150	$0.\bar{6}$
7	5	≈ 0.714	287	≈ 0.651	287	≈ 0.651
8	5	0.625	468	≈ 0.597	485	≈ 0.619
9	7	$0.\bar{7}$	765	≈ 0.590	783	≈ 0.604
10	7	0.7	1215	0.6	–	–
11	7	≈ 0.636	1716	≈ 0.567	–	–
12	9	0.75	2502	≈ 0.574	–	–
13	9	≈ 0.692	3510	≈ 0.577	–	–
14	9	≈ 0.643	4557	≈ 0.550	–	–
15	11	$0.7\bar{3}$	6210	≈ 0.563	–	–
16	11	≈ 0.688	8040	≈ 0.558	–	–
17	13	≈ 0.765	10030	≈ 0.542	–	–
18	13	$0.7\bar{2}$	12933	≈ 0.552	–	–
19	13	≈ 0.684	16017	≈ 0.548	–	–
20	15	0.75	19510	≈ 0.540	–	–
21	15	≈ 0.714	24045	≈ 0.545	–	–
22	15	≈ 0.681	28831	≈ 0.540	–	–
23	17	≈ 0.739	34385	≈ 0.537	–	–
24	17	≈ 0.708	41124	≈ 0.540	–	–
25	17	0.68	48100	≈ 0.534	–	–
26	19	≈ 0.731	56433	≈ 0.534	–	–
27	19	≈ 0.704	65934	≈ 0.535	–	–
28	19	≈ 0.679	75726	≈ 0.530	–	–
29	21	≈ 0.724	87638	≈ 0.532	–	–
30	21	0.7	100485	≈ 0.531	–	–
31	21	≈ 0.677	113863	≈ 0.527	–	–
32	23	≈ 0.719	130128	≈ 0.529	–	–
33	23	$0.6\bar{9}$	147213	≈ 0.528	–	–
34	25	≈ 0.735	165087	≈ 0.525	–	–
35	25	≈ 0.714	186445	≈ 0.527	–	–
36	25	≈ 0.694	208530	≈ 0.525	–	–

Table 3.3: Comparing the performance of cyclic matrices and almost cyclic matrices as almost AONTs for $s = 2, \ldots, 28$ and $q = 2$.

s	Cyc N_2	Cyc R_2	Adj Cyc N_2	Adj Cyc R_2
2	1	1	1	1
3	3	$0.\overline{3}$	7	$0.\overline{7}$
4	30	$0.8\overline{3}$	30	$0.8\overline{3}$
5	65	0.65	69	0.69
6	135	0.6	148	≈ 0.658
7	287	≈ 0.651	287	≈ 0.651
8	468	≈ 0.597	485	≈ 0.619
9	765	≈ 0.590	781	≈ 0.603
10	1215	0.6	1215	0.6
11	1716	≈ 0.567	1777	≈ 0.587
12	2502	≈ 0.574	2503	≈ 0.575
13	3510	≈ 0.577	3510	≈ 0.577
14	4557	≈ 0.550	4707	≈ 0.568
15	6210	≈ 0.563	6210	≈ 0.563
16	8040	≈ 0.558	8040	≈ 0.558
17	10030	≈ 0.542	10288	≈ 0.556
18	12933	≈ 0.552	12933	≈ 0.552
19	16017	≈ 0.548	16017	≈ 0.548
20	19510	≈ 0.540	19746	≈ 0.547
21	24045	≈ 0.545	24045	≈ 0.545
22	28831	≈ 0.540	28905	≈ 0.542
23	34385	≈ 0.537	34584	≈ 0.540
24	41124	≈ 0.540	41124	≈ 0.540
25	48100	≈ 0.534	48364	≈ 0.537
26	56433	≈ 0.534	56544	≈ 0.535
27	65934	≈ 0.535	65934	≈ 0.535
28	75726	≈ 0.530	76225	≈ 0.533

3.2.5.1 Random Constructions

We investigate the expected number of invertible 2 by 2 submatrices in a random s by s 0 - 1 matrix M. Suppose every entry of M is chosen to be a 1 with probability ϵ, independent of the values of all other entries. Using Fact 3.2.1, it is easy to see that a specified 2 by 2 submatrix is invertible with probability

$$4\epsilon^3(1-\epsilon) + 2\epsilon^2(1-\epsilon)^2 = 2\epsilon^2(1-\epsilon)(2\epsilon+1-\epsilon) = 2\epsilon^2(1-\epsilon^2).$$

This function is maximized by choosing $\epsilon = \sqrt{2}/2$. The expected number of invertible 2 by 2 submatrices in M is $\frac{1}{2}\binom{s}{2}^2$ (leading to an expected 2-density of 0.5). This method does not immediately yield an almost AONT because it seems difficult to ensure that the constructed matrix is itself invertible. However, the "adjusting step" method by Zhang et al. [53] (see Subsection 3.2.8 for more details) can be applied to flip some entries of M such that the resulting matrix is invertible. Even without using the adjusting step, random construction proves to be a useful method to achieve high R_2 values for small values of s.

3.2.6 Theoretical Constructions

Besides the random construction, we studied theoretical constructions, namely, a recursive method, SBIBDs, and cyclotomy, for invertible matrices with high R_2 values. In the following subsections we will discuss these constructions in detail.

3.2.6.1 Recursive Constructions

We start by investigating the recursive construction of almost AONTs. Specifically, we analyze a type of doubling construction in a particular case. We begin with the $(2, 4, 2)$-almost AONT from Example 3.2.4. Recall that this AONT arises from the matrix $J_4 - I_4$ and it achieves the optimal result $R_2(4) = 5/6$. We might try to use this matrix to construct a $(2, 8, 2)$-almost AONT. There are various ways in which we could try to do this; we present one method which leads to a reasonably good outcome. Consider the matrix

$$M = \left(\begin{array}{c|c} J_4 - I_4 & J_4 - I_4 \\ \hline J_4 - I_4 & J_4 \end{array}\right).$$

Table 3.4: 2 by 2 invertible submatrices of M

i, j	a_0, a_1, a_2, a_3	# invertible submatrices
$1 \leq i < j \leq 4$	$a_1 = 2, a_2 = 2, a_3 = 4$	20
$5 \leq i < j \leq 8$	$a_1 = 1, a_2 = 1, a_3 = 6$	13
$1 \leq i \leq 4, 5 \leq j \leq 8, j \neq i + 4$	$a_1 = 2, a_2 = 1, a_3 = 5$	17
$1 \leq i \leq 4, j = i + 4$	$a_0 = 1, a_1 = 1, a_3 = 6$	6

We first need to show that M is invertible. We show that $\det(M) = 1$ as follows. Consider a matrix of the form

$$M = \left(\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right),$$

where A, B, C, D are square submatrices and $CD = DC$. In this case, it is known that $\det(M) = \det(AD - BC)$.

In our construction, we have $CD = DC = 3J_4$, so this formula can be applied. We have $A = B = C = J_4 - I_4$ and it is easy to check that $BC = I_4$, $AD = J_4$. Therefore $AD - BC = J_4 - I_4$ and $\det(M) = \det(AD - BC) = \det(J_4 - I_4) = 1$.

Next, we proceed to compute the number of 2 by 2 invertible submatrices of M. We do this by looking at pairs of rows of M, say row i and row j, and computing the relevant numbers a_0, a_1, a_2, a_3 in each case (where we are using the notation from the proof of Lemma 3.2.8). We tabulate the results in Table 3.4.

The number of occurrences of the four cases enumerated in Table 3.4 is (respectively) $6, 6, 12$ and 4. Therefore,

$$N_2(M) = 6 \times 20 + 6 \times 13 + 12 \times 17 + 4 \times 6 = 426.$$

Finally, we compute

$$R_2(M) = \frac{426}{\binom{8}{2}^2} = \frac{426}{784} = 0.5434.$$

Summarizing, we have the following.

Theorem 3.2.13. *$N_{2,2}(8) \geq 426$ and $R_{2,2}(8) \geq 0.5434$.*

It is interesting to note that this recursive construction yields a better result than the direct constructions considered previously. For example, if $M = J_8 - I_8$, then we only get that $N_2 \geq 364$. Also, Theorem 3.2.7 (with $s = 8$, $t = 2$) only yields $N_2 \geq 322$.

3.2.6.2 Constructions from Symmetric BIBDs

We next give a construction that potentially achieves similar behavior as the random construction, using symmetric balanced incomplete block designs (SBIBDs). As we mentioned in Section 1.4, a (v, k, λ)-*balanced incomplete block design* (BIBD) is a pair $(X, \mathcal{B})$, where X is a set of v *points* and $\mathcal{B}$ is a collection of k-subsets of X called *blocks*, such that every pair of points occurs in exactly λ blocks. Denote $b = |\mathcal{B}|$. If $b = v$, the BIBD is called a symmetric BIBD (SBIBD).

Suppose $(X, \mathcal{B})$ is a (v, k, λ)-BIBD. Denote $X = \{x_i : 1 \leq i \leq v\}$ and $\mathcal{B} = \{B_j : 1 \leq j \leq b\}$. The *incidence matrix* of $(X, \mathcal{B})$ is the v by b 0 - 1 matrix $M = (m_{ij})$ where $m_{ij} = 1$ if $x_i \in B_j$, and $m_{ij} = 0$ if $x_i \notin B_j$.

Lemma 3.2.14. *Suppose M is the incidence matrix of a symmetric (v, k, λ)-BIBD. Then M is invertible over $\mathbb{F}_2$ if and only if k is odd and λ is even.*

Proof. It is well-known [13] that $\det(M)$ is an integer and

$$(\det(M))^2 = k^2(k - \lambda)^{v-1}.$$

Reducing modulo 2, we see that $\det(M) \equiv 1 \bmod 2$ if and only if k is odd and λ is even. □

Theorem 3.2.15. *Suppose M is the incidence matrix of a symmetric (v, k, λ)-BIBD where k is odd and λ is even. Then*

$$R_2(M) = \frac{k^2 - \lambda^2}{\binom{v}{2}}. \tag{3.4}$$

Proof. First, since k is odd and λ is even, M is invertible over $\mathbb{F}_2$ by Lemma 3.2.14. Consider two rows of M and define a_0, a_1, a_2, a_3 as in the proof of Theorem 3.2.9. Using the fact that M is the incidence matrix of a symmetric (v, k, λ)-BIBD, it is not hard to see that $a_0 = v - 2k + \lambda$, $a_1 = a_2 = k - \lambda$ and $a_3 = \lambda$. Then we can compute

$$a_1a_2 + a_1a_3 + a_2a_3 = 2\lambda(k - \lambda) + (k - \lambda)^2 = k^2 - \lambda^2.$$

From this, we have $N_2(M) = \binom{v}{2}(k^2 - \lambda^2)$ and (3.4) is easily derived. □

Now we try to figure out the best result that could possibly be obtained from Theorem 3.2.15. Suppose $k \approx cv$. Then from the equation $\lambda(v - 1) = k(k - 1)$, we see that $\lambda \approx c^2v$. Substituting into (3.4), we get $R_2(M) \approx 2(c^2 - c^4)$. Now we of course have $0 \leq c \leq 1$, and the function $2(c^2 - c^4)$ is maximized when $c = \sqrt{2}/2$. In this case, we would get

$R_2(M) \approx 1/2$, more-or-less matching the random construction from Section 3.2.5.1. We have also guaranteed that the matrix M is invertible. Of course, we would require a suitable SBIBD in order to get close to this bound.

We consider some examples to illustrate the application of Theorem 3.2.15.

Example 3.2.6. *It is known [13] that there is a* $(31, 21, 14)$*-SBIBD. Noting that* 21 *is odd and* 14 *is even, the incidence matrix of this design is invertible over* $\mathbb{F}_2$ *by Lemma 3.2.14. Observe that* $21/31$ *is quite close to* $\sqrt{2}/2$*, so we expect a good result. Applying Theorem 3.2.15, we get*

$$R_2(M) = \frac{21^2 - 14^2}{\binom{31}{2}} = \frac{49}{93} \approx 0.5269.$$

Example 3.2.7. *There also exists a* $(40, 27, 18)$*-SBIBD (see [13]). Noting that* 27 *is odd and* 18 *is even, the incidence matrix of this design is invertible over* $\mathbb{F}_2$ *by Lemma 3.2.14. Applying Theorem 3.2.15, we get*

$$R_2(M) = \frac{27^2 - 18^2}{\binom{40}{2}} = \frac{27}{52} \approx 0.5192.$$

Example 3.2.8. *A* $(4m-1, 2m-1, m-1)$*-SBIBD is called a* Hadamard design. *If* m *is odd, then* $\lambda = m - 1$ *is even. Certainly* $k = 2m - 1$ *is odd, so the incidence matrix* M *is invertible, by Lemma 3.2.14. These SBIBDs are known to exist for infinitely many (odd) values of* m*, e.g., whenever* $4m - 1 \equiv 3 \bmod 8$ *is a prime or a prime power (see [13]). From the incidence matrix of such a BIBD, we obtain*

$$R_2(M) = \frac{(2m-1)^2 - (m-1)^2}{\binom{4m-1}{2}} \approx \frac{3}{8}.$$

Example 3.2.9. *Here we make use of a classic result based on difference sets. Suppose* $q = 4t^2 + 9$ *is prime and* t *is odd. In this situation, it was shown by E. Lehmer that the quartic residues modulo* q*, together with* 0*, form a difference set which generates a* $(q, (q+3)/4, (q+3)/16)$*-SBIBD (e.g., see [13, p. 116]). If we complement this design (i.e., we replace all 0's by 1's and all 1's by 0's in the incidence matrix), the result is a* $(q, 3(q-1)/4, 3(3q-7)/16)$*-SBIBD. This SBIBD will have* k *odd and* λ *even, so its incidence matrix* M *is invertible, by Lemma 3.2.14. The first example is obtained when* $t = 5$*, yielding*

$$R_{2,2}(109) \geq \frac{329}{654}.$$

Asymptotically, from (3.4), we obtain

$$R_2(M) \approx \frac{63}{128}$$

if there exist sufficiently large primes q of the desired form. However, it is a famous unsolved conjecture that there exist infinitely many primes of the form $x^2 + 9$ [18], so we are not in a position to claim that this asymptotic result holds.

The following theorem generalizes Example 3.2.7.

Theorem 3.2.16. *Suppose m is a positive integer and $s = \frac{3^{m+1}-1}{2}$. Then*

$$R_{2,2}(s) \geq \frac{40 \times 3^{2m-3}}{(3^{m+1}-1)(3^m-1)}.$$

Proof. The points and hyperplanes of the m-dimensional projective geometry over $\mathbb{F}_3$ yield a $((3^{m+1}-1)/2, (3^m-1)/2, (3^{m-1}-1)/2)$-SBIBD. If we complement this design, we get a $((3^{m+1}-1)/2, 3^m, 2 \times 3^{m-1})$-SBIBD. This design has k odd and λ even, so we can apply Theorem 3.2.15. The result is that

$$R_{2,2}\left(\frac{3^{m+1}-1}{2}\right) \geq \frac{(3^m)^2 - (2 \times 3^{m-1})^2}{\binom{\frac{3^{m+1}-1}{2}}{2}} = \frac{40 \times 3^{2m-3}}{(3^{m+1}-1)(3^m-1)}.$$

□

Now we examine the asymptotic behavior of the result proven in Theorem 3.2.16. The SBIBD has $k \approx 2v/3$ and $\lambda \approx 4v/9$. It then follows from (3.4) that

$$R_2(M) = \frac{k^2 - \lambda^2}{\binom{v}{2}} \approx 2\left(\left(\frac{2}{3}\right)^2 - \left(\frac{4}{9}\right)^2\right) = \frac{40}{81}.$$

Therefore, we obtain the following corollary.

Corollary 3.2.17. *It holds that* $\limsup_{s\to\infty} R_{2,2}(s) \geq 40/81$.

We note that $40/81 \approx 0.494$. So there is a gap between our upper and lower asymptotic bounds on 2-density, which are respectively 0.625 (from Corollary 3.2.11) and 0.494 (and of course the lower bound only has been proven to hold within a certain infinite class of examples).

3.2.6.3 Constructions using Cyclotomy

We now look at constructions using cyclotomy. Let $p = 4f + 1$ be prime, where f is even, and let $\nu \in {\mathbb{F}_p}^*$ be a primitive element. Let $C_0 = \{\nu^{4i} : 0 \leq i \leq f - 1\}$; this is the unique subgroup of ${\mathbb{F}_p}^*$ having order f. The multiplicative cosets of C_0 are $C_j = \nu^j C_0$, for $j = 0, 1, 2, 3$. These cosets are often called *cyclotomic classes*.

We now construct a p by p 0 - 1 matrix $M' = (m_{ij})$ from C_0. The rows and columns of M' are indexed by $\mathbb{F}_p$, and $m_{ij} = 1$ if and only if $j - i \in C_0$. Note that the ith row of M' is the incidence vector of $i + C_0$. Finally, define M to be the complement of M' (i.e., replace all 1's by 0's and vice versa).

We will now compute the number of invertible 2 by 2 submatrices of M. Consider rows i_1 and i_2 of M. It is obvious that the number of invertible 2 by 2 submatrices contained in these two rows is the same as the number of invertible 2 by 2 submatrices contained in rows 0 and d, where $d = i_1 - i_2$. We can compute this number if we can determine the number n_d of columns c such that $m_{0c} = m_{dc} = 1$. It is clear that

$$n_d = |C_0 \cap (d + C_0)|.$$

However,

$$|C_0 \cap (d + C_0)| = |d^{-1}C_0 \cap (1 + d^{-1}C_0)|.$$

Now, $d^{-1}C_0 = C_j$, for some j, $0 \leq j \leq 3$, so

$$n_d = |C_j \cap (1 + C_j)|$$

for this particular value of j. This quantity is a *cyclotomic number of order* 4 and is denoted by (j, j).

We will make use of the following theorem from [22].

Theorem 3.2.18. *[22, Theorem 1] Suppose $p = 4f + 1$ is prime and f is even. Let $\nu \in \mathbb{F}_q$ be a primitive element. Let $p = \alpha^2 + \beta^2$, where $\alpha \equiv 1 \bmod 4$ and $\nu^f \equiv \alpha/\beta \bmod p$. Then*

the cyclotomic numbers (j,j) $(0 \le j \le 3)$*, are as follows:*

$$\begin{aligned}
(0,0) = A_0 &= \frac{p-11-6\alpha}{16} = \frac{4f-10-6\alpha}{16} \\
(1,1) = A_1 &= \frac{p-3+2\alpha-4\beta}{16} = \frac{4f-2+2\alpha-4\beta}{16} \\
(2,2) = A_2 &= \frac{p-3+2\alpha}{16} = \frac{4f-2+2\alpha}{16} \\
(3,3) = A_3 &= \frac{p-3+2\alpha+4\beta}{16} = \frac{4f-2+2\alpha+4\beta}{16}.
\end{aligned}$$

Remark 3.2.1. *A prime* $p \equiv 1 \bmod 4$ *can be expressed as the sum of two squares in a unique manner. If* $p = \alpha^2 + \beta^2$*, then one of* α, β *is odd and the other is even. So without loss of generality we can take* α *to be odd and* β *to be even. In this way,* α *and* β *are determined up to their signs. The condition* $\alpha \equiv 1 \bmod 4$ *now determines* α *uniquely, and, similarly,* $\nu^f \equiv \alpha/\beta \bmod p$ *determines* β *uniquely.*

Now we can compute the number of 2 by 2 submatrices contained in rows i_1 and i_2 of M. Again we define a_0, a_1, a_2, a_3 as in the proof of Theorem 3.2.9. Recalling that M is the complement of M', we have

$$a_1 = a_2 = f - (j,j)$$

and

$$a_3 = p - 2f + (j,j) = 2f + 1 + (j,j),$$

where $(i_1 - i_2)^{-1} C_0 = C_j$. Thus we obtain

$$a_1a_2 + a_1a_3 + a_2a_3 = 5f^2 + 2f - (j,j)(4f + 2 + (j,j)).$$

As we consider all $\binom{p}{2}$ pairs $\{i_1, i_2\}$ with $i_1 \neq i_2$, we see that (j,j) takes on each of the four possible values A_i $(1 \le i \le 4)$ one quarter of the time. Therefore, the total number of invertible 2 by 2 submatrices in M is

$$\begin{aligned}
&\binom{p}{2} \sum_{i=0}^{3} \left(\frac{5f^2 + 2f - A_i(4f + 2 + A_i)}{4} \right) \\
&= \binom{p}{2} \frac{252f^2 + 168f + 25 - 3\alpha^2 - 2\beta^2 - 6\alpha}{64},
\end{aligned}$$

where the last line is obtained by applying the formulas given in Theorem 3.2.18.

Table 3.5: Examples from Cyclotomy

p	f	α	β	$N_2(M)$	$R_2(M)$
17	3	1	4	9962	0.53860
97	5	9	−4	10831020	0.49962
193	5	−7	12	170314008	0.49613
241	7	−15	4	414228390	0.49527
401	3	1	−20	3177945050	0.49408
433	5	17	−12	4320175230	0.49388
449	3	−7	20	4995836216	0.49388

Example 3.2.10. *Suppose* $p = 17 = 4 \times 4 + 1$. *Then* $\nu = 3$ *is a primitive element. Since* $17 = 1^2 + 4^2$, *we have* $\alpha = 1$ *and* $\beta \in \{4, 13\}$. *We compute* $3^4 \equiv 13 \bmod 17$. *Since* $1/4 \equiv 13 \bmod 17$, *we have* $\beta = 4$. *The cyclotomic classes are*

$$\begin{aligned} C_0 &= \{1, 13, 16, 4\}, \\ C_1 &= \{3, 5, 14, 12\}, \\ C_2 &= \{9, 15, 8, 2\}, \\ C_3 &= \{10, 11, 7, 6\}. \end{aligned}$$

The cyclotomic numbers can now be computed from Theorem 3.2.18; they are

$$\begin{aligned} (0,0) = A_0 &= \frac{17 - 11 - 6}{16} = 0, \\ (1,1) = A_1 &= \frac{17 - 3 + 2 - 4 \times 4}{16} = 0, \\ (2,2) = A_2 &= \frac{17 - 3 + 2}{16} = 1, \\ (3,3) = A_3 &= \frac{17 - 3 + 2 + 4 \times 4}{16} = 2. \end{aligned}$$

The total number of invertible 2 *by* 2 *submatrices in* M *is* 9962.

It remains to consider the invertibility of the matrices M constructed above. The matrices in question are cyclic. Suppose a p by p cyclic 0 - 1 matrix M has as its initial

row the vector $(m_0, \ldots, m_{p-1})$. We associate with this vector the polynomial

$$m(x) = \sum_{i=0}^{p-1} m_i x^i \in \mathbb{Z}_2[x].$$

It is easy to see that M is invertible if and only if $\gcd(m(x), x^p - 1) = 1$. In this case, the inverse $m^{-1}(x)$ of $m(x)$ is defined in the quotient ring $\mathbb{Z}_2[x]/(x^p - 1)$. The cyclic matrix whose first row is determined by $m^{-1}(x)$ will in fact be the inverse of M. Therefore, to determine the invertibility of M, we just need to do a gcd computation.

Example 3.2.11. *Let $p = 17$. From Example 3.2.10, we have $C_0 = \{1, 13, 16, 4\}$. The first row of M is*

$$1\,0\,1\,1\,0\,1\,1\,1\,1\,1\,1\,1\,1\,0\,1\,1\,0,$$

which corresponds to the polynomial

$$m(x) = 1 + x^2 + x^3 + x^5 + x^6 + x^7 + x^8 + x^9 + x^{10} + x^{11} + x^{12} + x^{14} + x^{15}.$$

The inverse of $m(x) \mod x^{17} - 1$ is

$$m^{-1}(x) = 1 + x + x^3 + x^4 + x^5 + x^6 + x^7 + x^{10} + x^{11} + x^{12} + x^{13} + x^{14} + x^{16}.$$

By Dirichlet's Theorem, there are an infinite number of primes $p \equiv 1 \bmod 8$. However, we do not have a theoretical criterion to determine if a given matrix M in this class of examples is invertible. Therefore, we cannot prove that there are an infinite number of examples of this type. However, by computing gcds, as described above, we determined all the invertible matrices M of order less than 500 that can be constructed by this method. Some data about these matrices is presented in Table 3.5. Another observation is that, if this is in fact an infinite class, then it can be shown that the density of these examples approaches $63/128 \approx 0.492$ as f approaches infinity.

3.2.7 Values and Bounds on $N_{2,2}(s)$ for Small s

We summarize our upper and lower bounds on $N_{2,2}(s)$ for $s \leq 12$ in Table 3.6. For the cases $s = 5, 6, 7, 8$, we have exact values of $N_{2,2}(s)$ that are obtained from exhaustive computer searches. For $s = 9$, our lower bound is obtained from a partial (uncompleted) exhaustive search. For $s = 10, 11, 12$, the lower bounds come from randomly constructed matrices. All these matrices are published in a technical report [30].

Table 3.6: Values and Bounds on $N_{2,2}(s)$ for small s

s	$N_{2,2}(s)$	justification
2	$N_{2,2}(2) = 1$	Fact 3.2.1
3	$N_{2,2}(3) = 7$	Example 3.2.3
4	$N_{2,2}(4) = 30$	Example 3.2.4
5	$N_{2,2}(5) = 70$	exhaustive search ([14, Example 36])
6	$N_{2,2}(6) = 150$	exhaustive search ([14, Example 37])
7	$N_{2,2}(7) = 287$	exhaustive search ([14, Example 38])
8	$N_{2,2}(8) = 485$	exhaustive search ([14, Example 39])
9	$N_{2,2}(9) \geq 783$	[14, Example 40]
10	$N_{2,2}(10) \geq 1194$	[14, Example 41]
11	$N_{2,2}(11) \geq 1744$	[14, Example 42]
12	$N_{2,2}(12) \geq 2448$	[14, Example 43]

3.2.8 Updated Results

Based on and subsequent to our results [14], Zhang et al. [53] proved the existence of almost 2-AONTs with maximum $R_{2,2}$ values or "invertible binary matrices with maximum number of 2-by-2 invertible submatrices" [53], in their own words. The authors introduced a different quadratic programming model and using that model they proved that 0.5 is an asymptotic upper bound for the ratio of the number of invertible 2 by 2 submatrices to the total number of 2 by 2 submatrices in an invertible binary matrix. Then, they [53] extended our random construction (which we presented in Section 3.2.5.1) to prove the lower bound. Recall that the difficulty is to prove existence of a random matrix that is invertible. This problem can be solved by means of an "adjusting step" that alters entries on the diagonal of the matrix.

The upper and lower bounds can be combined to yield Theorem 3.2.19.

Theorem 3.2.19. $\lim_{s\to\infty} R_{2,2}(s) = 0.5$.

Zhang et al. [53] then presented a method of constructing invertible binary matrices with high $R_{2,2}$ values. Their construction is comprised of two steps: "main step" and "adjusting step". The main step is very similar to our construction using cyclotomy (see Section 3.2.6.3 for details) except that they use cyclotomic classes of order 7 instead of order 4. The result is a cyclic binary matrix with high $R_{2,2}$ values, namely, asymptotically equal to 1200/2401. Then, the adjusting step assures the invertibility of the matrix by

flipping some entries on the main diagonal of the matrix. Therefore, the matrix may not be cyclic at the end, but it is invertible.

3.3 Linear Transforms over $\mathbb{F}_3$

In this section, we extend the alphabet and consider invertible matrices with entries from $\mathbb{F}_3$, i.e., elements of $\{0, 1, 2\}$. For the case of $t = 1$, the following construction by Stinson [42] works for any prime power $q \geq 2$.

Construction 3.3.1. *Let x be a non-zero element in $\mathbb{F}_q$ such that $x \neq -1$. We define M to be as follows:*

$$M = \begin{pmatrix} x & x & x & \cdots & x \\ x & x+1 & x & \cdots & x \\ x & x & x+1 & \cdots & x \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x & x & x & \cdots & x+1 \end{pmatrix}.$$

To prove the invertibility of M, we subtract the first row from all the other rows. Using the cofactor expansion of the new matrix with respect to its first column, we can see that the determinant of M is equal to $x \neq 0$.

Since all elements of M are non-zero, M is a $(1, s, q)$-AONT. Hence, we now mostly focus on the case of $t = 2$. We use various computational methods of finding invertible matrices over $\mathbb{F}_3$ with high 2-density.

3.3.1 Random Construction

First, we will consider the random construction of such matrices. In this case, there are $(3^2 - 1)(3^2 - 3) = 48$ invertible 2×2 matrices, as listed below.

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}, \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix},$$

$$\begin{pmatrix} 0 & 2 \\ 2 & 0 \end{pmatrix}, \begin{pmatrix} 2 & 2 \\ 2 & 0 \end{pmatrix}, \begin{pmatrix} 2 & 2 \\ 0 & 2 \end{pmatrix}, \begin{pmatrix} 2 & 0 \\ 2 & 2 \end{pmatrix}, \begin{pmatrix} 0 & 2 \\ 2 & 2 \end{pmatrix}, \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}, \begin{pmatrix} 1 & 1 \\ 2 & 1 \end{pmatrix},$$

$$\begin{pmatrix} 1 & 2 \\ 1 & 1 \end{pmatrix}, \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}, \begin{pmatrix} 2 & 2 \\ 2 & 1 \end{pmatrix}, \begin{pmatrix} 2 & 2 \\ 1 & 2 \end{pmatrix}, \begin{pmatrix} 2 & 1 \\ 2 & 2 \end{pmatrix}, \begin{pmatrix} 1 & 2 \\ 2 & 2 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix},$$

$$\left(\begin{array}{cc} 0 & 1 \\ 2 & 0 \end{array}\right), \left(\begin{array}{cc} 0 & 2 \\ 1 & 0 \end{array}\right), \left(\begin{array}{cc} 2 & 0 \\ 0 & 1 \end{array}\right), \left(\begin{array}{cc} 0 & 1 \\ 1 & 2 \end{array}\right), \left(\begin{array}{cc} 0 & 1 \\ 2 & 1 \end{array}\right), \left(\begin{array}{cc} 0 & 2 \\ 1 & 1 \end{array}\right), \left(\begin{array}{cc} 1 & 1 \\ 0 & 2 \end{array}\right),$$
$$\left(\begin{array}{cc} 1 & 1 \\ 2 & 0 \end{array}\right), \left(\begin{array}{cc} 1 & 0 \\ 1 & 2 \end{array}\right), \left(\begin{array}{cc} 1 & 0 \\ 2 & 1 \end{array}\right), \left(\begin{array}{cc} 1 & 2 \\ 0 & 1 \end{array}\right), \left(\begin{array}{cc} 1 & 2 \\ 1 & 0 \end{array}\right), \left(\begin{array}{cc} 2 & 0 \\ 1 & 1 \end{array}\right), \left(\begin{array}{cc} 2 & 1 \\ 0 & 1 \end{array}\right),$$
$$\left(\begin{array}{cc} 2 & 1 \\ 1 & 0 \end{array}\right), \left(\begin{array}{cc} 0 & 2 \\ 2 & 1 \end{array}\right), \left(\begin{array}{cc} 0 & 2 \\ 1 & 2 \end{array}\right), \left(\begin{array}{cc} 0 & 1 \\ 2 & 2 \end{array}\right), \left(\begin{array}{cc} 2 & 2 \\ 0 & 1 \end{array}\right), \left(\begin{array}{cc} 2 & 2 \\ 1 & 0 \end{array}\right), \left(\begin{array}{cc} 2 & 0 \\ 2 & 1 \end{array}\right),$$
$$\left(\begin{array}{cc} 2 & 0 \\ 1 & 2 \end{array}\right), \left(\begin{array}{cc} 2 & 1 \\ 0 & 2 \end{array}\right), \left(\begin{array}{cc} 2 & 1 \\ 2 & 0 \end{array}\right), \left(\begin{array}{cc} 1 & 0 \\ 2 & 2 \end{array}\right), \left(\begin{array}{cc} 1 & 2 \\ 0 & 2 \end{array}\right), \left(\begin{array}{cc} 1 & 2 \\ 2 & 0 \end{array}\right)$$

A random 2×2 matrix $A = \{a_{i,j}\}$ that is generated by randomly assigning its entries to 0, 1, and 2, with the following probabilities

$$Pr(a_{i,j} = 1) = \alpha,\ Pr(a_{i,j} = 2) = \beta,\ Pr(a_{i,j} = 0) = \gamma,\ \alpha + \beta + \gamma = 1$$

is therefore invertible with probability

$$\begin{aligned} f(\alpha, \beta, \gamma) &= 2(\alpha^2\gamma^2 + \beta^2\gamma^2) + 4(\alpha^3\gamma + \beta^3\gamma) \\ &\quad + 4(\alpha^3\beta + \beta^3\alpha) + 12(\alpha^2\beta\gamma + \beta^2\alpha\gamma) + 4\alpha\beta\gamma^2. \end{aligned} \tag{3.5}$$

To maximize f we use the Lagrange multiplier method. First, define g as follows:

$$\begin{aligned} g(\alpha, \beta, \gamma, \lambda) &= f(\alpha, \beta, \gamma) - \lambda(1 - \alpha - \beta - \gamma) \\ &= 2(\alpha^2\gamma^2 + \beta^2\gamma^2) + 4(\alpha^3\gamma + \beta^3\gamma) \\ &\quad + 4(\alpha^3\beta + \beta^3\alpha) + 12(\alpha^2\beta\gamma + \beta^2\alpha\gamma) \\ &\quad + 4\alpha\beta\gamma^2 - \lambda(1 - \alpha - \beta - \gamma). \end{aligned} \tag{3.6}$$

Computing all four partial derivatives of g, we get

$$\frac{\partial g}{\partial \alpha} = 4\alpha\gamma^2 + 12\alpha^2\gamma + 12\alpha^2\beta + 4\beta^3 + 24\alpha\beta\gamma + 12\beta^2\gamma + 4\beta\gamma^2 - \lambda \tag{3.7}$$

$$\frac{\partial g}{\partial \beta} = 4\beta\gamma^2 + 12\beta^\gamma + 12\beta^2\alpha + 4\alpha^3 + 24\alpha\beta\gamma + 12\alpha^2\gamma + 4\alpha\gamma^2 - \lambda \tag{3.8}$$

$$\frac{\partial g}{\partial \gamma} = 4\alpha^2\gamma + 4\beta^2\gamma + 4\alpha^3 + 4\beta^3 + 12\alpha^2\beta + 12\beta^2\alpha + 8\alpha\beta\gamma - \lambda \tag{3.9}$$

$$\frac{\partial g}{\partial \lambda} = -\alpha - \beta - \gamma + 1 = 0. \tag{3.10}$$

Table 3.7: Highest $N_2(M)$ and $R_2(M)$ found for random matrices $M_{s\times s}$ over $\mathbb{F}_3$ where $s \in \{3, \ldots, 13\}$.

s	N_2	R_2	0 Frq	1 Frq	2 Frq
3	9	1	$0.\bar{2}$	$0.\bar{4}$	$0.\bar{2}$
4	34	$0.9\bar{4}$	0.25	0.25	0.5
5	86	0.86	0.20	0.48	0.32
6	185	$0.8\bar{2}$	$0.1\bar{6}$	$0.\bar{4}$	$0.3\bar{8}$
7	343	$0.\bar{7}$	≈ 0.204	≈ 0.388	≈ 0.408
8	591	≈ 0.754	≈ 0.156	≈ 0.391	≈ 0.453
9	965	≈ 0.745	≈ 0.173	≈ 0.395	≈ 0.432
10	1479	≈ 0.730	0.18	0.43	0.39
11	2189	≈ 0.724	≈ 0.190	≈ 0.397	≈ 0.413
12	3090	≈ 0.709	≈ 0.188	≈ 0.382	≈ 0.431
13	4306	≈ 0.708	≈ 0.172	≈ 0.402	≈ 0.426

We used Maple to solve $\frac{\partial g}{\partial \alpha} = 0, \frac{\partial g}{\partial \beta} = 0, \frac{\partial g}{\partial \gamma} = 0, \frac{\partial g}{\partial \lambda} = 0$, and the only solution that maximizes g, and therefore f, and which satisfies $\alpha, \beta, \gamma \in [0, 1]$ is $\alpha = \beta = \sqrt{6}/6$, $\gamma = 1 - \sqrt{6}/3, \lambda = 8/3$, from which we obtain $g(\sqrt{6}/6, \sqrt{6}/6, 1 - \sqrt{6}/3, 8/3) = 2/3$.

For each value of s, $3 \leq s \leq 13$, 10000 random matrices with elements chosen randomly from aforementioned distribution were generated. For each value of s, the number of invertible 2×2 submatrices was counted for any of the resulting invertible random matrices. The invertible matrices, found by the random search, with the maximum number of invertible 2×2 submatrices are reported in our work with Stinson [31]. Table 3.7 shows the largest number of invertible 2×2 submatrices found, in the invertible random matrices.

3.3.2 Exhaustive Search and Search for Cyclic and Almost Cyclic Matrices

Similar to the $q = 2$ case, the exhaustive search algorithms, as well as the algorithms for finding cyclic and almost cyclic matrices, were used to generate the ternary invertible matrices with the maximum number of invertible 2×2 submatrices. The algorithms are the same as those described in the previous section, only modified to fit the $q = 3$ case.

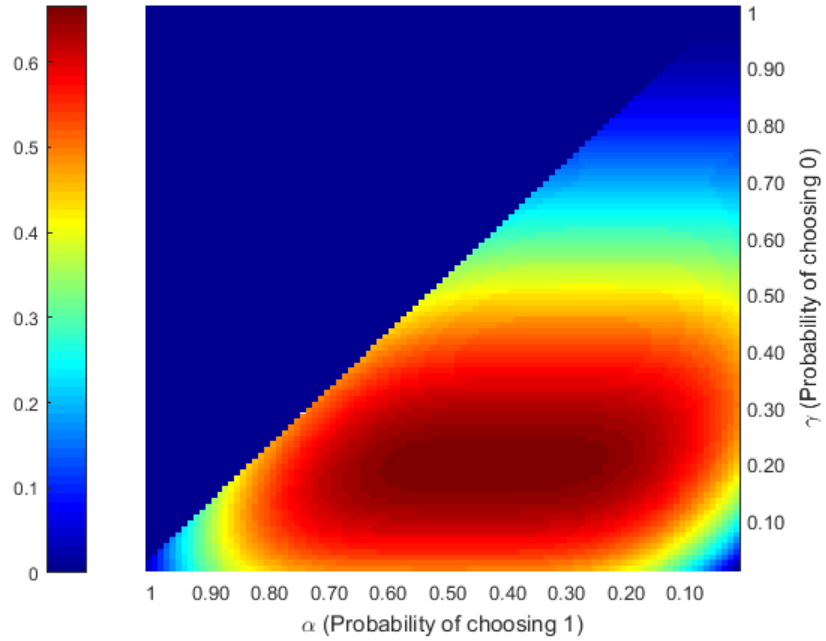


Figure 3.2: The value of function f for different values of α and γ

Specifically, the linear independence of the rows of the matrix cannot be checked by means of boolean functions any more. In the exhaustive search, eliminating search branches based on the permutations of columns is limited. In the search for cyclic and almost cyclic matrices on the other hand, in addition to cyclic shifts of each row, multiples of all its cyclic shifts, including the row itself, were also omitted from the search.

The algorithms were executed on a node on `linux.cs.uwaterloo.ca`, for $3 \leq s \leq 5$ for the exhaustive search and $3 \leq s \leq 20$ for the other two. The exhaustive search for $s = 6$ was computed on 160 virtual cores of a (`Tick`) node on RIPPLE server. Tables 3.8 and 3.9 present the results.

Table 3.8 shows, that in most cases, the frequencies of 0's, 1's, and 2's of the cyclic matrices found by the search do not exactly follow the results of the probabilistic analysis. To explain this difference, it is required to consider the behavior of function f. Figure 3.2 demonstrates that the value of function f is mostly sensitive to the changes in the value of γ, rather than changes in the values of α or β. This explains why the frequency of 0's in the cyclic matrices is around 20%, yet the frequencies of 1's and 2's vary considerably.

Comparing the results provided in Table 3.9 to those provided in Table 3.3, it can be

seen that applying the adjusting step improves the results of cyclic search less often and by smaller values for the $q = 3$ case as it does for the $q = 2$ case. While changing one entry of the binary cyclic matrices can improves the value of N_2 in 10 cases out of first 20 cases and by up to 236, this technique only changes N_2 in only 3 cases out of the first 20 cases in the ternary cyclic matrices. That the ternary matrices have more flexibility than the binary ones and the maximum value of N_2 is naturally higher for them can be explanations for this difference in the performance of this technique on the two different sets of matrices.

The next observation concerns the behavior of R_2 for the cyclic matrices and $R_2(s)$. In the binary case, the results of the exhaustive search algorithm show that the $R_2(s)$ decreases to about 0.6 as soon as s reaches 9. However, the maximum cyclic R_2 values, that were closely following $R_2(s)$, approach 0.52, which is consistent with the result by Zhang et al. [53], that the upper limit of $R_2(s)$ converges to 0.5, as we increase s. This result is also consistent with the expected value of R_2 when we set the frequency of 1 to be $\sqrt{2}/2$ [14]. Although there is not enough data for the ternary case, the results do not rule out the possibility of R_2 converging to $2/3$ when $q = 3$.

Finally, we will discuss the effect of the adjusting step. As mentioned in Section 2, the relative impact of one adjusting step is reduced as s grows. The effect of multiple adjusting steps is open to be studied. Also, the patterns of cyclic matrices that cannot be improved by one adjusting step, i.e., cyclic matrices with the highest R_2 among all the matrices in the neighborhood of being 1 entry away from that matrix, can be further investigated.

Table 3.8: Highest $N_2(M)$ and $R_2(M)$ found for cyclic matrices $M_{s\times s}$ over $\mathbb{F}_3$ where $s \in \{3,\ldots,20\}$ versus $N_{2,3}(s)$ and $R_{2,3}(s)$ for $s = 3,\ldots,6$.

s	Cyc N_2	Cyc R_2	0 Frq	1 Frq	2 Frq	$N_2(s)$	$R_2(s)$	1 Frq	2 Frq
3	9	1	$0.\bar{3}$	$0.\bar{6}$	0	9	1	$0.\bar{6}$	0
4	34	$0.9\bar{4}$	0.25	0.5	0.25	34	$0.9\bar{4}$	0.625	0.125
5	90	0.9	0.2	0.6	0.2	90	0.9	0.56	0.24
6	189	0.84	$0.1\bar{6}$	0.5	$0.\bar{3}$	195	$0.8\bar{6}$	$0.52\bar{7}$	$0.2\bar{7}$
7	357	≈ 0.810	≈ 0.143	≈ 0.571	≈ 0.286	–	–	–	–
8	600	≈ 0.765	0.25	0.5	0.25	–	–	–	–
9	1008	$0.\bar{7}$	$0.\bar{2}$	$0.\bar{3}$	$0.\bar{4}$	–	–	–	–
10	1550	≈ 0.765	0.2	0.3	0.5	–	–	–	–
11	2288	≈ 0.756	$0.1\bar{8}$	$0.4\bar{5}$	$0.3\bar{6}$	–	–	–	–
12	3264	≈ 0.749	$0.1\bar{6}$	$0.58\bar{3}$	0.25	–	–	–	–
13	4498	≈ 0.739	≈ 0.154	≈ 0.466	≈ 0.385	–	–	–	–
14	6069	≈ 0.733	≈ 0.214	≈ 0.357	$0\approx .429$	–	–	–	–
15	8085	$0.7\bar{3}$	0.2	$0.5\bar{3}$	$0.2\bar{6}$	–	–	–	–
16	10456	≈ 0.726	≈ 0.188	≈ 0.566	0.25	–	–	–	–
17	13413	≈ 0.725	0.177	≈ 0.471	≈ 0.353	–	–	–	–
18	16839	≈ 0.719	$0.1\bar{6}$	$0.3\bar{8}$	$0.\bar{4}$	–	–	–	–
19	20938	≈ 0.716	≈ 0.211	0.421	0.368	–	–	–	–
20	25840	≈ 0.716	0.2	0.5	0.3	–	–	–	–

Table 3.9: Comparing the performance of cyclic matrices and almost cyclic matrices as almost AONTs for $s = 2, \ldots, 20$ and $q = 3$.

s	Cyc N_2	Cyc R_2	Adj Cyc N_2	Adj Cyc R_2
3	9	1	9	1.0
4	34	$0.9\bar{4}$	34	$0.9\bar{4}$
5	90	0.9	90	0.9
6	189	0.84	189	0.84
7	357	≈ 0.810	357	≈ 0.810
8	600	≈ 0.765	608	≈ 0.776
9	1008	$0.\bar{7}$	1008	$0.\bar{7}$
10	1550	≈ 0.765	1550	≈ 0.765
11	2288	≈ 0.756	2288	≈ 0.756
12	3264	≈ 0.749	3264	≈ 0.749
13	4498	≈ 0.739	4498	≈ 0.739
14	6069	≈ 0.733	6080	≈ 0.734
15	8085	$0.7\bar{3}$	8085	$0.7\bar{3}$
16	10456	≈ 0.726	10482	≈ 0.728
17	13413	≈ 0.725	13413	≈ 0.725
18	16839	≈ 0.719	16839	≈ 0.719
19	20938	≈ 0.716	20938	≈ 0.716
20	25840	≈ 0.716	25840	≈ 0.716

Chapter 4

More Generalizations of All-or-Nothing Transforms

In Chapter 2, we discussed t-AONTs as a generalization of AONTs, followed by a study of structures close to 2-AONTs in Chapter 3. In this chapter, we introduce four additional generalizations of t-AONTs. Similar to the approach taken in Chapter 2, both weak and perfect security can be considered for all the generalizations in this chapter; however, we only provide the definitions for perfect security. Also, we consider the unbiased array representation of each of these generalizations. As was the case in Chapter 2, if all the input s-tuples are equiprobable, the unbiased array representations provide perfect security. Further, weak security is achieved for any probability distribution of the inputs (provided each input occurs with non-zero probability).

4.1 Range AONTs

As observed through different examples of AONTs, the existence of a (t, s, q)-AONT implies that of a $(t, s-1, q)$-AONT as shown in Theorem 2.4.2, as well as an $(s-t, s, q)$-AONT as mentioned by Wang et al. [50]. In Chapter 2, we showed that a t-AONT guarantees the perfect security of any t input blocks in the absence of any t output blocks; however, as soon as the adversary learns about more than $s-t$ elements, no security can be guaranteed, unless the AONT is also a $(t-1)$-AONT. If the t-AONT structure is a $(t-1)$-AONT as well, the adversary's knowledge of one additional output leaves any $t-1$ input elements undetermined. In order to prevent total loss of guaranteed security upon the knowledge of "one more" output element, we define *range AONTs*.

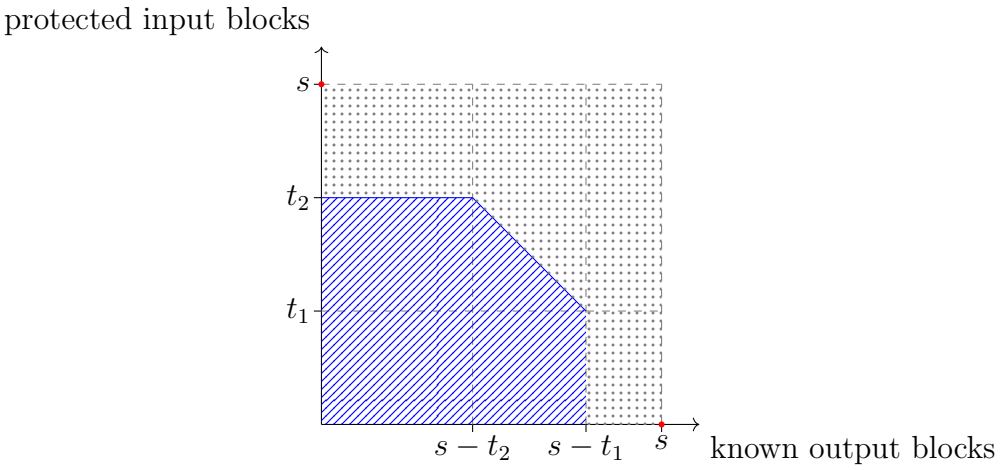


Figure 4.1: The behavior of a $([t_1, t_2], s, v)$-rangeAONT for different numbers of available output blocks.

Definition 4.1.1. *Suppose s, t_1, and t_2 are positive integers, where $t_1 \leq t_2 \leq s$. Let*

$$\mathbf{X_1}, \ldots, \mathbf{X_s}, \mathbf{Y_1}, \ldots, \mathbf{Y_s}$$

*be random variables taking on values in a finite set Σ of size v. These $2s$ random variables define a $([t_1, t_2], s, v)$-*rangeAONT *provided that the following conditions are satisfied:*

1. $\mathsf{H}(\mathbf{Y_1}, \ldots, \mathbf{Y_s} \mid \mathbf{X_1}, \ldots, \mathbf{X_s}) = 0$.
2. $\mathsf{H}(\mathbf{X_1}, \ldots, \mathbf{X_s} \mid \mathbf{Y_1}, \ldots, \mathbf{Y_s}) = 0$.
3. *For all $\mathcal{X} \subseteq \{\mathbf{X_1}, \ldots, \mathbf{X_s}\}$ with $|\mathcal{X}| = t$ such that $t_1 \leq t \leq t_2$, and for all $\mathcal{Y} \subseteq \{\mathbf{Y_1}, \ldots, \mathbf{Y_s}\}$ with $|\mathcal{Y}| = s - t$, it holds that*

$$\mathsf{H}(\mathcal{X} \mid \{\mathbf{Y_1}, \ldots, \mathbf{Y_s}\} \setminus \mathcal{Y}) = \mathsf{H}(\mathcal{X}). \tag{4.1}$$

Accordingly, a (t, s, v)-AONT is a $([t_1, t_2], s, v)$-rangeAONT when $t_1 = t_2 = t$.

Figure 4.1 depicts the behavior of a $([t_1, t_2, s, v])$-AONT. The area hatched in blue presents the number of protected input blocks are protected upon the availability of that many output blocks to the adversary.

If we consider the array representation of the a $([t_1, t_2], s, v)$-rangeAONT, similar to the (t, s, v)-AONT representation in Figure 2.1, for any integer t, $t_1 \leq t \leq t_2$, fixing any $s - t$ coordinates of an output, does not yield any information about any t input coordinates.

The following is a straightforward generalization of Theorem 2.2.1.

Theorem 4.1.1. *[14] A $([t_1, t_2], s, v)$-rangeAONT is equivalent to a $(v^s, 2s, v)$ array that is unbiased with respect to the following sets of columns:*

1. $\{1, \ldots, s\}$,
2. $\{s + 1, \ldots, 2s\}$,
3. $I \cup J$, *for any sets* I *and* J *where* $I \subseteq \{1, \ldots, s\}$, $|I| = t$ *and* $t_1 \leq t \leq t_2$, $J \subseteq \{s + 1, \ldots, 2s\}$, *and* $|J| = s - t$.

Note that, when $t_1 = t_2$ in Theorem 4.1.1, we obtain Theorem 2.2.1. The following result is an immediate generalization of Corollary 2.2.4.

Theorem 4.1.2. *Suppose there exists an $OA(s, 2s, v)$. Then there exists a $([t_1, t_2], s, v)$-rangeAONT for all t_1 and t_2 such that $1 \leq t_1 \leq t_2 \leq s$.*

In particular, if the range AONT maintains providing the guaranteed security as long as the adversary does not have access to at least one output element, we call it a *strong AONT*.

Definition 4.1.2. *A $([t_1, t_2], s, v)$-rangeAONT is a* (t, s, v)-strong AONT *if $t_1 = 1$ and $t_2 = t$.*

The following theorem is the direct result from combining Theorem 4.1.2 and Definition 4.1.2.

Corollary 4.1.3. *Suppose there exists an $OA(s, 2s, v)$. Then there exists a (t, s, v)-strong AONT for all t, $1 \leq t \leq s$.*

Therefore, for a (t, s, q)-strong AONT, if the adversary is missing i output elements, $1 < i \leq t$, obtaining one extra output element only makes it possible to compute functions of i input elements.

Similar to the case of AONTs and t-AONTs, we can define *linear range AONTs*, as a range AONT such that each output element is a linear function of the input elements. We write a linear range AONT in the form of $\mathbf{y} = \mathbf{x}M^{-1}$ and its inverse as $\mathbf{x} = \mathbf{y}M$, where M is an $s \times s$ matrix. Using Lemma 2.3.1 for all values of $t_1 \leq t \leq t_2$ results in the following corollary.

Corollary 4.1.4. *Suppose that q is a prime power and M is an invertible s by s matrix with entries from $\mathbb{F}_q$. M is a $([t_1, t_2], s, q)$-rangeAONT if and only if, for any value of t, where $t_1 \leq t \leq t_2$, all t by t submatrices of M are invertible.*

As mentioned in Chapter 2, all $t \times t$ submatrices of an $s \times s$ Cauchy matrix are invertible, for $t \in \{1, 2, \ldots, s\}$. Hence, any $s \times s$ Cauchy matrix over $\mathbb{F}_q$ is a linear $([t_1, t_2], s, q)$-rangeAONT, for all $1 \leq t_1 \leq t_2 \leq s$, and also a (t, s, q)-strong AONT for $1 \leq t \leq s$. However, it is known that, in a Cauchy matrix, $q \geq 2s$.

For fixed positive integers t_1, t_2, and any prime power q, define

$$\mathcal{S}_R(t_1, t_2, q) = \{s : \text{there exists a linear } ([t_1, t_2], s, q)\text{-rangeAONT}\}.$$

Cauchy matrices are evidence that $\lfloor \frac{q}{2} \rfloor \in \mathcal{S}_R(t_1, t_2, q)$, so $\mathcal{S}_R(t_1, t_2, q)$ is not empty. Also, from Theorem 2.5.3, Remark 2.5.1 and Theorem 2.4.2, there exists a maximum element in $\mathcal{S}_R([t_1, t_2], q)$, which we will denote by $M_R([t_1, t_2], q)$.

Since we know the behavior of the relation between s and q at the extremes of options for t_1 and t_2, i.e., $t_1 = t_2 = 1$, and $t_1 = 1, t_2 = s$, we studied intermediate values of t_1 and t_2 to achieve a better understanding of this relation in between the extreme cases, as allowed by our computational resources. To this end, we conducted a computer search for $([1, 2], s, q)$-, $([1, 3], s, q)$-, and $([2, 3], s, q)$-rangeAONTs. The results of our computer search are presented in the following examples. It can be observed that whenever $t_1 = 1$, the matrices do not have any zero entries because any 1 by 1 submatrix needs to be invertible.

Example 4.1.1. *A linear $([1, 2], 3, 4)$-rangeAONT:*

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix}$$

Example 4.1.2. *A linear $([1, 2], 5, 7)$-rangeAONT:*

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 3 & 4 & 5 & 6 \\ 1 & 4 & 5 & 6 & 2 \\ 1 & 5 & 6 & 2 & 4 \end{pmatrix}$$

Example 4.1.3. *A linear* $([1,2],7,8)$*-rangeAONT:*

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 3 & 4 & 5 & 6 & 7 & 2 \\ 1 & 4 & 5 & 6 & 7 & 2 & 3 \\ 1 & 5 & 6 & 7 & 2 & 3 & 4 \\ 1 & 6 & 7 & 2 & 3 & 4 & 5 \\ 1 & 7 & 2 & 3 & 4 & 5 & 6 \end{pmatrix}$$

Example 4.1.4. *A linear* $([1,3],3,4)$*-rangeAONT:*

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix}$$

Example 4.1.5. *A linear* $([1,3],4,7)$*-rangeAONT:*

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 1 & 5 & 6 & 2 \\ 1 & 6 & 5 & 3 \end{pmatrix}$$

Example 4.1.6. *A linear* $([1,3],5,9)$*-rangeAONT:*

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 3 & 8 & 5 & 7 \\ 1 & 4 & 5 & 8 & 6 \\ 1 & 8 & 6 & 7 & 2 \end{pmatrix}$$

Remark 4.1.1. *Any linear* $(s-1,s,q)$*-AONT is an* $([s-1,s],s,q)$*-rangeAONT.*

Example 4.1.7. *A linear* $([2,3],3,3)$*-rangeAONT, using a* $(2,3,3)$*-AONT construction:*

$$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

Example 4.1.8. *A linear* $([2,3],5,7)$*-rangeAONT:*

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 2 & 4 \\ 1 & 1 & 0 & 4 & 2 \\ 1 & 2 & 4 & 0 & 1 \\ 1 & 4 & 2 & 1 & 0 \end{pmatrix}$$

Example 4.1.9. *A linear* $([2,3],6,11)$*-rangeAONT:*

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 2 & 3 & 4 \\ 1 & 1 & 4 & 9 & 0 & 10 \\ 1 & 2 & 0 & 8 & 6 & 1 \\ 1 & 6 & 3 & 5 & 4 & 0 \\ 1 & 8 & 2 & 0 & 5 & 9 \end{pmatrix}$$

As we observed in Section 2.5, other than $q = 4$, whenever q is a prime power but not a prime, the exhaustive search did not find any $(2, q, q)$-AONTs of type q, for $q = 8, 9, 16$. From Remark 2.6.1, we know that type q is the only type possible. However, for range AONTs, there are different cases, e.g., $([1,2],7,8)$-rangeAONT from Example 4.1.3 and $([1,3],5,9)$-rangeAONT from Example 4.1.6, where prime power values of q are the smallest alphabet size where a range AONT exists for fixed values of t_1, t_2, and s.

4.2 Asymmetric AONTs

Let I_s and J_s be the identity matrix of size s and an $s \times s$ square matrix of all ones, respectively. Stinson [42] studied the AONT properties of $J_s - I_s$ as a linear $(1, s, 2)$-AONT, and its properties as an almost $(2, s, 2)$-AONT were further discussed by D'Arco et al. [14] and mentioned in Section 3.2. However, Karame et al. [21] took a slightly different approach to utilize this construction in distributed storage, and they named their scheme *bastion AONT*. In this section, we will discuss this new perspective on AONT and work on a generalization of AONTs based on it.

Instead of considering the security of 1 (or 2) input elements in the absence of 1 (or 2) output elements, Karame et al. [21] consider the security of a single input element in the absence of any pair of output elements. In order to achieve this goal for an even number

of elements, s, they use bastion AONT, which is a linear mapping over $\mathbb{F}_{2^\ell}$ represented by matrix $M = M^{-1} = J_s - I_s$. In the analysis of their scheme, Karame et al. [21] use the results by Stinson [42]. Stinson [42] observed that computing

$$\mathbf{y} = \mathbf{x}(J_s - I_s)$$

and

$$\mathbf{x} = \mathbf{y}(J_s - I_s)$$

is equivalent to calculating the following exclusive-OR (XOR) operations:

$$r = \bigoplus_{i=1}^{s} x_i, \text{ and } y_i = r \oplus x_i,$$

and

$$r' = \bigoplus_{i=1}^{s} y_i, \text{ and } x_i = r' \oplus y_i,$$

respectively. Stinson [42] then showed that the computational complexity of either transform includes $s-1$ XOR operations to calculate r or r', followed by s XOR operations to calculate the y_i's or x_i's. Therefore, in total it takes $2s-1$ XOR operations to compute either the s input elements or the s output elements from the others.

Informally, it can be observed that if any pair of y_i's are missing, each of the x_i's can take as many values as the y_i's can take.

Generalizing bastion AONT, we define an *asymmetric* (t_i, t_o, s, v)*-AONT* as follows.

Definition 4.2.1. *Suppose s, t_i, and t_o are three positive integers, where $t_i \leq t_o \leq s$. Let*

$$\mathbf{X_1}, \ldots, \mathbf{X_s}, \mathbf{Y_1}, \ldots, \mathbf{Y_s}$$

be random variables taking on values in a finite set Σ of size v. These $2s$ random variables define an asymmetric (t_i, t_o, s, v)-AONT *((t_i, t_o, s, v)-AsymAONT) provided that the following conditions are satisfied:*

1. $\mathsf{H}(\mathbf{Y_1}, \ldots, \mathbf{Y_s} \mid \mathbf{X_1}, \ldots, \mathbf{X_s}) = 0$.
2. $\mathsf{H}(\mathbf{X_1}, \ldots, \mathbf{X_s} \mid \mathbf{Y_1}, \ldots, \mathbf{Y_s}) = 0$.
3. *For all $\mathcal{X} \subseteq \{\mathbf{X_1}, \ldots, \mathbf{X_s}\}$ with $|\mathcal{X}| = t_i$, and for all $\mathcal{Y} \subseteq \{\mathbf{Y_1}, \ldots, \mathbf{Y_s}\}$ with $|\mathcal{Y}| = t_o$, it holds that*

$$\mathsf{H}(\mathcal{X} \mid \{\mathbf{Y_1}, \ldots, \mathbf{Y_s}\} \setminus \mathcal{Y}) = \mathsf{H}(\mathcal{X}). \tag{4.2}$$

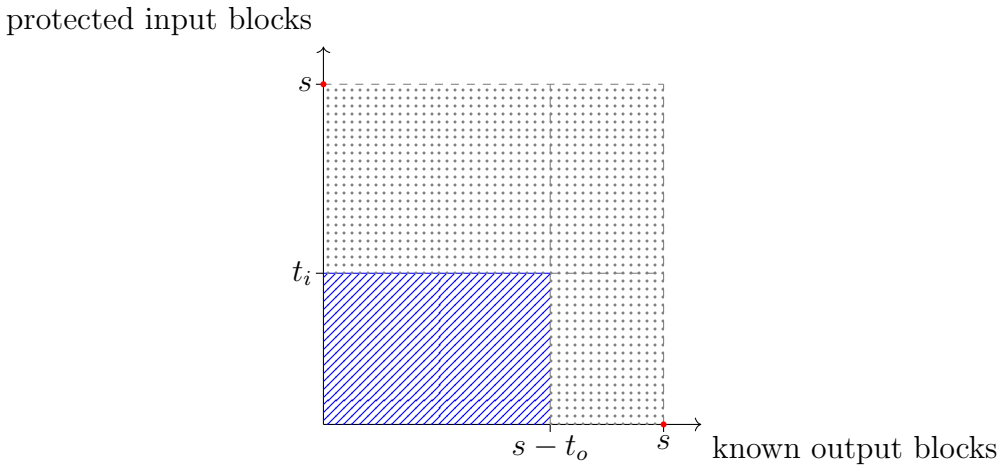


Figure 4.2: The behavior of a (t_i, t_o, s, v)-AsymAONT for different numbers of available output blocks.

Based on this definition, a (t, s, v)-AONT is equivalent to a (t, t, s, v)-AsymAONT.

Figure 4.2 shows this behavior. The area hatched in blue indicates the number of input blocks which are protected upon the availability of that many output blocks to the adversary.

If the output elements of a (t_i, t_o, s, v)-AsymAONT are $\mathbb{F}_q$-linear functions of the input elements, it is a *linear* (t_i, t_o, s, q)-AsymAONT. In particular, a bastion AONT is a linear $(1, 2, s, q)$-AsymAONT for even values of s, where q is a power of 2. We will discuss linear AsymAONTs in detail in the next subsection.

In the array representation of a (t_i, t_o, s, q)-AsymAONT, fixing any t_o output coordinates does not yield any information about any t_i input coordinates.

The following is a straightforward generalization of Theorem 2.2.1.

Theorem 4.2.1. *[14] Let $t_i \leq t_o \leq s$. A (t_i, t_o, s, v)-AsymAONT is equivalent to a $(v^s, 2s, v)$ array that is unbiased with respect to the following sets of columns:*

1. $\{1, \ldots, s\}$,
2. $\{s + 1, \ldots, 2s\}$,
3. $I \cup J$, *for any sets* I *and* J *where* $I \subseteq \{1, \ldots, s\}$, $|I| = t_i$, $J \subseteq \{s + 1, \ldots, 2s\}$, $|J| = s - t_o$, *and* $t_i \leq t_o$.

Note that, when $t_i = t_o$ in Theorem 4.2.1, we obtain Theorem 2.2.1.

The following result is an immediate generalization of Corollary 2.2.4.

Theorem 4.2.2. *Let $t_i \leq t_o \leq s$. Suppose there exists an $OA(s, 2s, v)$. Then there exists a (t_i, t_o, s, v)-AsymAONT for all t_i and t_o such that $1 \leq t_i \leq t_o \leq s$.*

From the definition of split orthogonal arrays from Subsection 1.4.2.2, we can obtain the following theorem.

Theorem 4.2.3. *Suppose there exists a (t_i, t_o, s, v)-AsymAONT. Then there exists an $SOA(t_i, s - t_o, s, s, v)$.*

Proof. In the array representation of a (t_i, t_o, s, q)-AsymAONT, if we set $n_1 = s, n_2 = s, t_1 = t_i$, and $t_2 = s - t_o$, then fixing any subset of t_2 output coordinates does not yield any information about any t_1 input coordinates. Hence, the array is unbiased with respect to any $s - t_o + t_i$ columns where t_i columns are chosen from the first set of columns and $s - t_o$ columns are chosen from the second set of columns. Therefore the array is an SOA$(t_i, s - t_o, s, s, v)$. □

4.2.1 Linear Asymmetric AONTs

If we focus on *linear* AsymAONTs and use the same definitions for I, J, and $M(I, J)$ as we used for previous linear AONTs, the following theorem holds.

Lemma 4.2.4. *Suppose that q is a prime power and M is an invertible s by s matrix with entries from $\mathbb{F}_q$. Let $\mathcal{X} \subseteq \{X_1, \ldots, X_s\}$, $|\mathcal{X}| = t_i$, let $\mathcal{Y} \subseteq \{Y_1, \ldots, Y_s\}$, $|\mathcal{Y}| = t_o$, and let $t_i \leq t_o \leq s$. Then the function $\phi(\mathbf{x}) = \mathbf{x}M^{-1}$ satisfies (??) with respect to $\mathcal{X}$ and $\mathcal{Y}$ if and only if the submatrix $M(I, J)$ is of rank t_i, where $I = \{i : X_i \in \mathcal{X}\}$ and $J = \{j : Y_j \in \mathcal{Y}\}$.*

Proof. Let $\mathbf{x}' = (x_i : i \in I)$. We have $\mathbf{x}' = \mathbf{y}M(I, \{1, \ldots, s\})$. Now assume that y_j is fixed for all $j \notin J$. Then we can write $\mathbf{x}' = \mathbf{y}'M(I, J) + \mathbf{c}$, where $\mathbf{y}' = (y_j : j \in J)$ and $\mathbf{c}$ is a vector of constants. If $M(I, J)$ is of rank t_i, then $\mathbf{x}'$ is completely undetermined, in the sense that $\mathbf{x}'$ takes on all values in $(\mathbb{F}_q)^{t_i}$ as $\mathbf{y}'$ varies over $(\mathbb{F}_q)^{t_o}$. On the other hand, if $t' = \mathsf{rank}(M(I, J)) < t_i$, then $\mathbf{x}'$ can take on only $(\mathbb{F}_q)^{t'}$ possible values. □

Corollary 4.2.5. *Suppose that q is a prime power, $t_i \leq t_o \leq s$, and M is an invertible s by s matrix with entries from $\mathbb{F}_q$. M is a (t_i, t_o, s, q)-AsymAONT if and only if all t_i by t_o submatrices of M are of rank t_i.*

Bastion AONTs only exist for even values of s because for odd values of s the transform is not invertible. Using the matrix representation, we can construct its odd counterpart, i.e., $(1, 2, s, 2)$-AsymAONTs, for odd values of s.

Construction 4.2.6. *Let s be an odd integer and let the matrix M^{-1} be an $s \times s$ matrix with a left bottom $J_{s-1} - I_{s-1}$ submatrix and 1's along the first row and last column. For example, for $s = 5$, M^{-1} is*

$$M_{5\times5}^{-1} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \end{pmatrix}.$$

A matrix with the suggested structure is invertible and its inverse is an $s \times s$ matrix with a right top I_{s-1} submatrix and 1's along the last row and first column. Hence

$$M_{5\times5} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

Similar to bastion AONT, this transform can also be represented by XOR functions. If $(x_1, x_2, \ldots, x_s)$ and $(y_1, y_2, \ldots, y_s)$ are the input elements and output elements, respectively, the transformation can be computed as follows.

1. *for $1 \leq i \leq s-1$, $y_i = x_1 \oplus x_{i+1}$, and*
2. *$y_s = \bigoplus_{i=1}^{s} x_i$.*

The inverse is computed as follows.

1. *$x_1 = \bigoplus_{i=2}^{s} y_i$, and*
2. *for $2 \leq i \leq s$, $x_i = x_1 \oplus y_{i-1}$.*

Theorem 4.2.7. *Any s by s matrix that is generated using Construction 4.2.6 is a $(1, 2, s, 2)$-AsymAONT.*

Proof. To show that such matrices are $(1, 2, s, 2)$-AsymAONT, we need to show: 1) they are invertible and 2) all their 1 by 2 submatrices have ranks greater than zero. Since the construction provides the inverse of the matrices, we know they are invertible. A matrix generated using Construction 4.2.6 has at most one 0 in each row. Hence, all its 1 by 2 submatrices are of rank 1. □

Computation of the transform and its inverse requires $2s-2$ and $2s-3$ XOR operations, respectively. Hence, this construction maintains both the security and the efficiency of bastion AONT for the odd values of s.

Remark 4.2.1. *The construction given by Stinson [42] and used by Karame et al. [21] and Construction 4.2.6 prove the existence of $(1, 2, s, 2)$-AsymAONTs, for any positive integers $s \geq 2$. The same transform can be applied on n bits, $n > 0$, instead of one bit as an element, to obtain a $(1, 2, s, 2^n)$-AsymAONTs. Therefore, $(1, 2, s, 2^n)$-AsymAONTs exist for any integer $s > 1$ and $n > 0$.*

Next, we show that the construction presented by Stinson [42] and used by Karame et al. [21] works for another set of parameters, namely $(2, s-1, s, 2)$-AsymAONT, as long as s is even. Before that, we provide a construction that covers the same parameters for odd values of s as well.

Construction 4.2.8. *Let $B_s = (b_{i,j})$ be the $s \times s$ matrix constructed as follows:*

1. *Start with an $s \times s$ all-zeros matrix.*
2. *Set the entries along the main diagonal, the last row, and the last column to 1.*
3. *Set $b_{1,2} = 1$ and $b_{s,s} = 0$.*

The matrix A_s is defined as follows:

$$A_s = \begin{cases} B_s & \text{if } 2 \nmid s, \\ J_s - I_s & \text{if } 2 \mid s. \end{cases}$$

For $s = 5, 6$, the A_s matrices are

$$A_5 = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix} \text{ and } A_6 = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}.$$

We now prove the asymmetric-AONT properties of A_s.

Theorem 4.2.9. *For any odd integer $s \geq 5$, A_s is a $(2, s-1, s, 2)$-AsymAONT.*

Proof. To prove the invertibility of A_s for odd values of s, consider the process of Gaussian elimination happening from the top row down towards the last row, with the entries along the main diagonal used as the pivots. The row reduction operation for the first row affects only the last row, by turning the first two entries of the last row to 0's and its last entry to 1. The row reduction on the second row does not change any rows because the second row is the only row, other than the first row, with a 1 in the second column. From the third to the second from the last row, the operation only changes the values of 1's underneath the pivot to 0's. It also flips the value of the right bottom entry. After the second from last row reduction operation, the value of the right bottom entry is 1, and since $s-3$ is even, after the row reduction on the second from the last row, the bottom right entry has the value 1. Therefore, A_s is a full rank matrix.

Now to prove that any $2 \times (s-1)$ submatrix has rank two, we show that at least two of $(0,1)^T, (1,0)^T$, and $(1,1)^T$ occur as columns in any such submatrix. First, consider the first $s-1$ rows of the matrix. For any $(s-1)$-subset of columns, the subset must either intersect the main diagonal at two entries, or it is guaranteed to intersect the main diagonal at one entry and includes the last column. In the former case, any two rows will have exactly one $(0,1)^T$ and one $(1,0)^T$, and for the latter case, any two rows will contain either of $(0,1)^T$ or $(1,0)^T$, and $(1,1)^T$; hence, the submatrix has rank 2. Now, let us consider the submatrices formed by the last row and any other row. The $2 \times s$ submatrix created by these rows has at least one copy of each of $(0,1)^T, (1,0)^T$, and $(1,1)^T$. Therefore the any 2 by $(s-1)$ submatrix contains at least two of those columns. Hence again, the rank of the submatrix is 2. □

Remark 4.2.2. *As shown in Example 3.2.1, $(2,3,2)$-AONTs do not exist, which means $(2,2,3,2)$-AsymAONTs do not exist. Therefore, we start our analysis of $(2, s-1, s, 2)$-AsymAONTs from $s = 5$.*

Theorem 4.2.10. *For any even integer $s \geq 4$, A_s is a $(2, s-1, s, 2)$-AsymAONT.*

Proof. The invertibility of A_s for even values of s, over $\mathbb{F}_2$ is already proven by Stinson [42]. Regarding the rank of $2 \times (s-1)$ submatrices, it can be observed that any choice of $s-1$ columns will contain at most $s-2$ copies of $(1,1)^T$ and at least one of $(0,1)^T$ or $(1,0)^T$. Therefore, the rank of the submatrix is exactly 2, and A_s is an AsymAONT. □

Corollary 4.2.11. *For all integers $s \geq 4$, A_s is a $(2, s-1, s, 2)$-AsymAONT.*

Another approach to construct asymmetric AONTs is to use t-AONTs or other asymmetric AONTs. The following statements will present various such constructions.

Lemma 4.2.12. *If $t_i \leq t_o \leq s$, the existence of a linear (t_i, t_o, s, q)-AsymAONT implies the existence of a linear $(t_i, t_o, s-1, q)$-AsymAONT.*

Proof. Let M be a matrix for a linear (t_i, t_o, s, q)-AsymAONT. Since M is invertible, if we calculate its determinant using the cofactor expansion of M with respect to its first row, at least one of the $(s-1) \times (s-1)$ submarices is invertible. Regarding the asymmetric AONT property, any $t_i \times t_o$ submatrix of M, including those in the invertible submatrix, are of rank t_i. Hence, the invertible submatrix is a $(t_i, t_o, s-1, q)$-AsymAONT. □

Lemma 4.2.13. *If $t_i \leq t_o \leq s$, then the existence of a linear (t_i, t_o, s, q)-AsymAONT implies the existence of a linear (t_i, t'_o, s, q)-AsymAONT for any $t'_o \geq t_o$.*

Proof. Consider the matrix representation of the linear $(t_i, t_o, s, q)-$AONT. Every $t_i \times t'_o$ submatrix is rank t_i, because all its $t_i \times t_o$ submatrices are of rank t_i. □

Corollary 4.2.14. *Suppose $t_o \leq s$. Then the existence of a linear $(t, s, q)-$AONT implies the existence of a linear (t, t_o, s, q)-AsymAONT for any $t_o \geq t$.*

Lemma 4.2.15. *If $t_i \leq t_o \leq s$, then the existence of a linear (t_i, t_o, s, q)-AsymAONT implies the existence of a linear (t'_i, t_o, s, q)-AsymAONT for any $t'_i \leq t_i$.*

Proof. Suppose there is a way to obtain information about t'_i input elements in the absence of t_o input elements. This will contradict the condition (4.2). □

Corollary 4.2.16. *Assume $t_i \leq s$. Then the existence of a linear $(t, s, q)-$AONT implies the existence of a linear (t_i, t, s, q)-AsymAONT for any $t_i \leq t$.*

Lemma 4.2.17. *The existence of a linear (t_i, t_o, s, q)-AsymAONT does not necessarily imply the existence of a linear (t_i, s, q)-AONT.*

Proof. The statement will be proven by a counterexample. Consider the linear $(2, 3, 4, 2)$-AsymAONT presented by the following matrix

$$\begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}.$$

While every 2×3 submatrix of the matrix above is of rank 2, $(2, s, 2)$-AONTs do not exist if $t > 2$, as stated in Theorem 2.5.4. □

Theorem 4.2.18. *In a linear (t_i, t_o, s, q)-AsymAONT, $s \leq (t_o - t_i + 1)(q^{t_i} - 1)/(q - 1)$.*

Proof. Fix any t_i rows of the matrix corresponding to the inverse of the AsymAONT. There are q^{t_i} possible t_i-tuples for any given column. However, the AsymAONT condition requires that any t_i by t_o submatrix must be of rank t_i. We can replace an all-zero t_i-tuple with any other t_i-tuple, and it does not impact the invertibility of any t_i by t_o submatrix. Hence, we can assume that there is no all-zero t_i-tuple among the columns of the chosen t_i rows. Therefore, we are left with $q^{t_i} - 1$ possible combinations for columns. For any two t_i-tuples, a and b, define $a \sim b$ if there is a nonzero element $\alpha \in \mathbb{F}_q$ such that $a = \alpha b$. Clearly $\sim$ is an equivalence relation, and there are $(q^{t_i} - 1)/(q - 1)$ equivalence classes, each of size $q - 1$.

Now we use proof by contradiction to show that in the t_i by s submatrix, the number of columns from each equivalence class is upper-bounded by $t_o - t_i + 1$. If there are at least $t_o - t_i + 2$ columns from an equivalence class in the $t_i \times s$ submatrix, it is possible to choose $t_o - t_i + 2$ of those columns together with any other $t_i - 2$ columns. Since the latter set of $t_i - 2$ columns does not contribute more than $t_i - 2$ to the rank, and all the equivalent columns contribute at most 1, the rank of the $t_i \times t_o$ submatrix formed by all the selected columns cannot have a rank greater than $t_i - 1$, which is a contradiction to the AsymAONT property of the matrix. Therefore, each column can appear at most $t_o - t_i + 1$ times.

Therefore, there are $(q^{t_i} - 1)/(q - 1)$ equivalent classes, and members of each can appear at most $t_o - t_i + 1$ times. Hence, the matrix cannot have more than $(t_o - t_i + 1)(q^{t_i} - 1)/(q - 1)$ columns. □

Theorem 4.2.19. *Suppose $t_i = 2$. Then*

$$s \leq \max\{1 + (t_o - 2)(q + 1), 2 + (t_o - 1)(q - 1)\}.$$

Proof. We divide the proof into two cases. Consider a 2 by s submatrix and let a_0 be the number of $(0, 0)^T$ columns in this submatrix.

case (1) Suppose $a_0 \geq 1$ for some 2 by s submatrix. We claim that this submatrix contains at most $t_o - a_0 - 1$ columns from any one equivalence class C_i, as introduced in the proof of Theorem 4.2.18. This follows because $t_o - a_0$ columns from one equivalence class, together with the a_0 columns of 0's, would yield a submatrix having rank 1. Excluding the column of two 0's, there are $q + 1$ possible equivalence classes of columns. Therefore,

$$s \leq a_0 + (t_o - a_0 - 1)(q + 1) \leq 1 + (t_o - 2)(q + 1).$$

case (2) Suppose $a_0 = 0$ for every 2 by s submatrix. There can be at most one 0 in each column of the s by s matrix, so there are at most s occurrences of 0 in the entire matrix. Therefore, there must be two rows that contain a total of at most two 0's. We focus on this 2 by s submatrix.

Let the number of zeros in this 2 by s submatrix be denoted by a; we have noted that $a \leq 2$. In the $s - a$ columns that do not contain a 0, there are at most $t_o - 1$ columns from any equivalence class C_i. Note that we have excluded two C_i's, i.e., $(*, 0)^T$ and $(0, *)^T$, so

$$s \leq a + (t_o - 1)(q - 1) \leq 2 + (t_o - 1)(q - 1).$$

Since one of the two cases must hold, we have

$$s \leq \max\{1 + (t_o - 2)(q + 1), 2 + (t_o - 1)(q - 1)\}.$$

□

We note that

$$1 + (t_o - 2)(q + 1) < (t_o - 1)(q + 1)$$

and

$$2 + (t_o - 1)(q - 1) < (t_o - 1)(q + 1),$$

so

$$\max\{1 + (t_o - 2)(q + 1), 2 + (t_o - 1)(q - 1)\} < (t_o - 1)(q + 1).$$

Hence the bound from Theorem 4.2.19 is an improvement on Theorem 4.2.18 when $t_i = 2$.

Remark 4.2.3. *For positive integers t_i and s, where $t_i \leq s$, and a prime power q, I_s, the $s \times s$ identity matrix, is a (t_i, s, s, q)-AsymAONT.*

Table 4.1: Examples of bounds by Theorems 4.2.18 and 4.2.19.

t_i	q	Upper bound for s	Justification
2	2	$t_o + 1$ for $t_o = 2, 3$, and $3t_o - 5$ for $t_o \geq 3$	Theorem 4.2.19
2	3	$2t_o$ for $t_o = 2, 3$, and $4t_o - 7$ for $t_o \geq 4$	Theorem 4.2.19
2	4	$3t_o - 1$ for $t_o = 2, 3, 4$, and $5t_o - 9$ for $t_o \geq 4$	Theorem 4.2.19
3	3	$13(t_o - 2)$	Theorem 4.2.18
3	4	$40(t_o - 2)$	Theorem 4.2.18
3	5	$121(t_o - 2)$	Theorem 4.2.18

Let $\mathcal{T}_o(t_i, s, q) = \{t_o : \text{a } (t_i, t_o, s, q)\text{-AsymAONT exists}\}$. From Remark 4.2.3, we know that $\mathcal{T}_o(t_i, s, q)$ is not empty because $s \in \mathcal{T}_o(t_i, s, q)$. From Definition 4.2.1, we know that all elements of $\mathcal{T}_o$ are greater than or equal to t_i. Hence, there exists a minimum value in this set, which we denote as $\mu_o(t_i, s, q)$.

In particular, if $\mu_o(t_i, s, q) = t_i$, then there exists a (t_i, s, q)-AONT. Accordingly, for fixed values of t, s, and v, $\mu_o(t, s, q) - t$ can be used as another measure, i.e., besides the t-density of a transform, of showing how close can we get to a (t, s, q)-AONT.

4.2.2 Computational Results

In order to examine the bound presented in Theorem 4.2.19, and Table 4.1, we used a non-exhaustive computer search. The results are presented in Examples 4.2.1 to 4.2.17.

Example 4.2.1. *A linear* $(2, 4, 6, 2)$*-AsymAONT:*

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

Example 4.2.2. *A linear* $(2, 5, 8, 2)$*-AsymAONT:*

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

Example 4.2.3. *A linear* $(2, 6, 10, 2)$*-AsymAONT:*

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}.$$

Example 4.2.4. *A linear* $(2, 7, 12, 2)$*-AsymAONT:*

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

Example 4.2.5. *A linear* $(2, 3, 6, 3)$*-AsymAONT:*

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 2 & 2 \\ 1 & 1 & 0 & 2 & 1 & 2 \\ 1 & 1 & 2 & 0 & 2 & 1 \\ 1 & 2 & 1 & 2 & 0 & 1 \\ 1 & 2 & 2 & 1 & 1 & 0 \end{pmatrix}.$$

Example 4.2.6. *A linear* $(2,3,6,3)$*-AsymAONT:*

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 2 & 2 \\ 1 & 1 & 0 & 2 & 1 & 2 \\ 1 & 1 & 2 & 0 & 2 & 1 \\ 1 & 2 & 1 & 2 & 0 & 1 \\ 1 & 2 & 2 & 1 & 1 & 0 \end{pmatrix}.$$

Example 4.2.7. *A linear* $(2,3,6,3)$*-AsymAONT:*

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 2 & 2 \\ 1 & 1 & 0 & 2 & 1 & 2 \\ 1 & 1 & 2 & 0 & 2 & 1 \\ 1 & 2 & 1 & 2 & 0 & 1 \\ 1 & 2 & 2 & 1 & 1 & 0 \end{pmatrix}.$$

Example 4.2.8. *A linear* $(2,4,8,3)$*-AsymAONT:*

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 2 \\ 1 & 0 & 1 & 1 & 0 & 0 & 2 & 1 \\ 1 & 0 & 1 & 2 & 1 & 2 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 2 & 0 & 2 \\ 1 & 1 & 0 & 2 & 2 & 0 & 1 & 0 \\ 1 & 1 & 2 & 0 & 0 & 1 & 2 & 0 \\ 1 & 2 & 1 & 0 & 2 & 0 & 0 & 2 \end{pmatrix}.$$

Example 4.2.9. *A linear* $(2, 5, 10, 3)$*-AsymAONT:*

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 2 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 2 & 1 \\ 1 & 0 & 0 & 1 & 2 & 1 & 2 & 2 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 2 & 0 & 2 & 0 & 2 \\ 1 & 0 & 1 & 2 & 0 & 0 & 2 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 2 & 2 & 0 & 1 & 0 \\ 1 & 1 & 0 & 2 & 0 & 0 & 1 & 2 & 2 & 0 \\ 1 & 1 & 1 & 0 & 2 & 0 & 2 & 1 & 2 & 2 \\ 1 & 1 & 2 & 1 & 0 & 2 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Example 4.2.10. *A linear* $(2, 6, 12, 3)$*-AsymAONT:*

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 2 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 2 & 1 \\ 1 & 0 & 0 & 1 & 1 & 2 & 1 & 1 & 2 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 2 & 1 & 2 & 0 & 2 \\ 1 & 0 & 1 & 0 & 2 & 1 & 1 & 0 & 2 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 2 & 2 & 2 & 1 & 0 & 2 & 2 & 0 \\ 1 & 0 & 1 & 2 & 0 & 2 & 0 & 2 & 2 & 1 & 1 & 0 \\ 1 & 0 & 2 & 1 & 2 & 0 & 1 & 2 & 0 & 0 & 2 & 2 \\ 1 & 0 & 2 & 2 & 0 & 2 & 1 & 0 & 1 & 2 & 0 & 1 \\ 1 & 1 & 0 & 0 & 2 & 2 & 0 & 2 & 1 & 0 & 2 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 2 & 2 & 0 & 2 & 1 & 0 \end{pmatrix}.$$

Example 4.2.11. *A linear* $(2, 7, 14, 3)$*-AsymAONT:*

$$\begin{pmatrix}
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 2 \\
1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 2 & 1 \\
1 & 0 & 0 & 0 & 1 & 1 & 2 & 1 & 1 & 2 & 0 & 2 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 2 & 1 & 2 & 0 & 0 & 2 \\
1 & 0 & 0 & 1 & 0 & 2 & 1 & 2 & 0 & 2 & 1 & 2 & 1 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & 2 & 2 & 1 & 0 & 2 & 0 & 1 & 1 \\
1 & 0 & 0 & 1 & 2 & 2 & 0 & 0 & 1 & 1 & 0 & 2 & 2 & 1 \\
1 & 0 & 1 & 0 & 0 & 2 & 1 & 1 & 1 & 0 & 2 & 0 & 2 & 0 \\
1 & 0 & 1 & 0 & 1 & 2 & 0 & 2 & 0 & 2 & 2 & 1 & 0 & 2 \\
1 & 0 & 1 & 0 & 2 & 0 & 1 & 0 & 2 & 2 & 0 & 0 & 1 & 1 \\
1 & 0 & 1 & 1 & 2 & 0 & 2 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\
1 & 0 & 1 & 2 & 0 & 0 & 2 & 2 & 2 & 0 & 0 & 2 & 0 & 2 \\
1 & 0 & 2 & 2 & 1 & 2 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0
\end{pmatrix}.$$

Example 4.2.12. *A linear* $(2, 3, 8, 4)$*-AsymAONT:*

$$\begin{pmatrix}
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 1 & 1 & 2 & 2 & 3 & 3 \\
1 & 1 & 0 & 2 & 1 & 3 & 2 & 3 \\
1 & 1 & 2 & 0 & 3 & 1 & 3 & 2 \\
1 & 2 & 1 & 3 & 0 & 3 & 1 & 2 \\
1 & 2 & 3 & 1 & 3 & 0 & 2 & 1 \\
1 & 3 & 2 & 3 & 1 & 2 & 0 & 1 \\
1 & 3 & 3 & 2 & 2 & 1 & 1 & 0
\end{pmatrix}.$$

Example 4.2.13. *A linear* $(2, 4, 8, 4)$*-AsymAONT:*

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 2 \\ 1 & 0 & 1 & 1 & 0 & 0 & 2 & 1 & 2 & 3 \\ 1 & 0 & 1 & 2 & 1 & 2 & 0 & 3 & 0 & 1 \\ 1 & 0 & 1 & 3 & 2 & 1 & 3 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 2 & 3 & 0 & 3 & 2 \\ 1 & 1 & 0 & 2 & 3 & 0 & 2 & 2 & 1 & 0 \\ 1 & 1 & 2 & 0 & 0 & 1 & 0 & 3 & 1 & 3 \\ 1 & 2 & 1 & 0 & 3 & 0 & 1 & 0 & 3 & 1 \\ 1 & 2 & 3 & 1 & 2 & 3 & 2 & 3 & 1 & 1 \end{pmatrix}.$$

Example 4.2.14. *A linear* $(2, 5, 12, 4)$*-AsymAONT:*

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 2 & 2 & 2 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 3 & 1 & 3 & 3 \\ 1 & 0 & 0 & 1 & 2 & 1 & 2 & 3 & 0 & 3 & 0 & 1 \\ 1 & 0 & 0 & 2 & 1 & 2 & 1 & 3 & 2 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 3 & 2 & 0 & 3 & 1 & 2 \\ 1 & 0 & 1 & 0 & 3 & 3 & 2 & 0 & 1 & 1 & 0 & 3 \\ 1 & 0 & 1 & 1 & 0 & 3 & 0 & 2 & 2 & 0 & 2 & 1 \\ 1 & 1 & 0 & 0 & 2 & 3 & 3 & 0 & 2 & 0 & 3 & 2 \\ 1 & 1 & 0 & 2 & 3 & 0 & 2 & 2 & 0 & 0 & 2 & 3 \\ 1 & 1 & 0 & 3 & 0 & 1 & 0 & 2 & 3 & 3 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 3 & 0 & 2 & 3 & 0 \end{pmatrix}.$$

Example 4.2.15. *A linear* $(2, 3, 8, 5)$*-AsymAONT:*

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 2 & 2 & 3 \\ 1 & 1 & 1 & 0 & 0 & 2 & 3 & 2 \\ 1 & 1 & 2 & 1 & 2 & 0 & 4 & 4 \\ 1 & 1 & 4 & 2 & 3 & 1 & 2 & 0 \\ 1 & 2 & 1 & 3 & 4 & 3 & 4 & 1 \\ 1 & 3 & 2 & 4 & 1 & 4 & 3 & 1 \\ 1 & 4 & 3 & 2 & 1 & 3 & 0 & 2 \end{pmatrix}.$$

Example 4.2.16. *A linear* $(2, 3, 12, 5)$*-AsymAONT:*

$$\left(\begin{array}{cccccccccccc}
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 2 & 3 & 3 \\
1 & 0 & 1 & 1 & 0 & 0 & 1 & 3 & 3 & 4 & 4 & 4 \\
1 & 0 & 1 & 2 & 1 & 2 & 3 & 0 & 0 & 3 & 1 & 2 \\
1 & 0 & 1 & 3 & 2 & 3 & 0 & 1 & 2 & 0 & 2 & 1 \\
1 & 0 & 2 & 2 & 3 & 3 & 4 & 3 & 4 & 2 & 0 & 0 \\
1 & 1 & 0 & 1 & 0 & 2 & 4 & 0 & 4 & 1 & 2 & 3 \\
1 & 1 & 0 & 3 & 3 & 4 & 0 & 2 & 3 & 4 & 1 & 0 \\
1 & 1 & 2 & 0 & 2 & 0 & 3 & 4 & 2 & 3 & 0 & 4 \\
1 & 1 & 3 & 4 & 1 & 3 & 2 & 2 & 4 & 3 & 4 & 1 \\
1 & 2 & 3 & 0 & 3 & 4 & 1 & 4 & 0 & 0 & 4 & 2 \\
1 & 3 & 4 & 1 & 2 & 4 & 3 & 0 & 1 & 0 & 3 & 0
\end{array}\right)$$

Example 4.2.17. *A linear* $(2, 3, 9, 7)$*-AsymAONT:*

$$\left(\begin{array}{ccccccccc}
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 1 & 1 & 2 & 2 & 3 & 3 \\
1 & 0 & 1 & 0 & 2 & 1 & 3 & 2 & 4 \\
1 & 0 & 2 & 2 & 3 & 3 & 5 & 5 & 0 \\
1 & 1 & 0 & 2 & 0 & 5 & 6 & 1 & 5 \\
1 & 1 & 1 & 3 & 3 & 0 & 0 & 4 & 6 \\
1 & 1 & 2 & 0 & 6 & 4 & 1 & 0 & 3 \\
1 & 1 & 3 & 1 & 4 & 3 & 4 & 2 & 2 \\
1 & 2 & 1 & 6 & 4 & 4 & 5 & 6 & 1
\end{array}\right).$$

Table 4.2 demonstrates the maximum values of s for which an asymmetric AONT was found by a computer search, for some fixed values of t_i, t_o, and q. The computer search forced a fix first row and column on the matrices. Hence the results are not necessarily the maximum values possible. It should be noticed that as the alphabet size and t_o increase, the search takes more time and some processes were terminated before they can search the entire search domain, even with the given restriction on the first row and column. These instances are distinguished by the $*$ near the reported value of s.

Table 4.2: Lower bounds for s for $(2, t_o, s, q)$-AsymAONTs

t_i	t_o	q	max. value of s	Upper bound from Thrm. 4.2.19
2	4	2	6	7
2	5	2	8	10
2	6	2	10	13
2	7	2	12	16
2	8	2	14	19
2	3	3	6	6
2	4	3	8	9
2	5	3	10	13
2	6	3	12	17
2	7	3	14	21
2	3	4	8	8
2	4	4	10∗	11
2	5	4	13∗	16
2	3	5	8∗	10
2	4	5	12∗	14
2	3	7	9∗	14

4.3 Restricted t-AONT

The generalization presented in this section is an obvious generalization of ℓ-restricted AONTs (due to Pham et al. [33]) to ℓ-restricted t-AONT. Pham et al. [33] introduced *R-restricted AONTs* as a structure with fewer constraints than regular AONT. Therefore, they are more likely to exist. Pham et al. [33] introduced the following definition for an R-restricted AONT.

Definition 4.3.1. *Let R be a fixed t-subset of $\{1, 2, \ldots, s\}$ and let*

$$\mathbf{X_1}, \ldots, \mathbf{X_s}, \mathbf{Y_1}, \ldots, \mathbf{Y_s}$$

*be random variables taking on values in a finite set Σ of size v. These $2s$ random variables define an R-*restricted AONT *provided that the following conditions are satisfied:*

1. $\mathsf{H}(\mathbf{Y_1}, \ldots, \mathbf{Y_s} \mid \mathbf{X_1}, \ldots, \mathbf{X_s}) = 0$.
2. $\mathsf{H}(\mathbf{X_1}, \ldots, \mathbf{X_s} \mid \mathbf{Y_1}, \ldots, \mathbf{Y_s}) = 0$.

3. *For* $\mathcal{Y} = \{\mathbf{Y_i} : i \in R\}$, *it holds that*

$$\mathsf{H}(\mathbf{X_i} \mid \{\mathbf{Y_1}, \ldots, \mathbf{Y_s}\} \setminus \mathcal{Y}) = \mathsf{H}(\mathbf{X_i}) \tag{4.3}$$

for all $i \in \{1, 2, \ldots, s\}$.

In particular, if $R = \{1, 2, \ldots, \ell\}$, the AONT is called an *ℓ-restricted AONT*. An R-restricted AONT is *linear* if each output is a linear combination of the inputs.

Pham et al. [33] use these structures in a setting where there is an unconditionally secure communication channel, with a limited bandwidth, as well as a channel that can be observed by the adversary. In this setting, a portion of the message is sent through the secure channel, while the rest is transmitted over the regular one. Pham et al. [33] design the security of their system based on the adversary's lack of access to the portion of the message sent over the secure channel. Since the sender knows which parts are sent over the secure channel, they do not need to guarantee the security of any input block in the absence of any output block. They only need to prove that it is impossible for the adversary to gain any information about any block, in the absence of the blocks sent over the secure channel.

The above definition can be generalized and extended in various ways. A trivial generalization considers the security of any t blocks, where $t \leq |R|$, in the absence of all the blocks in R, i.e., the blocks sent over the secure channel. Our generalization uses a stronger assumption and considers the security of any $t \leq |R|$ input blocks provided that the adversary can learn at most all the output blocks except for t of the blocks sent over the secure channel. Of course, if there are exactly t blocks sent over the secure channel, then the adversary is assumed to have access to none of them, and these two generalizations are equivalent.

The generalization with the stronger assumption leads to the following new definition of an *R-restricted (t, s, v)-AONT*.

Definition 4.3.2. *Let* $R \subseteq \{1, 2, \ldots, s\}$, *let* $t \leq |R|$, *and let*

$$\mathbf{X_1}, \ldots, \mathbf{X_s}, \mathbf{Y_1}, \ldots, \mathbf{Y_s}$$

be random variables taking on values in the finite set Σ *of size* v. *These* $2s$ *random variables define an* R-restricted (t, s, v)-AONT *provided that the following conditions are satisfied:*

1. $\mathsf{H}(\mathbf{Y_1}, \ldots, \mathbf{Y_s} \mid \mathbf{X_1}, \ldots, \mathbf{X_s}) = 0$.

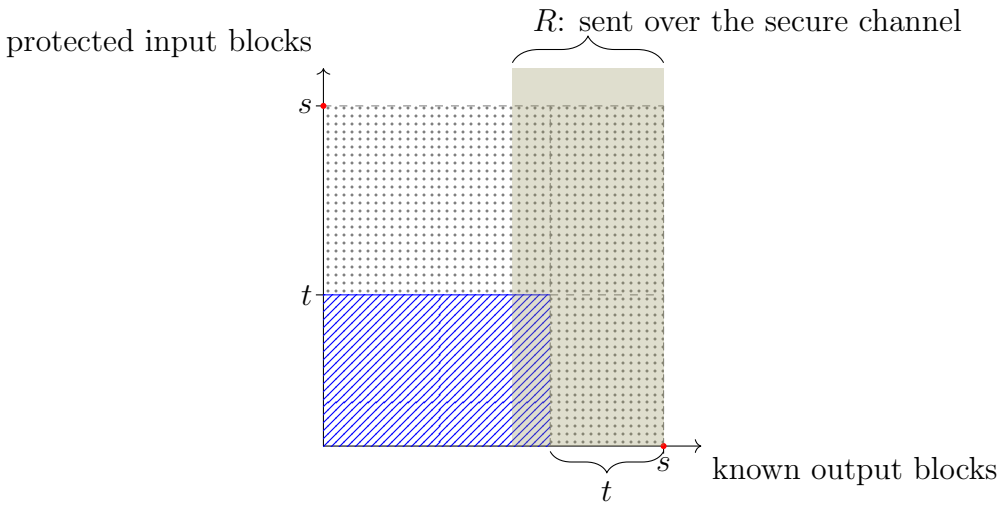


Figure 4.3: The behavior of a R-restricted (t, s, v)-AONT for different numbers of available output blocks.

2. $\mathsf{H}(\mathbf{X_1}, \ldots, \mathbf{X_s} \mid \mathbf{Y_1}, \ldots, \mathbf{Y_s}) = 0$.

3. *For all* $\mathcal{X} \subseteq \{\mathbf{X_1}, \ldots, \mathbf{X_s}\}$ *with* $|\mathcal{X}| = t$, *and for any* $\mathcal{Y} \subseteq \{\mathbf{Y_i}, i \in R\}$, *with* $|\mathcal{Y}| = t$, *it holds that*

$$\mathsf{H}(\mathcal{X} \mid \{\mathbf{Y_1}, \ldots, \mathbf{Y_s}\} \setminus \mathcal{Y}) = \mathsf{H}(\mathcal{X}). \tag{4.4}$$

The array representation of an R-restricted (t, s, v)-AONT is an array with $2s$ columns and v^s rows such that fixing all the output coordinates except for t output coordinates in R does not yield any information about any t input coordinates.

Figure 4.3 presents the behavior of these structures. The area hatched in blue indicates the number of input blocks which are protected upon the availability of that many output blocks to the adversary.

The following is a straightforward generalization of Theorem 2.2.1.

Theorem 4.3.1. *[14] Suppose* s *is a positive integer,* $R \subseteq \{1, 2, \cdots, s\}$, *and* t *is an integer such that* $1 \leq t \leq |R|$. *An* R*-restricted* (t, s, v)*-AONT is equivalent to a* $(v^s, 2s, v)$ *array that is unbiased with respect to the following sets of columns:*

1. $\{1, \ldots, s\}$,

2. $\{s+1, \ldots, 2s\}$,

3. $I \cup J$, *for any sets* I *and* J *where* $I \subseteq \{1, \ldots, s\}$, $|I| = t$, $J \subseteq \{s+1, \ldots, 2s\}$, $|J| = s - t$, *and* $|R' \setminus J| = t$, *where* $R' = \{i + s : i \in R\}$.

The following result is an immediate generalization of Corollary 2.2.4.

Theorem 4.3.2. *Suppose there exists an* $OA(s, 2s, v)$. *Let* $R \subseteq \{1, 2, \cdots, s\}$ *(this order can be achieved through a permutation of the columns). Then there exists an* R*-restricted* (t, s, v)*-AONT for all* t, $1 \leq t \leq |R|$.

Suppose $\mathcal{Y} = \{1, 2, \ldots, \ell\}$, for $\ell \geq t$. Then using the result from Lemma 2.3.1, the following corollary describes the restricted AONT property in the matrix representation of *linear* restricted AONTs.

Corollary 4.3.3. *Suppose that* q *is a prime power,* $t \leq \ell$, *and* M *is an invertible* s *by* s *matrix with entries from* $\mathbb{F}_q$. M *is a* $\{1, 2, \ldots, \ell\}$*-restricted* (t, s, q)*-AONT if and only if, all* t *by* t *submatrices of* M *that are contained in the first* ℓ *rows of* M *are invertible.*

As mentioned earlier, this relaxation of conditions allows for restricted AONTs with parameters which would not yield an AONT. For instance, while Theorem 2.5.3 and our search results from Chapter 2 showed that $(2, 6, 5)$-AONT and $(2, 9, 9)$-AONT do not exist, Examples 4.3.1 and 4.3.2 present $\{1, 2\}$-restricted AONTs for these parameters, respectively.

Example 4.3.1. *A linear* $\{1, 2\}$*-restricted* $(2, 6, 5)$*-AONT:*

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 2 & 3 & 4 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Example 4.3.2. *A linear* $\{1,2\}$*-restricted* $(2,9,9)$*-AONT:*

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Specifically for $\ell = t$, Corollary 4.3.3 indicates that any t columns of the t by s submatrix, formed by the first t rows, are linearly independent. To construct such t by n matrices, we can use the parity check matrix of maximum distance separable (MDS) codes (see Section 1.4.2 for details). For example, triply extended Reed-Solomon codes [27, p. 323] can be used to construct $\{1,2,3\}$-restricted $(3, 2^n+2, 2^n)$-AONTs as shown by Theorem 4.3.4 and doubly extended Reed-Solomon codes [27, p. 323] can be utilized in the construction of $\{1,2,\ldots,t\}$-restricted $(t, q+1, q)$-AONT as Theorem 4.3.5 states. The other $s-t$ rows of the matrix need to be chosen such that the entire matrix is invertible.

Theorem 4.3.4. *Let* n *be a positive integer and let* $q = 2^n$*. Then a* $\{1,2,3\}$*-restricted* $(3, 2^n+2, 2^n)$*-AONT exists.*

Proof. Let $\omega_1, \omega_2, \ldots, \omega_{q-1}$ be distinct elements in the finite field $\mathbb{F}_q$. The following matrix is the parity check matrix of a $(q+2, q)$ triply extended Reed-Solomon code over $\mathbb{F}_q$ [27, Ch. 11, Theorem 10].

$$H = \begin{pmatrix} 1 & 1 & \cdots & 1 & 1 & 0 & 0 \\ \omega_1 & \omega_2 & \cdots & \omega_{q-1} & 0 & 1 & 0 \\ \omega_1^2 & \omega_2^2 & \cdots & \omega_{q-1}^2 & 0 & 0 & 1 \end{pmatrix}$$

Any three columns for H are linearly independent. Hence, we only need to construct the next $q-1$ rows of the AONT such that the resulting matrix is invertible. This goal can be achieved by choosing rows with an entry of 1 on the main diagonal and 0's elsewhere. Since any three columns of the parity check matrix are linearly independent, the final matrix is a $\{1,2,3\}$-restricted $(3, q+2, q)$-AONT. □

If we use the dual code of the code used in Theorem 4.3.4, we can also construct a $\{1,2,\ldots,q-1\}$-restricted $(q-1, q+2, q)$-AONT.

Theorem 4.3.5. *Let q be a prime power and let $t \leq q+1$. Then a $\{1, 2, \ldots, t\}$-restricted $(t, q+1, q)$-AONT exists.*

Proof. For any value of k, we can construct a doubly extended Reed-Solomon code of length $q+1$ and dimension k [27, Ch. 11, Theorem 9]. The parity check matrix of this code can be extended by k rows such that the final matrix is invertible. Since any $q-k+1$ columns of the parity check matrix are linearly independent, the final matrix is a $\{1, 2, \ldots, q-k+1\}$-restricted $(q-k+1, q+1, q)$-AONT. □

4.4 Rectangular AONT

Before discussing the use of AONTs in information dispersal, we need to discuss confidentiality and availability of threshold schemes, in general, for distributing files. To distribute files among several storage devices, we divide each file into fragments, called *shares*, and store each share on a different storage device. To securely distribute files, our scheme requires the following properties.

1. The data stored on certain subsets of the storage devices are sufficient to recover the original file, which we call *availability condition*. In threshold secret sharing schemes, these subsets are referred to as *authorized sets*.

2. The data stored on certain subsets of the storage devices is not sufficient to obtain any information about the original file, which we call *confidentiality condition*. In threshold secret sharing schemes, these subsets are referred to as *forbidden sets*.

For instance, for a (t, n)-threshold scheme with uniformly distributed shares, the availability condition is that at least t storage devices are contributing their shares. Any other subset of the participating storage devices, i.e., any subset of $t-1$ or fewer storage devices, satisfies the confidentiality condition. For an (ℓ, t, n)-ramp scheme, these conditions are participation of at least t storage devices and participation of at most ℓ storage devices, respectively. The availability and confidentiality conditions together may cover all possible subsets, as they do in threshold schemes, or they may not cover some subsets. For example, in ramp schemes, all the subsets with more than ℓ and fewer than t shares are not covered by either of the conditions. If we consider security from an information theoretic aspect, the confidentiality condition for AONTs is satisfied only if the adversary does not have access to any output block, and the availability condition requires a user to know all the output blocks.

From the definitions, it is obvious that the availability condition and the confidentiality condition cannot be satisfied simultaneously. Based on the behavior of these conditions in ramp schemes and AONTs, we know that it is possible for some subsets not to satisfy either of the conditions. However, the possibility of an adversary gaining access to such subsets of shares is not ruled out. This scenario is very important when we are using AONTs, as it covers the entire power set of shares minus the empty set and the set of all shares.

For Shamir secret sharing, ramp schemes, and AONTs, since the availability and confidentiality conditions are determined by the number of shares available, the conditions are described by thresholds, which we will call availability threshold and confidentiality threshold, respectively. In the following subsections, we will discuss different parameters and properties of AONTs that impact the the availability threshold, and the system's behavior in between these two thresholds.

4.4.1 Availability Threshold and Rectangular AONTs

It is possible for certain servers to be permanently or temporarily unavailable due to network issues, damage, etc. For the purpose of information dispersal, it is important to consider the unavailability of a few servers at the time of retrieval. Therefore, the availability threshold needs to be extended. We consider a transformation from s input blocks to n output blocks, where $n \geq s$, that satisfies the following conditions:

1. the s input blocks determine all n output blocks.
2. any s output blocks can recover all the s input blocks.
3. any $s - t$ output blocks do not yield any information about any t input blocks.

Oliveira et al. [32] introduced matrices with such properties and provided a brief formulation of them as all-or-nothing transforms. In this section, we study these structures as general and linear AONTs. To achieve this goal, we define rectangular all-or-nothing transforms as follows.

Definition 4.4.1. *Suppose s, n, and t are three positive integers, where $t \leq s \leq n$. Let*

$$\mathbf{X_1}, \ldots, \mathbf{X_s}, \mathbf{Y_1}, \ldots, \mathbf{Y_n}$$

*be random variables taking on values in a finite set Σ of size v. These $s + n$ random variables define a (t, s, n, v)-*recAONT *provided that the following conditions are satisfied:*

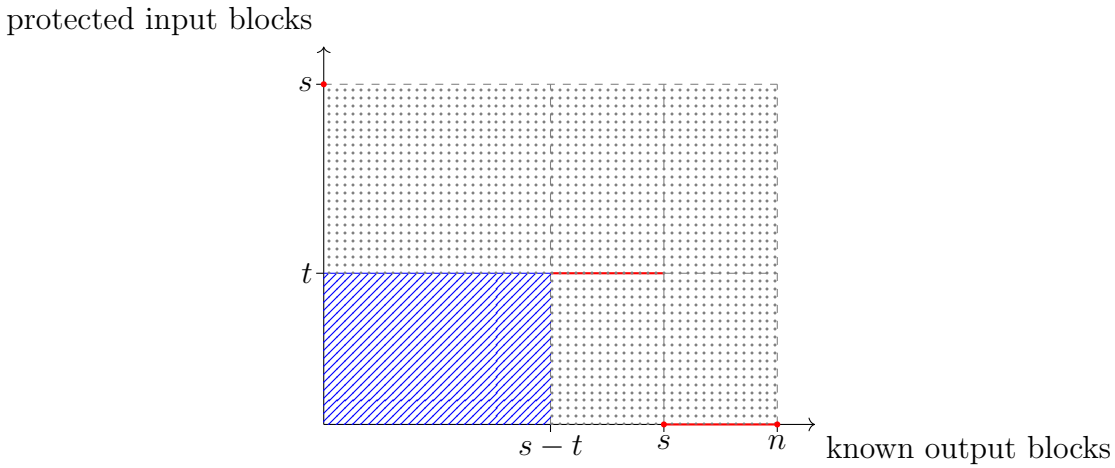


Figure 4.4: The behavior of a (t, s, n, v)-recAONT for different numbers of available output blocks.

1. $\mathsf{H}(\mathbf{Y_1}, \ldots, \mathbf{Y_n} \mid \mathbf{X_1}, \ldots, \mathbf{X_s}) = 0$.
2. *For any s-subset $\mathcal{Y}_s \subseteq \{\mathbf{Y_1}, \ldots, \mathbf{Y_n}\}$,* $\mathsf{H}(\mathbf{X_1}, \ldots, \mathbf{X_s} \mid \mathcal{Y}_s) = 0$.
3. *For all $\mathcal{X} \subseteq \{\mathbf{X_1}, \ldots, \mathbf{X_s}\}$ with $|\mathcal{X}| = t$, and for all $\mathcal{Y} \subseteq \{\mathbf{Y_1}, \ldots, \mathbf{Y_n}\}$ with $|\mathcal{Y}| = n - s + t$, it holds that*

$$\mathsf{H}(\mathcal{X} \mid \{\mathbf{Y_1}, \ldots, \mathbf{Y_n}\} \setminus \mathcal{Y}) = \mathsf{H}(\mathcal{X}). \tag{4.5}$$

For $n = s$, this generalization of the AONT definition matches Definition 2.1.2 for t-AONT, from Chapter 2.

The behavior of rectangular AONTs is shown in Figure 4.4. The area hatched in blue indicates the number of input blocks which are protected upon the availability of that many output blocks to the adversary.

The following is a straightforward generalization of Theorem 2.2.1.

Theorem 4.4.1. *A (t, s, n, v)-recAONT is equivalent to a $(v^s, s + n, v)$ array that is unbiased with respect to the following sets of columns:*

1. $\{1, \ldots, s\}$

2. *any* $J \subseteq \{s+1, \ldots, s+n\}$ *where* $|J| = s$

3. $I \cup J$, *for any sets* I *and* J *where* $I \subseteq \{1, \ldots, s\}$, $|I| = t$, $J \subseteq \{s+1, \ldots, s+n\}$ *and* $|J| = s - t$.

Note that, when $n = s$ in Theorem 4.4.1, we obtain Theorem 2.2.1.

The following result is an immediate generalization of Corollary 2.2.4.

Theorem 4.4.2. *Suppose there exists an* $OA(s, s+n, v)$ *where* $n \geq s$. *Then there exists a* (t, s, n, v)*-recAONT for all* t, $1 \leq t \leq s$.

Recall from Section 1.4.2 that an $\mathrm{OA}(2, k, v)$ is equivalent to $k-2$ mutually orthogonal Latin squares (MOLS) of order v. Many results on MOLS can be found in the *Handbook of Combinatorial Designs* [13]. These results also provide constructions of recAONT with $s = 2$ for alphabet sizes that are not required to be a prime power.

For example, we consider $k = 5$. It is well-known that an $\mathrm{OA}(2, 5, v)$ exists for all $v \geq 4$, $v \neq 6, 10$ [13, p. 126]. Hence we have the following existence result for recAONT.

Corollary 4.4.3. *Suppose* $v \geq 4$, $v \neq 6, 10$. *Then there exists a* $(t, 2, 3, v)$*-recAONT for* $t = 1, 2$.

We now observe that $\mathrm{OA}(2, k, v)$ are equivalent to certain recAONT.

Theorem 4.4.4. *An* $OA(2, k, v)$ *is equivalent to a* $(1, 2, k-2, v)$*-recAONT.*

Proof. Applying Theorem 4.4.2 with $s = 2, t = 1$, it follows that existence of an $\mathrm{OA}(2, k, v)$ implies the existence of a $(1, 2, k-2, v)$-recAONT. For the converse, we observe that the array representation of a $(1, 2, k-2, v)$-recAONT is unbiased with respect to any two columns, and hence it is also the array representation of an $\mathrm{OA}(2, k, v)$. □

The following result due to Bill Martin (private communication) is straightforward.

Theorem 4.4.5. *Suppose there exists a* (t, s, n, v)*-recAONT. Then there exists an* $SOA(t, s-t; s, n; v)$.

Proof. From Theorem 4.4.1 we know that a (t, s, n, v)-recAONT is equivalent to a $(v^s, s+n, v)$-array, that is unbiased with respect to $I \cup J$, for any sets I and J where $I \subseteq \{1, \ldots, s\}$, $|I| = t$, $J \subseteq \{s+1, \ldots, s+n\}$ and $|J| = s - t$. If we set $n_1 = s, n_2 = n, t_1 = t$, and $t_2 = s - t$, then from the definition of split orthogonal arrays (see 1.4.2.2), such an array is an $\mathrm{SOA}(t, s-t; s, n; v)$. □

Hence, from a design theoretic perspective, rectangular AONTs are structures between orthogonal arrays and split orthogonal arrays, in the sense that existence of a suitable orthogonal array implies the existence of a certain recAONT, which in turn implies the existence of a certain split orthogonal array.

Similar to the other types of AONT structures discussed so far, a recAONT is *linear* if its outputs are a linear combination of its inputs. Note that we write a linear recAONT in the form $\mathbf{y} = \mathbf{x}N$, where N is an s by n matrix that satisfies certain properties, as given in the following theorem.

Lemma 4.4.6. *Suppose that q is a prime power and N is an s by n matrix with entries from $\mathbb{F}_q$. Then $\phi(\mathbf{x}) = \mathbf{x}N$, defines a linear (t, s, n, q)-recAONT if and only if the following conditions are satisfied:*

1. *every s by s submatrix of N is invertible, and*
2. *every $(s-t)$ by $(s-t)$ submatrix of N is invertible.*

Proof. Clearly property 1 in Definition 4.4.1 is satisfied if and only if every s by s submatrix of N is invertible. We prove that property 2 holds if and only if every $(s-t)$ by $(s-t)$ submatrix of N is invertible.

Let N' be a matrix consisting of any s columns of N. Then $\mathbf{y}' = \mathbf{x}N'$ is a (t, s, v)-AONT. Therefore, from Corollary 2.3.4, any $(s-t)$ by $(s-t)$ submatrix of N' is invertible. □

4.5 Application

In Section 4.4, we introduced the confidentiality and availability thresholds. Then we discussed the behavior of the scheme when fewer than the confidentiality threshold or more than the availability threshold are available. Now, we focus on the behavior of different AONTs when the number of available shares is smaller than availability threshold and greater than the confidentiality threshold. The behavior of the scheme between these two thresholds is of interest because, under certain circumstances, it could be possible for an adversary to gain access to a set of shares that satisfy neither the confidentiality condition nor the availability condition, yet it is important to know what can the adversary infer from those shares. Since, in rectangular AONTs the availability threshold is s, i.e., the same as corresponding square AONTs, we only consider square AONTs.

In a (t, s, v)-AONT, the availability of fewer than $s - t$ shares does not provide any information about any t blocks of the message. However, this scheme does not guarantee any security regarding more than t blocks or when there are more than $s - t$ shares are available. Informally, using a (t, s, v)-AONT introduces one extra threshold between the confidentiality and availability thresholds, which considers the security of parts of the message. This particular threshold can be extended through range AONTs. In a $([t_1, t_2], s, v)$-rangeAONT, for any $t_1 \leq t \leq t_2$, the guarantees from a (t, s, v)-AONT are provided. Hence, the behavior of the scheme is known for the availability of fewer than the confidentiality threshold shares, any number of shares from t_1 to t_2, and more than the availability threshold shares. In strong AONTs, this range is extended to all the values from the confidentiality threshold to the availability threshold.

In a (t_i, t_o, s, v)-AsymAONT, the availability of fewer than $s-t_o$ shares does not provide any information about any t_i blocks of the message. Similar to a (t, s, v)-AONT, this scheme does not guarantee any security for more than t_i input blocks or when more than $s - t_o$ shares are available. In an R-restricted (t, s, v)-AONT, as long as t shares in R are not available, no information can be obtained about any t blocks of the message. Generally, it is possible to apply extensions similar to those applied on (t, s, v)-AONTs to define range and strong versions of asymmetric AONTs and restricted AONTs. Such extensions would enforce the behavior on all possible t_o values and all possible choices of R.

4.5.1 Information Dispersal using AONTs

In a linear (t, s, n, q)-recAONT, any s output elements can be used to recover all the input blocks. Therefore, these AONTs can be used as an erasure code. This transform will take s blocks as input and output n blocks. Therefore, any set of s outputs can retrieve the original blocks, and any coalition of $s - t$ or fewer parties cannot gain any information about any t input-blocks.

To distribute a document F using a (t, s, n, q)-recAONT, we break the document into s blocks, $F = (x_1, x_2, \ldots, x_s)$. Then we apply the rectangular AONT on these blocks to get n output blocks $(y_1, y_2, \ldots, y_n)$. Each output block y_i is stored on a separate server, S_i, for $1 \leq i \leq n$. To recover the file from these devices, the user needs to collect s distinct y_i values and compute the inverse of the AONT for those blocks to obtain the x_i's.

Figure 4.5 depicts how the output blocks are distributed over servers after an AONT is applied on the input blocks.

To discuss the confidentiality of this scheme, we need to assume that the adversary can access at most $s - t$ of the n shares. Thus, suppose that an adversary confiscates

$s - t$ shares, $\bar{y}_1, \bar{y}_2, \ldots, \bar{y}_{s-t}$, and that the adversary has access to t blocks of a document $\bar{F}$ consisting of the s blocks $\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_s$. Due to the AONT property of the transform, the adversary cannot verify, even probabilistically, that the confiscated shares correspond to the t file blocks.

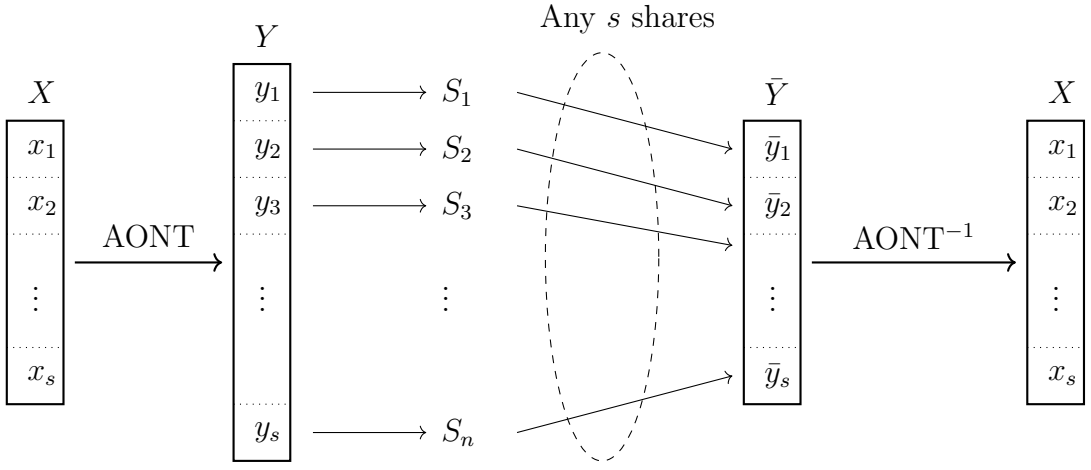


Figure 4.5: Using a (t, s, n, q)-recAONT to distribute a file, X, over n servers, $(S_1, S_2, \ldots, S_n)$, and recovering X from the shares.

Chapter 5

Conclusion

5.1 Summary

Let M be a message that is represented by s blocks/symbols that are chosen from an alphabet of size v. An all-or-nothing transform (AONT) is a bijection that maps these s blocks to s output blocks such that obtaining information about any single input block requires all the output blocks. We have presented five generalizations of unconditionally secure all-or-nothing transforms in Chapter 2 and Chapter 4, namely:

- (t, s, v)-AONTs guarantee that the adversary cannot obtain any information about any t input blocks, as long as t or more output blocks are missing. This behavior is shown in Figure 2.2.

- $([t_1, t_2], s, v)$-rangeAONTs guarantee that the adversary cannot obtain any information about any t input blocks, as long as t or more output blocks are missing, for any $t_1 \leq t \leq t_2$. Figure 4.1 presents the behavior of this transform. Specifically, if t_1 is 1 and t_2 is t, the $([1, t], s, v)$-rangeAONT is called a (t, s, v)-strong AONT.

- (t_i, t_o, s, v)-AsymAONTs guarantee that the adversary cannot obtain any information about any t_i input blocks, as long as t_o or more output blocks are missing. Figure 4.2 shows this behavior.

- R-restricted (t, s, v)-AONTs guarantee that the adversary cannot obtain any information about any t input blocks, as long as t or more of the output blocks sent over a secure channel are not available to them. The set of blocks that are transmitted

using the secure channel is denoted by R. Figure 4.3 presents the behavior of these structures.

- (t, s, n, v)-recAONTs create n output blocks and guarantee that any s output blocks can reconstruct the entire input and the adversary cannot obtain any information about any t input blocks, as long as $n - s + t$ or more output blocks are missing. The behavior of rectangular AONTs is shown in Figure 4.4.

For each of these definitions, we studied the relationship between the AONT structures and other combinatorial structures, for instance orthogonal arrays and unbiased arrays, and we provided instances that are found using computer searches or constructed based on their relationship with other structures. We showed that Cauchy matrices can be used to construct all these transforms. Furthermore, we obtained bounds on parameters that indicate the nonexistence of some of these structures. Some of these bounds where shown to be tight or close to tight by ourselves and others, using computational and theoretical results.

For t-AONTs in particular, we presented two applications: in extended package transform and in a hash-based group signature scheme, in Chapter 2. Also, we ended Chapter 4 with a discussion on the application of (t, s, n, v)-recAONTs in secure distributed storage.

In Chapter 3, we studied almost (t, s, v)-AONT structures, where the transform may fail to protect the security of some sets of t input blocks in the absence of all sets of t output blocks. We defined the ratio of the number of protected sets of t input blocks to the number of all possible sets of t input blocks, as a measure of closeness to a (t, s, v)-AONT. Then we provided examples of such structures using theoretical constructions and computational results. Finally, we introduced bounds on this closeness measure for different parameter sets, specifically for $t = 2$.

In summary, we introduced *five generalizations of all-or-nothing transforms* and showed their application in three different schemes. The use of these transforms allowed us to utilize the schemes with *more flexible parameter sets*. In doing so, we studied the relationship between all-or-nothing transforms and different combinatorial structures, including orthogonal arrays, split orthogonal arrays, and error-correcting codes. We also faced many other interesting problems, which we are going to present in the following section.

5.2 Future Research

Further research on AONTs may be concerned with different aspects of the topic. Some interesting problems that are motivated this research are provided below:

- In this thesis, the main purpose of using computer searches was to find instances of the transforms studied; however, a comprehensive list of instances of a set of parameters goes beyond the scope of this thesis. To find more instances of some types of these all-or-nothing transforms, more elaborate and customized algorithms are needed. For instance, with the exception of a few cases in range AONTs and asymmetric AONTs, we mostly focused on the case where $t = 2$. Many properties of t-AONTs for $t > 2$ remain to be explored.

- The security of unconditionally all-or-nothing transforms and all the generalizations follows directly from their definition. However, if we step back to computational AONTs and consider Rivest's package transform, there is no proof of security provided in the literature. Similarly, the extended package transform, which we introduced in Section 4.5, is also discussed in an informal manner. Formal security proofs for these protocols would be of interest. However, we note that formal proofs of security have been provided for certain instances of AONTs in the computational setting. For example, the security of optimal asymmetric encryption padding (OAEP) [5] as an AONT is studied by Boyko [6].

- This thesis introduced several generalizations of unconditionally secure AONTs, and we focused mainly on the theoretical aspects of the problem. However, this focus on the theory does not mean that the application side of the topic is not significant. To facilitate research on applications of these transforms, it is important and helpful to study and design efficient constructions and implementations of these structures for parameter sets that cover real-life requirements.

- Wang et al. [50] showed that, for prime power values of q, $(2, \phi(q), q)$-AONTs exist, where $\phi(\cdot)$ is the Euler's totient function. However, using a search algorithm, we found $(2, q-1, q)$-AONTs for all prime powers smaller than 16. This observation motivates the question about the existence of $(2, q-1, q)$-AONTs for all prime powers.

- Throughout the thesis, some of the provided results only apply to linear AONTs. However, the validity of these results for the general case, where both linear and non-linear AONTs are concerned, needs to be studied. The tightness of $s \leq v + 1$ bound for $(2, s, v)$-AONTs, Theorem 2.4.2, Lemmas 4.2.13 and 4.2.15, and Corollaries 4.2.14 and 4.2.16 are examples of such results.

- In Chapter 2, we stated that if all the input s-tuples are equiprobable, then an unbiased array is a perfectly secure AONT. The study of the converse of this statement is also an interesting question: If an unbiased array is a perfectly secure AONT, can we say that all input s-tuples must be equally probable?

- In our research, we mostly considered the AONT mapping publicly known and available to the adversary. The security of different generalizations of AONTs under a weaker condition where the adversary does not know about the mapping used can be studied. Wu et al. [51] have used 1-AONTs in this setting.

- In Chapter 2, we used the reduced form of an AONT to distinguish equivalent AONTs. This computational method allowed us to expedite our search. However, an interesting linear algebraic question concerns the characteristics shared between two equivalent linear AONTs. In particular, is there a set of properties of matrices that can uniquely identify a class of equivalent linear AONTs?

Bibliography

[1] Great Internet Mersenne Prime Search. https://www.mersenne.org. Page retrieved October 28, 2020.

[2] Ross Anderson and Eli Biham. Two practical and provably secure block ciphers: Bear and lion. In *Fast Software Encryption,* LNCS volume 1039, pages 113–120. Springer, 1996.

[3] Enrico Bacis, Sabrina De Capitani di Vimercati, Sara Foresti, Stefano Paraboschi, Marco Rosa, and Pierangela Samarati. Mix&slice: Efficient access revocation in the cloud. In *Computer and Communications Security*, pages 217–228. ACM, 2016.

[4] Marco Baldi, Nicola Maturo, Eugenio Montali, and Franco Chiaraluce. AONT-LT: a data protection scheme for cloud and cooperative storage systems. In *International Conference on High Performance Computing & Simulation (HPCS)*, pages 566–571. IEEE, 2014.

[5] Mihir Bellare and Phillip Rogaway. Optimal asymmetric encryption. In *Advances in Cryptology – EUROCRYPT' 94,* LNCS volume 950, pages 92–111. Springer, 1994.

[6] Victor Boyko. On the security properties of OAEP as an all-or-nothing transform. In *Advances in Cryptology – CRYPTO' 99,* LNCS volume 1666, pages 503–518. Springer, 1999.

[7] Ernest F. Brickell. Some Ideal Secret Sharing Schemes. In *Advances in Cryptology — EUROCRYPT '89,* LNCS volume 1666, pages 468–475. Springer, 1990.

[8] Ran Canetti, Yevgeniy Dodis, Shai Halevi, Eyal Kushilevitz, and Amit Sahai. Exposure-resilient functions and all-or-nothing transforms. In *Advances in Cryptology – EUROCRYPT 2000,* LNCS volume 1807, pages 453–469. Springer, 2000.

[9] Roberto G. Cascella, Zhen Caoy, Mario Gerlay, Bruno Crispo, and Roberto Battiti. Weak data secrecy via obfuscation in network coding based content distribution. *Wireless Days, 1st IFIP*, pages 1–5, 2008.

[10] Liqun Chen, Thalia M. Laing, and Keith M. Martin. Revisiting and extending the AONT-RS scheme: a robust computationally secure secret sharing scheme. In *Progress in Cryptology – AFRICACRYPT 2017,* LNCS volume 10239, pages 40–57. Springer, 2017.

[11] Yi-Ruei Chen, Cheng-Kang Chu, Wen-Guey Tzeng, and Jianying Zhou. CloudHKA: A cryptographic approach for hierarchical access control in cloud computing. In *Applied Cryptography and Network Security,* LNCS volume 7954, pages 37–52. Springer, 2013.

[12] Yong Cheng, Zhi-ying Wang, Jun Ma, Jiang-jiang Wu, Song-zhu Mei, and Jiang-chun Ren. Efficient revocation in ciphertext-policy attribute-based encryption based cryptographic cloud storage. *Journal of Zhejiang University SCIENCE C*, 14(2):85–97, 2013.

[13] Charles J. Colbourn and Jeffrey H. Dinitz. *Handbook of Combinatorial Designs, Second Edition*. Chapman & Hall/CRC, 2006.

[14] Paolo D'Arco, Navid Nasr Esfahani, and Douglas R. Stinson. All or nothing at all. *The Electronic Journal of Combinatorics*, 23(4):Paper P4.10, 2016.

[15] Whitfield Diffie and Susan Landau. The export of cryptography in the 20th century and the 21st. *Sun Microsystems Laboratories The First Ten Years*, 2001.

[16] Yevgeniy Dodis, Amit Sahai, and Adam Smith. On perfect and adaptive security in exposure-resilient cryptography. In *Advances in Cryptology – EUROCRYPT 2001,* LNCS volume 2045, pages 301–324. Springer, 2001.

[17] Qin Guo, Mingxing Luo, Lixiang Li, and Yixian Yang. Secure network coding against wiretapping and byzantine attacks. *EURASIP Journal on Wireless Communications and Networking*, 2010(1):1–9, 2010.

[18] Godfrey H. Hardy, John E. Littlewood, et al. Some problems of 'Partitio numerorum'; III: On the expression of a number as a sum of primes. *Acta Mathematica*, 44:1–70, 1923.

[19] Wen-Ai Jackson and Keith M. Martin. A combinatorial interpretation of ramp schemes. *Australasian Journal of Combinatorics*, 14:51–60, 1996.

[20] Katarzyna Kapusta and Gerard Memmi. Selective all-or-nothing transform: Protecting outsourced data against key exposure. In *Cyberspace Safety and Security,* LNCS 11161, pages 181–193. Springer, 2018.

[21] Ghassan O. Karame, Claudio Soriente, Krzysztof Lichota, and Srdjan Capkun. Securing cloud data under key exposure. *IEEE Transactions on Cloud Computing*, (1):1–1, 2017.

[22] S. A. Katre and A. R. Rajwade. Resolution of the sign ambiguity in the determination of the cyclotomic numbers of order 4 and the corresponding Jacobsthal sum. *Mathematica Scandinavica*, 60:52–62, 1987.

[23] Hugo Krawczyk. Secret sharing made short. In *Advances in Cryptology – CRYPTO' 93,* LNCS volume 773, pages 136–146. Springer, 1993.

[24] Loukas Lazos and Marwan Krunz. Selective jamming/dropping insider attacks in wireless mesh networks. *IEEE Network*, 25(1):30–34, 2011.

[25] Vladimir Levenshtein. Split orthogonal arrays and maximum independent resilient systems of functions. *Designs, Codes and Cryptography*, 12:131–160, 1997.

[26] Michael Luby and Charles Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM Journal on Computing*, 17(2):373–386, 1988.

[27] Florence Jessie MacWilliams and Neil James Alexander Sloane. *The Theory of Error-Correcting Codes*. Elsevier, 1977.

[28] Robert J. McEliece and Dilip V. Sarwate. On sharing secrets and Reed-Solomon codes. *Communications of the ACM*, 24(9):583–584, 1981.

[29] Navid Nasr Esfahani, Ian Goldberg, and Douglas R. Stinson. Some results on the existence of t-all-or-nothing transforms over arbitrary alphabets. *IEEE Transactions on Information Theory*, 64(4):3136–3143, 2017.

[30] Navid Nasr Esfahani and Douglas R. Stinson. A list of close to AONT matrices found by computer search. *Technical Report*. Cheriton School of Computer Science, University of Waterloo, `http://cacr.uwaterloo.ca/techreports/2016/cacr2016-08.pdf`, 2016.

[31] Navid Nasr Esfahani and Douglas R. Stinson. Computational results on invertible matrices with the maximum number of invertible $2\times$ 2 submatrices. *Australasian Journal of Combinatorics*, 69(1):130–144, 2017.

[32] Paulo F. Oliveira, Luísa Lima, Tiago TV Vinhoza, João Barros, and Muriel Médard. Coding for trusted storage in untrusted networks. *IEEE Transactions on Information Forensics and Security*, 7(6):1890–1899, 2012.

[33] Hai Pham, Rainer Steinwandt, and Adriana Suárez Corona. Integrating classical preprocessing into an optical encryption scheme. *Entropy*, 21(9):872, 2019.

[34] Alejandro Proano and Loukas Lazos. Selective jamming attacks in wireless networks. In *IEEE International Conference on Communications*, pages 1–6. IEEE, 2010.

[35] Alejandro Proano and Loukas Lazos. Packet-hiding methods for preventing selective jamming attacks. *IEEE Transactions on Dependable and Secure Computing*, 9(1):101–114, 2011.

[36] Michael O. Rabin. Efficient dispersal of information for security, load balancing, and fault tolerance. *Journal of the ACM (JACM)*, 36(2):335–348, 1989.

[37] Jason K. Resch and James S. Plank. AONT-RS: blending security and performance in dispersed storage systems. In *the 9th USENIX Conference on File and Storage Technologies*, pages 191–202. USENIX Association, 2011.

[38] Ronald L. Rivest. All-or-nothing encryption and the package transform. In *Fast Software Encryption,* LNCS volume 1267, pages 210–218. Springer, 1997.

[39] Masoumeh Shafieinejad and Navid Nasr Esfahani. A scalable post-quantum hash-based group signature. *Designs, Codes and Cryptography.* (under review).

[40] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, November 1979.

[41] Douglas R. Stinson. Resilient functions and large sets of orthogonal arrays. *Congressus Numerantium*, 92:105–110, 1993.

[42] Douglas R. Stinson. Something about all or nothing (transforms). *Designs, Codes and Cryptography*, 22(2):133–138, 2001.

[43] Douglas R. Stinson. *Combinatorial Designs: Constructions and Analysis.* Springer, 2003.

[44] Douglas R. Stinson. Ideal ramp schemes and related combinatorial objects. *Discrete Mathematics*, 341(2):299–307, 2018.

[45] Douglas R. Stinson and James L. Massey. An infinite class of counterexamples to a conjecture concerning nonlinear resilient functions. *Journal of Cryptology*, 8(3):167–173, 1995.

[46] Mohit Tawarmalani and Nikolaos V. Sahinidis. A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming*, 103(2):225–249, 2005.

[47] Scott A. Vanstone and Paul C. Van Oorschot. *An Introduction to Error Correcting Codes with Applications*. Springer, 2013.

[48] Rangarajan Vasudevan, Ajith Abraham, and Sugata Sanyal. A novel scheme for secured data transfer over computer networks. *Journal of Universal Computer Science*, 11(1):104–121, 2005.

[49] Stephen A. Vavasis. *Nonlinear Optimization: Complexity Issues*. Oxford University Press, Inc., 1991.

[50] Xin Wang, Jie Cui, and Lijun Ji. Linear $(2, p, p)$-AONTs exist for all primes p. *Designs, Codes and Cryptography*, 87(10):2185–2197, 2019.

[51] Danye Wu, Zhiwei Xu, Bo Chen, and Yujun Zhang. Towards access control for network coding-based named data networking. In *IEEE Global Communications Conference – GLOBECOM 2017*, pages 1–6. IEEE, 2017.

[52] Qin Zhang and Loukas Lazos. Collusion-resistant query anonymization for location-based services. In *IEEE International Conference on Communications*, pages 768–774. IEEE, 2014.

[53] Yiwei Zhang, Tao Zhang, Xin Wang, and Gennian Ge. Invertible binary matrices with maximum number of 2-by-2 invertible submatrices. *Discrete Mathematics*, 340(2):201–208, 2017.