STUDIES ON ALMOST PERFECT NONLINEAR FUNCTIONS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF APPLIED MATHEMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

GÜNEŞ BATMAZ KESKİN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
CRYPTOGRAPHY

SEPTEMBER 2022

Approval of the thesis:

STUDIES ON ALMOST PERFECT NONLINEAR FUNCTIONS

submitted by **GÜNEŞ BATMAZ KESKİN** in partial fulfillment of the requirements for the degree of **Master of Science in Cryptography Department, Middle East Technical University** by,

Prof. Dr. Sevtap Selçuk Kestel _____
Dean, Graduate School of **Applied Mathematics**

Assoc. Prof. Dr. Oğuz Yayla _____
Head of Department, **Cryptography**

Prof. Dr. Ferruh Özbudak _____
Supervisor, **Mathematics Department, METU**

Examining Committee Members:

Assoc. Prof. Dr. Ali Doğanaksoy
Mathematics Department, METU _____

Prof. Dr. Ferruh Özbudak
Mathematics Department, METU _____

Assoc. Prof. Dr. Burcu Gülmez Temür
Mathematics Department, Atılım University _____

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: GÜNEŞ BATMAZ KESKİN

Signature :

ABSTRACT

STUDIES ON ALMOST PERFECT NONLINEAR FUNCTIONS

KESKİN, GÜNEŞ BATMAZ
M.S., Department of Cryptography
Supervisor : Prof. Dr. Ferruh Özbudak

September 2022, 29 pages

This thesis presents some parts of our continuing work on Almost Perfect Nonlinear(APN) Functions. Through this thesis, first of all the significance of APN functions and couple of generation methods for APN Functions are presented. Some methods are considered in a deeper context. Finally various algorithms to obtain functions related APN functions using Python programming language have implemented and some results are analyzed.

Keywords: APN Functions, DFS, BFS

ÖZ

NEREDEYSE MÜKEMMEL DOĞRUSAL OLMAYAN FONKSİYONLAR ÜZERİNE ÇALIŞMALAR

KESKİN, GÜNEŞ BATMAZ
Yüksek Lisans, Kriptografi Bölümü
Tez Yöneticisi : Prof. Dr. Ferruh Özbudak

Eylül 2022, 29 sayfa

Bu tez, Neredeyse Mükemmel Doğrusal Olmayan (APN) Fonksiyonlar üzerine devam eden çalışmalarımızın bazı bölümlerini sunmaktadır. Bu tez ile öncelikle APN Fonksiyonlarının önemi ve APN Fonksiyonları için birkaç üretim yöntemi sunulmuştur. Bazı yöntemler daha derin bir bağlamda ele alınmıştır. Son olarak, Python programlama dilini kullanarak APN fonksiyonları elde etmek için çeşitli algoritmalar uygulanmış ve bazı sonuçlar analiz edilmiştir.

Anahtar Kelimeler: APN Fonksiyonları, DFS, BFS

To my dear parents, sister, and to my loving husband

ACKNOWLEDGMENTS

I would like to express my appreciation and gratitude towards my thesis supervisor Prof. Dr. Ferruh Özbudak for his patience and guidance during this thesis. His encouragements guide me through difficult times and pushed me towards to fulfilling this requirements.

I am thankful to my thesis defence committee for taking their valuable time for this thesis.

I would like to express my deepest appreciations to my parents and my sister who supported me through thick and thin, believing in me when I did not believe in myself and standing behind me, and to my loving husband for knowing my struggles, helping me through difficult times to accomplish my hearts desires.

Finally, I would like to thank The Scientific and Technological Research Council of Turkey (TÜBİTAK) for financial support under the program 2210-A.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

S-Box	Substitution Box
APN	Almost Perfect Nonlinear
BFS	Breadth First Search
DFS	Depth First Search
R-DFS	Randomized Depth First Search
SQL	Structured Query Language

CHAPTER 1

INTRODUCTION

In modern life, need for enhancement of data encryption is rising rapidly. This creates a higher demand over search for a new encryption method. Currently, various cryptographic algorithms, including Symmetric Key Cryptography are based of of S-Box systems. Key to generating new S-Boxes is to finding the right and new combinations of Linear and Nonlinear Functions. This need for new functions introduces heated search for new APN functions.

Organisation of this thesis is as follows:

- In Chapter 2, literature review of APN generation functions are given.
- In Chapter 3, a new method introduced by Sălăgean and Özbudak explained.
- In Chapter 4, different implementation of a new generation method for Quadratic APN Functions is explored and some results are shared.
- In Chapter 5, conclusion of the thesis given.

CHAPTER 2

A NOTION OF APN FUNCTION GENERATION

In this chapter, basic definition of APN functions is given, then continued with a useful resource gathered and updated regularly in the Boolean functions is introduced and briefly introduced various work done on APN generation.

2.1 Almost Perfect Nonlinear (APN) Functions

In this part, basic definition of APN function is given.

Definition 1. Almost Perfect Nonlinear(APN) Functions is defined as a mapping from $GF(p^n)$ to $GF(p^n)$ if each equation

$$F(t+a) - F(t) = b, a \in GF(p^n)^*, b \in GF(p^n)$$

has at most two solutions for $t \in GF(p^n)$ [9].

This property of APN functions provides an optimal differential attacks which makes these functions more resistant towards different attacks.

2.2 Boole Web Page

In this section, I will go through the contents of Boole web page. In this web page, experts on the field aims to provide an overview over topics for the researchers in Boolean Functions and Cryptography [2].

2.2.1 Contents

This section will include an overview of its content.

- **People in Boolean Functions**: Includes the list of all notable people working on the field
- **Books on Boolean Functions**: Includes the list of recommended books on the field.
- **Papers on Boolean Functions**: Includes a selection of foundational papers on the field
- **Conferences related to Boolean Functions**: Includes noteworthy conferences on the field
- **Tables**: Includes tables of known instances of APN functions over F_{2^n} and some other differential uniformities.
- **Magma Code**: Includes files containing the implementations of different functions related field implemented with Magma Programming Language. These codes include fundamental calculations such as Algebraic Normal Form(ANF),

APN property, CCZ-Equivalences, Gamma-rank, Delta Rank and Multiplier Group for vectorial Boolean Functions.

- **Notation**: Includes the notations used in the web page.
- **Algorithms for testing equivalence**: Includes couple of algorithms to test equivalence.

In addition to previously mentioned content, this site contains a summary of the field. This summary includes the following headlines.

- Boolean Functions
  - Bent Functions
- Vectorial Boolean Functions
  - Bent Functions
  - Almost Bent Functions
  - Almost Perfect Nonlinear (APN) Functions
    - APN Permutations
  - Plateaued Functions
  - Equivalence Relations
  - Nonlinearity
- Commutative Presemifields and Semifields

2.2.2 Significance of Boole Web Page on APN Research

This web page contains the summary of general information about Boolean Functions. Information gathered is properly cited, properly structured with a user friendly manner, and gathered by people working on the field. Also, it is presented in a simple manner to enhance the understanding of the reader. In addition to that, this web page presents information that is hard to gather in one place, such as known APN functions over different fields, and includes the generation methods and even Magma Codes, if accessible. To summarize, this web page enables researchers work on further level, rather than the very beginning.

2.3 Generating APN Functions

In this section, I will go through couple of different APN generation methods. First generation method is to use the known infinite APN power functions over $GF(2^n)$, some others tried generating new APN functions from the known ones, and many others tried to find new APN functions using other informed search methods.

2.3.1 Known infinite families for APN functions in $GF(2^n)$

There are known infinite families of APN functions of the term $F(x) = x^d$. These families implies that, when the conditions are satisfied, for any n, there exists an APN function with in this family. In Table 2.1 you may find the known infinite families for APN functions in $GF(2^n)$.

Table 2.1: Known infinite families for APN functions in $GF(2^n)$ [2]

Family	Exponent	Conditions	$deg(x^d)$	Reference
Gold	$2i+1$	$gcd(i,n)=1$	2	[13], [16]
Kasami	$2^{2i}-2^i+1$	$gcd(i,n)=1$	$i+1$	[14], [15]
Welch	2^t+3	$n=2t+1$	3	[11]
Niho	$2^t+2^{t/2}-1, teven$ $2^t+2^{(3t+1)/2}-1, todd$	$n=2t+1$	$(t+2)/2$ $t+1$	[10]
Inverse	$2^{2t}-1$	$n=2t+1$	$n-1$	[16], [5]
Dobbertin	$2^{4i}+2^{3i}+2^{2i}+2^i-1$	$n=5i$	$i+3$	[12]

2.3.2 Other Work Done on APN Generation

To grasp the concept of generating new APN functions, we made a literature review through the papers that have been focused on generating new APN functions. Initially, we started with Beierle and Leander's "New Instances of Quadratic APN Functions" [4]. In this paper, they presented a recursive tree search to find new APN permutations. Their search generated more than 10,000 permutations and 35 new quadratic APN functions in eight dimension.

In this research, basically, build up to their previous work of [3] where they have done exhaustive search to obtain new APN permutations in dimensions of 6,7 and 8 that admits linear self-equivalence. In [3], they could not generate any new APN permutations. After this search they enhanced their search by initialized a look-up table in the beginning and each time a new entry is fix the program checks whether it is a candidate for an APN.

Another approach to generate new APN Function is explained in "Constructing new APN functions from known ones" by Budaghyan et. al. [6]. As it is self-explanatory from the name, Budaghyan et. al., they applied their algorithm given in [6], Chapter 3 on Gold power functions, given in Table 2.1, and could contract a function $x^3+tr(x^9)$

over F_{2^n}.

CHAPTER 3

A DIFFERENT ANGLE FOR QUADRATIC APN GENERATION

In the previous chapters, different workes done on generating new APN Functions are covered. On this chapter, Sălăgean and Özbudak's [17] presentation of a new definition to generate Quadratic APN Functions will be covered. In Sălăgean and Özbudak's work, a new function type is introduced as Pre-APN functions. In their work, definition Pre-APN functions is given as follows.

Theorem 3.1 $2 \leqslant m < n$. Assume that $g : \mathbb{F}_{2^m} \to \mathbb{F}_{2^n}$ is a quadratic pre-APN. For $A_1, \ldots, A_m \in \mathbb{F}_{2^n}$ consider the linear map $F : \mathbb{F}_2^m \to \mathbb{F}_{2^n}$ defined as $L(x_1, \ldots, x_m) = A_1 x_1 + \cdots A_m x_m$. Let $f : \mathbb{F}_2^{m+1} = F_2^m \times F_2 :\to \mathbb{F}_{2^n}$ defined using g and L as

$$f(\underbrace{x_1, \ldots, x_m}_{x}, y) = g(x_1, \ldots, x_m) + yL(x_1, \ldots, x_m).$$

Then f is pre- APN (actually APN if $m = n - 1$) iff

$$D_u g|_x + L(u) \neq 0 \text{ for any } u, x \in \mathbb{F}_2^m \text{ with } u \neq 0$$

[17]

As it can be gather from this theorem, for a given $g : \mathbb{F}_{2^m} \to \mathbb{F}_{2^n}$ (quadratic pre-APN), when conditions are met, where $g : \mathbb{F}_{2^{m+1}} \to \mathbb{F}_{2^n}$ is a quadratic pre-APN and can obtain a APN when $m = n - 1$.

The main objective of our work using this theorem is to find a way to show that this method can be converted into an efficient algorithm that will help us to generate f functions such that f is an APN $f : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$.

CHAPTER 4

QUADRATIC APN FUNCTION GENERATION IMPLEMENTATION

In this chapter, we will present the implementation methods that are used to generated our results as well as the issues that we faced through this process will be explained.

4.1 Different Implementation Approaches

As it is mentioned in the previous chapter, this work is based on the work of Sălăgean and Özbudak's [17] work of Pre-APN Functions. To obtain some results, I tried couple of different approaches through the process. Firstly, I started with a Naive, also known as brute force, approach to obtain results. At this point, I was planning to use already in use technologies such as Sage(9.2) and SBoxU library for Python. While using these, I encounter issues that are not easy to overcome. After that, I decided to change my approach and implemented all the required functionalities myself with Python 3.8. After implementing required functions, I could gather my first results. But, as m get greater, the time consumption of the process increment drastically. Because of the time consumption aspect, we needed to re-design our approach and

implement different algorithms to try and obtain results.

Due to the nature of our data, it was easy to project the generation process to a Search Tree to run Breadth First Search through the Search Tree generated with all possible A values, and pruning the ends that will not produce any viable outcomes to enhance the run-time. Even though this approach was viable for generating results up to $n = m = 5$, after that point was time expectancy was not plausible for regular computers. After that point, we changed our approach to Depth First Search (DFS) to be able to reach some results without trcing all through the domain. At this point, we could be able to retrieve some results on $n = m = 6$, we could not retrieve any results for $n = m = 7$. We consider the lack of results might be caused by consecutive execution, and decided to try for a Randomized Depth First Search(R-DFS) Algorithm to generate results. Even though using randomized algorithm drastically speed our process to generate results up to $n = m = 6$, we could not retrieve any results for $n = m = 7$ yet.

4.1.1 Naive Implementation with Existing Libraries

As it is mentioned before, we started with a Naive implementation with existing libraries. Algorithm of naive implementation was straight forward. It was calculating all possible outcomes without considering any data flow. It was using SageMath 9.2 for Boolean Functions and derivations.

Unfortunately, this method was not viable due to a memory leak in the implementation of SageMath, which is a known bug [1], I encountered with memory overload, which causes the computer to crash after around 6000-7000 tries. This number was only applicable to calculations to extent $F_{2^2} - > F_{2^3}$ to $F_{2^3} - > F_{2^3}$ since it only

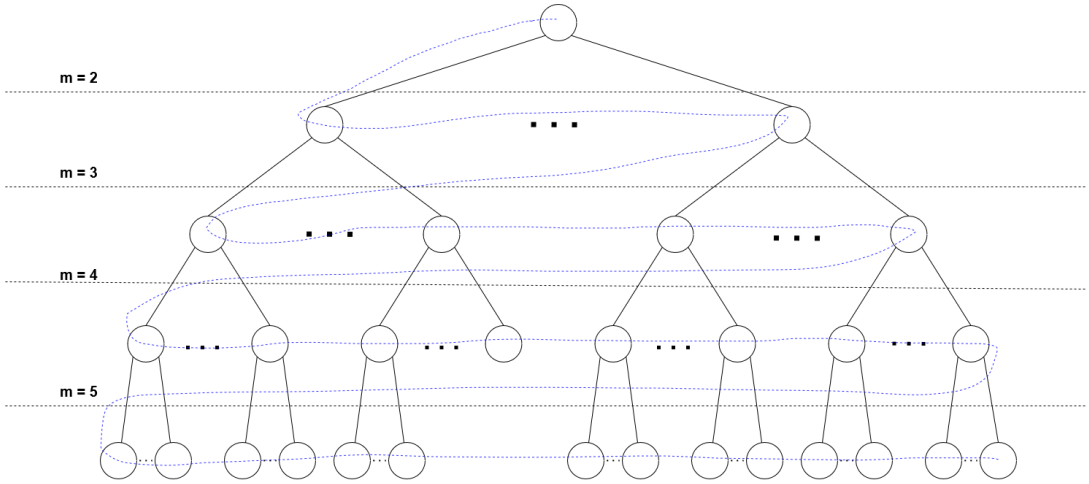

Figure 4.1: Visualization of Breadth First Search Algorithm.

requires 448 tries. But, from $F_{2^2} -> F_{2^4}$ to $F_{2^3} -> F_{2^4}$ which takes 3840 tries to obtain all Pre-APN functions in $F_{2^3} -> F_{2^4}$, and for each selected Pre-APN, we should 3840 tries to extent functions from $F_{2^3} -> F_{2^4}$ to $F_{2^4} -> F_{2^4}$ to obtain APN functions it becomes impossible to continue with SageMath. Because of that, we decided to change our approach and implement Boolean Functions and derivations in Python 3.8 and use those implementations instead from this point on.

4.1.2 Breadth First Search(BFS) Implementation

As explained above, after trying to generate Pre-APN functions through naive implementation, we had a better understanding of the structure. This understanding lead us to implement a more compatible data structure as a base for our study.

We decided to use General Tree Data Structure introduced in [7] since this data structure is more compatible to data that we are working on. Also, we added tree pruning to our implementation from this point on. Even thought pruning is a common practice in tree algorithms, we were inspired to add this to our implementation is Beierle et. al. [4] work on to similar generation methods. In addition to that, we decided to

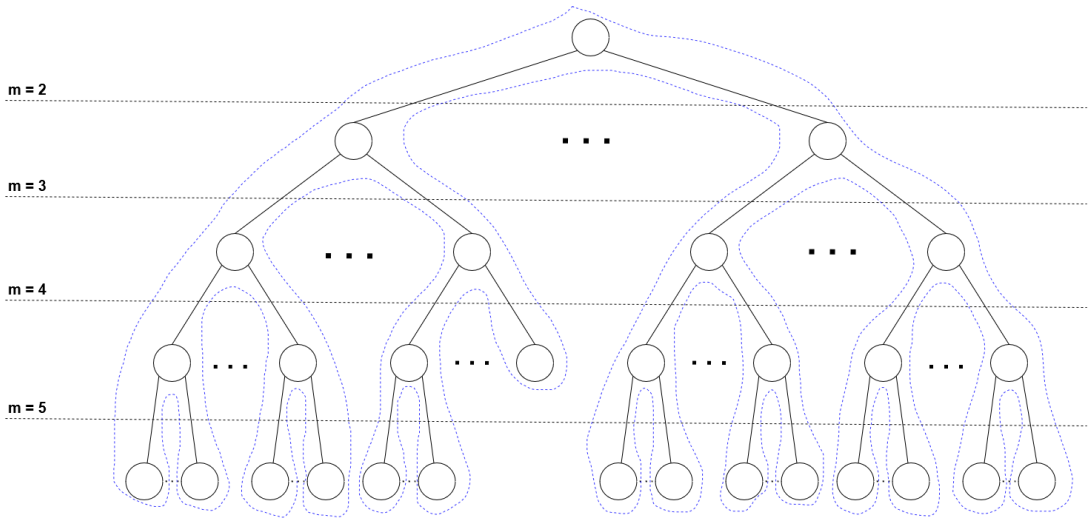

Figure 4.2: Visualization of Depth First Search Algorithm.

make calculations over vectors rather than fields since vector calculations are easier to implement and also are cheaper calculations for computers. Then, we implemented a BFS for our data set. We implemented the BFS algorithm as it is given in [8] with slight changes for it to fit our structure better. Our objective to implement BFS was to generate all possible instances in each level, which would hopefully generate all possible Quadratic APN functions. As us can observed from Figure 4.1, BFS Algorithm traces each level until the level is complete, and move to the next level once the current level is completed.

This implementation did worked as expected and generated all the results in each level. Even thought this seemed like it would provide more than other approach, time constraint of this search becomes infeasible after starting point of $m = 2$ and $n = 5$.

This lead us to change our perspective from generating all possible outcomes to generating some outcomes and working through the results as they are calculated.

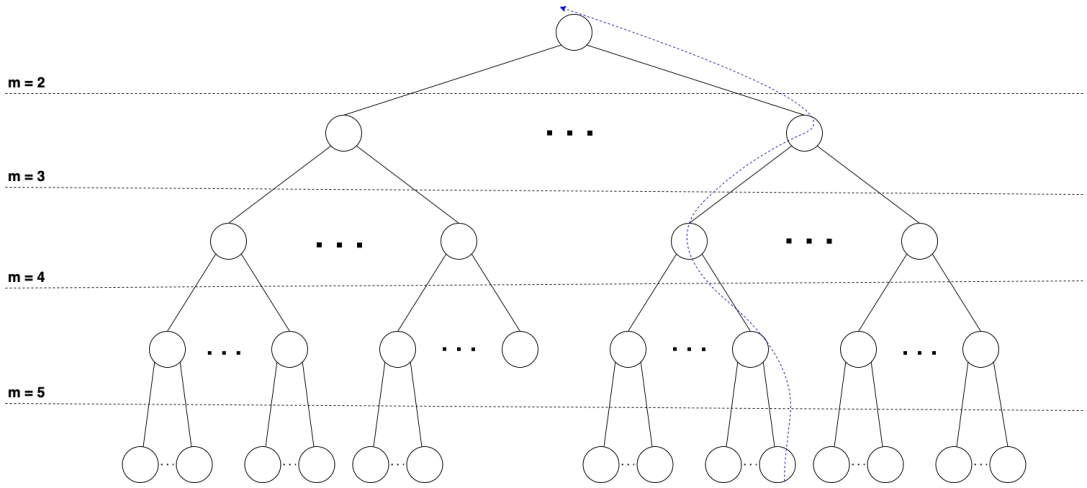

Figure 4.3: Visualization of Randomized Breadth First Search Algorithm.

4.1.3 Depth First Search(DFS) Implementation

As it s explained in the previous section, after $n = 5$, we needed to change my approach to find all of the result and change the objective to find some results in starting from $m = 2$ and ending up with $m = n$, so we can find at least some APN functions. This approach would be more applicable with Depth First Search (DFS) algorithm, so implemented this algorithm as it is given in [8], similer to BFS, we did slight changes in the structures to fit our data better.

Using this approach we could produce couple of results up to $m = n = 6$, which we could not produced in any other method. Unfortunately, even the algorithm run for over 15 days, it could not generate any extensions from $F_{2^6} -> F_{2^7}$ to $F_{2^7} -> F_{2^7}$ even though it produced extensions up to $F_{2^6} -> F_{2^7}$.

4.1.4 Randomized Depth First Search (R-DFS) Implementation

Due to the difficulties we encountered with time constraint, we decided to change the existing DFS algorithms iterative flow with a randomized algorithm. This algorithms

main difference from the existing DFS algorithm is, as it can be observed from its title, rather than a consecutive execution, it would randomly select one of the many available paths to try and find a result.

Using this algorithm, we could be able to generate many solutions extended to $F_{2^6} -> F_{2^6}$ in just a couple of minutes, and any extension less than $n < 6$ can produce results in less than a minute. Unfortunately, even though we run this program for over 10 days, we could not find any solutions that extends from $F_{2^6} -> F_{2^7}$ to $F_{2^7} -> F_{2^7}$, even though we could generated extensions up to $F_{2^6} -> F_{2^7}$ very easily.

4.2 Results

In this section, I will share multiple, randomly selected results which can be seen in the Appendix A. Then, columns in the results will be briefly explained, and a brief summary of the calculations that are made to generate these results.

As you can observe from the results in Appendix A, the result are displayed in columns and these columns contains $g(X)$, $f(X)$ and $A_i s.t. i \in \{0, 1, .., m-1\}$.

The resulting function $f(X)$ is found using $f(X, y) = g(X) + y * L(X)$ where $X = [x_0, x_1, ..., x_m]$, $L(x) = x_0 * A_0 + x_1 * A_1 + .. + x_m * A_m$ from Theorem 3.1. As you can see from the results in Appendix A, vectorial implementations are used. This implementation have been chosen to ease both human readability and and computer implementation.

In Table A.1 you may see randomly selected 50 different extensions from $g(x) : F_{2^2} \rightarrow F_{2^3}$ to $f(x) : F_{2^3} \rightarrow F_{2^3}$. As you can see, since the extension started with

$m = 2$, there are only 2 A_i are used to generate $L(X)$. In this table, you may find different $g(X)$ functions to start with. Since, $m = n-1$, only one extension is enough to reach a potential Quadratic APN functions.

In Table A.2 again, you may see randomly selected 50 different extensions, but this time from $g(x) : F_{2^2} \rightarrow F_{2^4}$ to $f(x) : F_{2^3} \rightarrow F_{2^4}$. Similar to Table A.1, there are only 2 A_i are used, but unlike that, this time the found $f(x)$ functions are $m \neq n$. Then, we need to take another step, do another extension, to reach $m = n$. In Tables A.3, A.4, A.5, A.6, you may find the extensions of 4 different $f(x) : F_{2^3} \rightarrow F_{2^4}$ calculated in the previous step. In those steps, A.2 's $f(x)$ values becomes $g(x)$'s of Tables A.3, A.4, A.5, A.6. In these tables, since $m = 3$, there are 3 A_i values to calculate $L(X)$. The results of Tables A.3, A.4, A.5, A.6 are potential Quadratic APN functions.

Theoretically, as it is given in Theorem 3.1, they are supposed to be Quadratic APN functions. But, no validation process is executed, we prefer to refer them as potential Quadratic APN functions.

CHAPTER 5

CONCLUSION AND FUTURE WORKS

To conclude, in this thesis, we covered through the different APN generations methods including an ongoing work of Sălăgean and Özbudak's recently developed approach of APN generations, and different implementations done to create an application of their ongoing work.

The implementations that are done, although they could not go beyond $F_{2^6} - > F_{2^6}$ for now, we could generate millions of different results, currently stored in a SQL database with size about 8 GB.

From this point on, we are aiming to understand and analyze the results we gathered whilst working towards optimizing our current algorithms.

REFERENCES

[1] Ask sage math: Memory leak somewhere?, 2016 (accessed August 20, 2022). URL: https://ask.sagemath.org/question/34364/memory-leak-somewhere/.

[2] Boolean functions, 2022 (accessed August 20, 2022). URL: https://boolean.h.uib.no/mediawiki/index.php/Main_Page.

[3] C. Beierle, M. Brinkmann, and G. Leander. Linearly self-equivalent apn permutations in small dimension. *IEEE Transactions on Information Theory*, 67(7):4863–4875, 2021. doi:10.1109/TIT.2021.3071533.

[4] C. Beierle and G. Leander. New instances of quadratic apn functions. *IEEE Transactions on Information Theory*, 68(1):670–678, 2022. doi:10.1109/TIT.2021.3120698.

[5] T. Beth and C. Ding. On almost perfect nonlinear permutations. In T. Helleseth, editor, *Advances in Cryptology — EUROCRYPT '93*, pages 65–76, Berlin, Heidelberg, 1994. Springer Berlin Heidelberg.

[6] L. Budaghyan, C. Carlet, and G. Leander. Constructing new apn functions from known ones. *Finite Fields and Their Applications*, 15(2):150–159, 2009. URL: https://www.sciencedirect.com/science/article/pii/S1071579708000622, doi:https://doi.org/10.1016/j.ffa.2008.10.001.

[7] F. Carrano and T. Henry. *Data Abstraction & Problem Solving with C++: Walls and Mirrors*. Pearson, 2013. URL: https://books.google.com.tr/books?id=bEHBuwAACAAJ.

[8] T. Cormen, C. Leiserson, R. Rivest, and C. Stein. *Introduction to Algorithms, third edition*. Computer science. MIT Press, 2009. URL: https://books.google.com.tr/books?id=i-bUBQAAQBAJ.

[9] H. Dobbertin. Almost perfect nonlinear power functions on gf(2n): The niho case. *Information and Computation*, 151(1):57–72, 1999. URL: https://www.sciencedirect.com/science/article/pii/S089054019892764X, doi:https://doi.org/10.1006/inco.1998.2764.

[10] H. Dobbertin. Almost perfect nonlinear power functions on gf(2n): The niho case. *Information and Computation*, 151(1):57–72, 1999. URL: https://www.sciencedirect.com/science/article/pii/S089054019892764X, doi:https://doi.org/10.1006/inco.1998.2764.

[11] H. Dobbertin. Almost perfect nonlinear power functions on gf(2/sup n/): the welch case. *IEEE Transactions on Information Theory*, 45(4):1271–1275, 1999. doi:10.1109/18.761283.

[12] H. Dobbertin. Almost perfect nonlinear power functions on gf(2n): A new case for n divisible by 5. In D. Jungnickel and H. Niederreiter, editors, *Finite Fields and Applications*, pages 113–121, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[13] R. Gold. Maximal recursive sequences with 3-valued recursive cross-correlation functions (corresp.). *IEEE Transactions on Information Theory*, 14(1):154–156, 1968. doi:10.1109/TIT.1968.1054106.

[14] H. Janwa and R. M. Wilson. Hyperplane sections of fermat varieties in p3 in char. 2 and some applications to cyclic codes. In G. Cohen, T. Mora, and O. Moreno, editors, *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, pages 180–194, Berlin, Heidelberg, 1993. Springer Berlin Heidelberg.

[15] T. Kasami. The weight enumerators for several classes of subcodes of the 2nd order binary reed-muller codes. *Information and Control*, 18(4):369–394, 1971. URL: https://www.sciencedirect.com/science/article/pii/S0019995871904736, doi:https://doi.org/10.1016/S0019-9958(71)90473-6.

[16] K. Nyberg. Differentially uniform mappings for cryptography. In T. Helleseth, editor, *Advances in Cryptology — EUROCRYPT '93*, pages 55–64, Berlin, Heidelberg, 1994. Springer Berlin Heidelberg.

[17] A. Sălăgean and F. Özbudak. Pre-apn, pre-pn, contructions and related results. preprint, 2020.

APPENDIX A

SOME RESULTS

Here is presented various results of the calculations done in this thesis. These results are as follows:

- Randomly selected 50 results of extension from $g(X) : F_{2^2} \rightarrow F_{2^3}$ to $f(X) : F_{2^3} \rightarrow F_{2^3}$ in Table A.1
- Randomly selected 50 results of extension from $g(x) : F_{2^2} \rightarrow F_{2^4}$ to $f(x) : F_{2^3} \rightarrow F_{2^4}$ in Table A.2
- From a single $g(X) = [0, x_1 * x_2, x_0 * x_2, x_0 * x_1]$, randomly selected 50 results of extension from $g(X) : F_{2^3} \rightarrow F_{2^4}$ to $f(C) : F_{2^4} \rightarrow F_{2^4}$ in Table A.3
- From a single $g(X) = [x_1 * x_2, x_1 * x_2, x_0 * x_2, x_0 * x_1]$, randomly selected 50 results of extension from $g(X) : F_{2^3} \rightarrow F_{2^4}$ to $f(C) : F_{2^4} \rightarrow F_{2^4}$ in Table A.4
- From a single $g(X) = [x_0 * x_2 + x_1 * x_2, x_1 * x_2, 0, x_0 * x_1 + x_0 * x_2]$, randomly selected 50 results of extension from $g(X) : F_{2^3} \rightarrow F_{2^4}$ to $f(x) : F_{2^4} \rightarrow F_{2^4}$ in Table A.5
- From a single $g(X) = [x_0 * x_2, x_1 * x_2, x_0 * x_2, x_0 * x_1]$, randomly selected 50 results of extension from $g(X) : F_{2^3} \rightarrow F_{2^4}$ to $f(X) : F_{2^4} \rightarrow F_{2^4}$ in Table A.6

Table A.1: Randomly selected 50 results of extension from $g(X) : F_{2^2} \rightarrow F_{2^3}$ to $f(X) : F_{2^3} \rightarrow F_{2^3}$

$A1$	$A2$	$g(X)$	$f(X)$
$[1,0,1]$	$[0,1,1]$	$[x_0*x_1, x_0*x_1, x_0*x_1]$	$[x_0*x_1+x_0*x_2, x_0*x_1+x_1*x_2, x_0*x_1+x_0*x_2+x_1*x_2]$
$[1,0,1]$	$[1,0,0]$	$[x_0*x_1, x_0*x_1, x_0*x_1]$	$[x_0*x_1+x_0*x_2+x_1*x_2, x_0*x_1, x_0*x_1+x_0*x_2]$
$[1,0,0]$	$[0,1,0]$	$[0, 0, x_0*x_1]$	$[x_0*x_2, x_1*x_2, x_0*x_1]$
$[0,0,1]$	$[0,1,1]$	$[x_0*x_1, x_0*x_1, x_0*x_1]$	$[x_0*x_1, x_0*x_1+x_1*x_2, x_0*x_1+x_0*x_2+x_1*x_2]$
$[1,1,0]$	$[1,0,1]$	$[0, 0, x_0*x_1]$	$[x_0*x_2+x_1*x_2, x_0*x_2, x_0*x_1+x_1*x_2]$
$[1,0,1]$	$[1,1,1]$	$[0, 0, x_0*x_1]$	$[x_0*x_2+x_1*x_2, x_1*x_2, x_0*x_1+x_0*x_2+x_1*x_2]$
$[0,1,0]$	$[1,1,0]$	$[x_0*x_1, 0, x_0*x_1]$	$[x_0*x_1+x_1*x_2, x_0*x_2+x_1*x_2, x_0*x_1]$
$[1,1,1]$	$[1,0,1]$	$[x_0*x_1, 0, 0]$	$[x_0*x_1+x_0*x_2+x_1*x_2, x_0*x_2, x_0*x_2+x_1*x_2]$
$[0,1,0]$	$[0,0,1]$	$[x_0*x_1, x_0*x_1, x_0*x_1]$	$[x_0*x_1, x_0*x_1+x_0*x_2, x_0*x_1+x_1*x_2]$
$[1,0,0]$	$[0,1,0]$	$[x_0*x_1, x_0*x_1, x_0*x_1]$	$[x_0*x_1+x_0*x_2, x_0*x_1+x_1*x_2, x_0*x_1]$
$[0,1,1]$	$[1,0,1]$	$[0, x_0*x_1, 0]$	$[x_1*x_2, x_0*x_1+x_0*x_2, x_0*x_2+x_1*x_2]$
$[0,0,1]$	$[0,1,0]$	$[x_0*x_1, x_0*x_1, 0]$	$[x_0*x_1, x_0*x_1+x_1*x_2, x_0*x_2]$
$[0,0,1]$	$[1,0,1]$	$[x_0*x_1, x_0*x_1, 0]$	$[x_0*x_1+x_1*x_2, x_0*x_1, x_0*x_2+x_1*x_2]$
$[0,0,1]$	$[1,0,0]$	$[0, x_0*x_1, 0]$	$[x_1*x_2, x_0*x_1, x_0*x_2]$
$[1,0,1]$	$[0,0,1]$	$[0, x_0*x_1, 0]$	$[x_0*x_2, x_0*x_1, x_0*x_2+x_1*x_2]$
$[0,1,0]$	$[1,0,0]$	$[x_0*x_1, 0, x_0*x_1]$	$[x_0*x_1+x_1*x_2, x_0*x_2, x_0*x_1]$
$[1,0,0]$	$[1,1,0]$	$[0, x_0*x_1, x_0*x_1]$	$[x_0*x_2+x_1*x_2, x_0*x_1+x_1*x_2, x_0*x_1]$
$[0,1,0]$	$[1,1,0]$	$[0, x_0*x_1, x_0*x_1]$	$[x_1*x_2, x_0*x_1+x_0*x_2+x_1*x_2, x_0*x_1]$
$[1,0,1]$	$[0,1,0]$	$[0, 0, x_0*x_1]$	$[x_0*x_2, x_1*x_2, x_0*x_1+x_0*x_2]$
$[0,0,1]$	$[1,1,1]$	$[x_0*x_1, 0, x_0*x_1]$	$[x_0*x_1+x_1*x_2, x_1*x_2, x_0*x_1+x_0*x_2+x_1*x_2]$
$[0,0,1]$	$[1,1,1]$	$[0, x_0*x_1, 0]$	$[x_1*x_2, x_0*x_1+x_1*x_2, x_0*x_2+x_1*x_2]$
$[0,0,1]$	$[1,0,0]$	$[0, x_0*x_1, x_0*x_1]$	$[x_1*x_2, x_0*x_1, x_0*x_1+x_0*x_2]$
$[1,0,0]$	$[0,1,1]$	$[0, 0, x_0*x_1]$	$[x_0*x_2, x_1*x_2, x_0*x_1+x_1*x_2]$
$[0,0,1]$	$[1,0,0]$	$[x_0*x_1, x_0*x_1, 0]$	$[x_0*x_1+x_1*x_2, x_0*x_1, x_0*x_2]$
$[0,0,1]$	$[1,1,1]$	$[0, x_0*x_1, x_0*x_1]$	$[x_1*x_2, x_0*x_1+x_1*x_2, x_0*x_1+x_0*x_2+x_1*x_2]$
$[1,0,0]$	$[1,1,0]$	$[0, 0, x_0*x_1]$	$[x_0*x_2+x_1*x_2, x_1*x_2, x_0*x_1]$
$[0,1,1]$	$[1,1,1]$	$[0, x_0*x_1, 0]$	$[x_1*x_2, x_0*x_1+x_0*x_2+x_1*x_2, x_0*x_2+x_1*x_2]$
$[1,1,1]$	$[1,0,0]$	$[0, 0, x_0*x_1]$	$[x_0*x_2+x_1*x_2, x_0*x_2, x_0*x_1+x_0*x_2]$
$[0,1,1]$	$[0,0,1]$	$[x_0*x_1, x_0*x_1, x_0*x_1]$	$[x_0*x_1, x_0*x_1+x_0*x_2, x_0*x_1+x_0*x_2+x_1*x_2]$
$[1,0,0]$	$[0,1,1]$	$[0, x_0*x_1, 0]$	$[x_0*x_2, x_0*x_1+x_1*x_2, x_1*x_2]$
$[1,0,1]$	$[1,0,0]$	$[0, x_0*x_1, 0]$	$[x_0*x_2+x_1*x_2, x_0*x_1, x_0*x_2]$
$[1,1,0]$	$[1,1,1]$	$[x_0*x_1, 0, 0]$	$[x_0*x_1+x_0*x_2+x_1*x_2, x_0*x_2+x_1*x_2, x_1*x_2]$
$[0,0,1]$	$[0,1,1]$	$[x_0*x_1, x_0*x_1, 0]$	$[x_0*x_1, x_0*x_1+x_1*x_2, x_0*x_2+x_1*x_2]$
$[0,1,1]$	$[0,0,1]$	$[x_0*x_1, 0, 0]$	$[x_0*x_1, x_0*x_2, x_0*x_2+x_1*x_2]$
$[1,1,0]$	$[0,1,0]$	$[x_0*x_1, x_0*x_1, x_0*x_1]$	$[x_0*x_1+x_0*x_2, x_0*x_1+x_0*x_2+x_1*x_2, x_0*x_1]$
$[0,0,1]$	$[0,1,0]$	$[x_0*x_1, x_0*x_1, x_0*x_1]$	$[x_0*x_1, x_0*x_1+x_1*x_2, x_0*x_1+x_0*x_2]$
$[0,1,0]$	$[1,1,0]$	$[0, 0, x_0*x_1]$	$[x_1*x_2, x_0*x_2+x_1*x_2, x_0*x_1]$
$[1,1,1]$	$[0,1,0]$	$[0, x_0*x_1, x_0*x_1]$	$[x_0*x_2, x_0*x_1+x_0*x_2+x_1*x_2, x_0*x_1+x_0*x_2]$
$[1,0,0]$	$[1,1,1]$	$[x_0*x_1, 0, x_0*x_1]$	$[x_0*x_1+x_0*x_2+x_1*x_2, x_1*x_2, x_0*x_1+x_1*x_2]$
$[1,1,1]$	$[0,1,1]$	$[x_0*x_1, 0, x_0*x_1]$	$[x_0*x_1+x_0*x_2, x_0*x_2+x_1*x_2, x_0*x_1+x_0*x_2+x_1*x_2]$
$[1,1,0]$	$[0,1,1]$	$[x_0*x_1, x_0*x_1, x_0*x_1]$	$[x_0*x_1+x_0*x_2, x_0*x_1+x_0*x_2+x_1*x_2, x_0*x_1+x_1*x_2]$
$[1,0,0]$	$[1,1,0]$	$[x_0*x_1, 0, x_0*x_1]$	$[x_0*x_1+x_0*x_2+x_1*x_2, x_1*x_2, x_0*x_1]$
$[0,1,1]$	$[0,0,1]$	$[x_0*x_1, 0, x_0*x_1]$	$[x_0*x_1, x_0*x_2, x_0*x_1+x_0*x_2+x_1*x_2]$
$[1,1,0]$	$[0,0,1]$	$[0, x_0*x_1, x_0*x_1]$	$[x_0*x_2, x_0*x_1+x_0*x_2, x_0*x_1+x_1*x_2]$
$[0,0,1]$	$[1,0,1]$	$[x_0*x_1, x_0*x_1, x_0*x_1]$	$[x_0*x_1+x_1*x_2, x_0*x_1, x_0*x_1+x_0*x_2+x_1*x_2]$
$[1,0,1]$	$[1,0,0]$	$[0, x_0*x_1, x_0*x_1]$	$[x_0*x_2+x_1*x_2, x_0*x_1, x_0*x_1+x_0*x_2]$
$[1,1,0]$	$[0,1,1]$	$[0, x_0*x_1, 0]$	$[x_0*x_2, x_0*x_1+x_0*x_2+x_1*x_2, x_1*x_2]$
$[1,1,0]$	$[0,1,0]$	$[x_0*x_1, 0, x_0*x_1]$	$[x_0*x_1+x_0*x_2, x_0*x_2+x_1*x_2, x_0*x_1]$
$[1,1,1]$	$[1,0,0]$	$[0, x_0*x_1, 0]$	$[x_0*x_2+x_1*x_2, x_0*x_1+x_0*x_2, x_0*x_2]$
$[1,0,1]$	$[0,1,0]$	$[x_0*x_1, x_0*x_1, 0]$	$[x_0*x_1+x_0*x_2, x_0*x_1+x_1*x_2, x_0*x_2]$

Table A.2: Randomly selected 50 results of extension from $g(x) : F_{2^2} \rightarrow F_{2^4}$ to $f(x) : F_{2^3} \rightarrow F_{2^4}$

$g(X)$	$f(X)$	$A1$	$A2$
$[x_0 * x_1, x_0 * x_1, 0, 0]$	$[x_0 * x_1, x_0 * x_1 + x_1 * x_2, x_0 * x_2 + x_1 * x_2, 0]$	$[0, 1, 0]$	$[1, 1, 0]$
$[x_0 * x_1, 0, x_0 * x_1, 0]$	$[x_0 * x_1, x_1 * x_2, x_0 * x_1 + x_1 * x_2, x_0 * x_2 + x_1 * x_2]$	$[0, 0, 1]$	$[1, 1, 1]$
$[x_0 * x_1, x_0 * x_1, 0, 0]$	$[x_0 * x_1, x_0 * x_1 + x_0 * x_2 + x_1 * x_2, x_0 * x_2 + x_1 * x_2, x_1 * x_2]$	$[1, 1, 0]$	$[1, 1, 1]$
$[x_0 * x_1, 0, 0, x_0 * x_1]$	$[x_0 * x_1 + x_1 * x_2, x_0 * x_2, x_1 * x_2, x_0 * x_1 + x_0 * x_2]$	$[1, 0, 1]$	$[1, 1, 1]$
$[0, x_0 * x_1, x_0 * x_1, x_0 * x_1]$	$[x_0 * x_2 + x_1 * x_2, x_0 * x_1 + x_1 * x_2, x_0 * x_1 + x_0 * x_2 + x_1 * x_2, x_0 * x_1 + x_1 * x_2]$	$[1, 1, 1]$	$[1, 1, 1]$
$[x_0 * x_1, 0, x_0 * x_1, x_0 * x_1]$	$[x_0 * x_1 + x_0 * x_2, x_0 * x_2, x_0 * x_1 + x_1 * x_2, x_0 * x_1 + x_0 * x_2 + x_1 * x_2]$	$[1, 1, 1]$	$[0, 1, 1]$
$[x_0 * x_1, 0, x_0 * x_1, 0]$	$[x_0 * x_1 + x_0 * x_2 + x_1 * x_2, x_0 * x_2 + x_1 * x_2, x_0 * x_1 + x_1 * x_2, 0]$	$[1, 1, 1]$	$[1, 1, 1]$
$[0, x_0 * x_1, 0, 0]$	$[x_0 * x_2, x_0 * x_1, 0, x_1 * x_2]$	$[1, 1, 1]$	$[0, 0, 1]$
$[x_0 * x_1, x_0 * x_1, 0, 0]$	$[x_0 * x_1 + x_1 * x_2, x_0 * x_1 + x_0 * x_2 + x_1 * x_2, 0, x_0 * x_2 + x_1 * x_2]$	$[1, 0, 1]$	$[1, 1, 1]$
$[x_0 * x_1, 0, 0, 0]$	$[x_0 * x_1 + x_1 * x_2, 0, x_1 * x_2, x_0 * x_2 + x_1 * x_2]$	$[0, 0, 1]$	$[1, 1, 1]$
$[0, 0, x_0 * x_1, x_0 * x_1]$	$[x_0 * x_2 + x_1 * x_2, x_1 * x_2, x_0 * x_1 + x_0 * x_2 + x_1 * x_2, x_0 * x_1 + x_0 * x_2]$	$[1, 1, 1]$	$[1, 1, 1]$
$[x_0 * x_1, 0, x_0 * x_1, 0]$	$[x_0 * x_1, x_1 * x_2, x_0 * x_1, x_0 * x_2 + x_1 * x_2]$	$[0, 0, 1]$	$[1, 0, 1]$
$[x_0 * x_1, x_0 * x_1, 0, 0]$	$[x_0 * x_1 + x_1 * x_2, x_0 * x_1 + x_0 * x_2 + x_1 * x_2, x_1 * x_2, x_1 * x_2]$	$[1, 0, 0]$	$[1, 1, 1]$
$[x_0 * x_1, 0, x_0 * x_1, 0]$	$[x_0 * x_1 + x_0 * x_2, x_1 * x_2, x_0 * x_1, x_0 * x_2 + x_1 * x_2]$	$[1, 1, 1]$	$[1, 0, 1]$
$[x_0 * x_1, 0, 0, x_0 * x_1]$	$[x_0 * x_1 + x_0 * x_2, 0, x_0 * x_2, x_0 * x_1 + x_1 * x_2]$	$[1, 1, 1]$	$[0, 0, 1]$
$[0, x_0 * x_1, 0, x_0 * x_1]$	$[x_0 * x_2, x_0 * x_1 + x_0 * x_2 + x_1 * x_2, 0, x_0 * x_1 + x_0 * x_2]$	$[1, 1, 1]$	$[1, 0, 0]$
$[0, x_0 * x_1, 0, x_0 * x_1]$	$[x_1 * x_2, x_0 * x_1 + x_0 * x_2 + x_1 * x_2, x_0 * x_2, x_0 * x_1]$	$[1, 1, 0]$	$[1, 1, 1]$
$[x_0 * x_1, 0, 0, 0]$	$[x_0 * x_1 + x_0 * x_2 + x_1 * x_2, 0, x_0 * x_2 + x_1 * x_2, x_1 * x_2]$	$[1, 1, 1]$	$[1, 1, 1]$
$[x_0 * x_1, x_0 * x_1, 0, x_0 * x_1]$	$[x_0 * x_1 + x_0 * x_2, x_0 * x_1 + x_0 * x_2, x_0 * x_2, x_0 * x_1 + x_0 * x_2 + x_1 * x_2]$	$[1, 1, 1]$	$[0, 0, 1]$
$[x_0 * x_1, x_0 * x_1, 0, x_0 * x_1]$	$[x_0 * x_1 + x_1 * x_2, x_0 * x_1, x_0 * x_2 + x_1 * x_2, x_0 * x_1 + x_1 * x_2]$	$[0, 1, 0]$	$[1, 1, 1]$
$[0, 0, x_0 * x_1, 0]$	$[x_1 * x_2, x_0 * x_2 + x_1 * x_2, x_0 * x_1, x_1 * x_2]$	$[1, 0, 0]$	$[1, 1, 1]$
$[x_0 * x_1, x_0 * x_1, x_0 * x_1, 0]$	$[x_0 * x_1 + x_0 * x_2 + x_1 * x_2, x_0 * x_1, x_0 * x_1 + x_0 * x_2, x_0 * x_2 + x_1 * x_2]$	$[1, 1, 1]$	$[1, 1, 1]$
$[x_0 * x_1, 0, 0, 0]$	$[x_0 * x_1 + x_1 * x_2, x_0 * x_2, x_0 * x_2 + x_1 * x_2, x_1 * x_2]$	$[1, 1, 0]$	$[1, 1, 1]$
$[x_0 * x_1, x_0 * x_1, 0, 0]$	$[x_0 * x_1 + x_0 * x_2, x_0 * x_1, x_1 * x_2, x_1 * x_2]$	$[1, 1, 1]$	$[0, 1, 1]$
$[0, x_0 * x_1, x_0 * x_1, 0]$	$[x_0 * x_2, x_0 * x_1, x_0 * x_1 + x_1 * x_2, x_1 * x_2]$	$[1, 1, 1]$	$[0, 1, 1]$
$[0, x_0 * x_1, 0, 0]$	$[x_0 * x_2, x_0 * x_1 + x_0 * x_2 + x_1 * x_2, x_0 * x_2, x_1 * x_2]$	$[1, 1, 1]$	$[1, 0, 1]$
$[x_0 * x_1, 0, 0, 0]$	$[x_0 * x_1, x_0 * x_2, x_1 * x_2, x_0 * x_2]$	$[1, 0, 1]$	$[0, 1, 0]$
$[x_0 * x_1, x_0 * x_1, x_0 * x_1, 0]$	$[x_0 * x_1 + x_1 * x_2, x_0 * x_1 + x_0 * x_2, x_0 * x_1, x_0 * x_2 + x_1 * x_2]$	$[1, 0, 1]$	$[1, 1, 1]$
$[0, x_0 * x_1, 0, x_0 * x_1]$	$[x_0 * x_2, x_0 * x_1, x_1 * x_2, x_0 * x_1]$	$[1, 1, 1]$	$[0, 1, 0]$
$[0, x_0 * x_1, x_0 * x_1, x_0 * x_1]$	$[x_0 * x_2 + x_1 * x_2, x_0 * x_1 + x_0 * x_2, x_0 * x_1 + x_0 * x_2 + x_1 * x_2, x_0 * x_1 + x_1 * x_2]$	$[1, 1, 1]$	$[1, 1, 1]$
$[0, x_0 * x_1, 0, x_0 * x_1]$	$[0, x_0 * x_1 + x_1 * x_2, x_1 * x_2, x_0 * x_1 + x_0 * x_2 + x_1 * x_2]$	$[0, 0, 1]$	$[1, 1, 1]$
$[x_0 * x_1, x_0 * x_1, x_0 * x_1, x_0 * x_1]$	$[x_0 * x_1, x_0 * x_1 + x_0 * x_2, x_0 * x_1 + x_0 * x_2, x_0 * x_1 + x_1 * x_2]$	$[1, 1, 0]$	$[0, 0, 1]$
$[x_0 * x_1, 0, 0, 0]$	$[x_0 * x_1 + x_0 * x_2, 0, x_0 * x_2, x_0 * x_2 + x_1 * x_2]$	$[1, 1, 1]$	$[0, 0, 1]$
$[x_0 * x_1, 0, 0, 0]$	$[x_0 * x_1 + x_0 * x_2, x_1 * x_2, 0, x_0 * x_2 + x_1 * x_2]$	$[1, 1, 1]$	$[1, 0, 1]$
$[0, x_0 * x_1, 0, x_0 * x_1]$	$[x_1 * x_2, x_0 * x_1, x_0 * x_2 + x_1 * x_2, x_0 * x_1 + x_1 * x_2]$	$[0, 1, 0]$	$[1, 1, 1]$
$[x_0 * x_1, x_0 * x_1, 0, 0]$	$[x_0 * x_1 + x_0 * x_2, x_0 * x_1 + x_0 * x_2, x_0 * x_2, x_1 * x_2]$	$[1, 1, 1]$	$[0, 0, 1]$
$[0, x_0 * x_1, 0, 0]$	$[x_1 * x_2, x_0 * x_1 + x_0 * x_2, 0, x_0 * x_2 + x_1 * x_2]$	$[1, 0, 1]$	$[1, 1, 1]$
$[0, x_0 * x_1, x_0 * x_1, x_0 * x_1]$	$[x_1 * x_2, x_0 * x_1, x_0 * x_1 + x_0 * x_2, x_0 * x_1 + x_0 * x_2]$	$[0, 1, 1]$	$[1, 1, 1]$
$[x_0 * x_1, 0, 0, 0]$	$[x_0 * x_1 + x_1 * x_2, x_0 * x_2 + x_1 * x_2, x_1 * x_2, x_0 * x_2]$	$[1, 0, 1]$	$[1, 1, 1]$
$[x_0 * x_1, 0, x_0 * x_1, 0]$	$[x_0 * x_1 + x_0 * x_2 + x_1 * x_2, x_0 * x_2, x_0 * x_1 + x_0 * x_2, x_0 * x_2 + x_1 * x_2]$	$[1, 1, 1]$	$[1, 1, 1]$
$[x_0 * x_1, 0, 0, 0]$	$[x_0 * x_1 + x_0 * x_2 + x_1 * x_2, x_0 * x_2, x_1 * x_2, x_0 * x_2]$	$[1, 1, 1]$	$[1, 1, 1]$
$[x_0 * x_1, 0, 0, x_0 * x_1]$	$[x_0 * x_1 + x_0 * x_2, x_0 * x_2, x_1 * x_2, x_0 * x_1 + x_1 * x_2]$	$[1, 1, 1]$	$[0, 1, 1]$
$[x_0 * x_1, 0, x_0 * x_1, 0]$	$[x_0 * x_1, x_1 * x_2, x_0 * x_1 + x_0 * x_2, x_1 * x_2]$	$[0, 1, 0]$	$[1, 0, 1]$
$[x_0 * x_1, 0, 0, x_0 * x_1]$	$[x_0 * x_1 + x_0 * x_2 + x_1 * x_2, x_1 * x_2, x_1 * x_2, x_0 * x_1]$	$[1, 1, 1]$	$[1, 1, 1]$
$[x_0 * x_1, 0, 0, 0]$	$[x_0 * x_1, x_0 * x_2, x_1 * x_2, x_1 * x_2]$	$[1, 0, 0]$	$[0, 1, 1]$
$[0, 0, x_0 * x_1, 0]$	$[0, x_0 * x_2, x_0 * x_1 + x_1 * x_2, x_1 * x_2]$	$[1, 0, 0]$	$[0, 1, 1]$
$[0, 0, 0, x_0 * x_1]$	$[0, x_0 * x_2, x_0 * x_2 + x_1 * x_2, x_0 * x_1 + x_1 * x_2]$	$[1, 1, 0]$	$[0, 1, 1]$
$[x_0 * x_1, 0, x_0 * x_1, 0]$	$[x_0 * x_1, x_0 * x_2 + x_1 * x_2, x_0 * x_1 + x_0 * x_2, x_0 * x_2 + x_1 * x_2]$	$[1, 1, 1]$	$[1, 0, 1]$
$[x_0 * x_1, 0, 0, x_0 * x_1]$	$[x_0 * x_1, x_1 * x_2, x_0 * x_2, x_0 * x_1 + x_0 * x_2]$	$[0, 1, 1]$	$[1, 0, 0]$
$[x_0 * x_1, x_0 * x_1, x_0 * x_1, x_0 * x_1]$	$[x_0 * x_1 + x_1 * x_2, x_0 * x_1 + x_0 * x_2, x_0 * x_1 + x_0 * x_2 + x_1 * x_2, x_0 * x_1 + x_0 * x_2 + x_1 * x_2]$	$[1, 1, 1]$	$[1, 1, 1]$

Table A.3: From a single $g(X) = [0, x_1 * x_2, x_0 * x_2, x_0 * x_1]$, randomly selected 50 results of extension from $g(X) : F_{2^3} \to F_{2^4}$ to $f(C) : F_{2^4} \to F_{2^4}$

$f(X)$	A_0	A_1	A_2
$[x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_0 * x_3 + x_2 * x_3]$	[0, 1, 1, 1]	[0, 0, 1, 0]	[1, 1, 1, 1]
$[x_0 * x_3, x_1 * x_2 + x_2 * x_3, x_0 * x_2 + x_1 * x_3, x_0 * x_1 + x_1 * x_3 + x_2 * x_3]$	[1, 0, 0, 0]	[0, 0, 1, 1]	[0, 1, 0, 1]
$[x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_1 + x_0 * x_3]$	[0, 1, 1, 1]	[0, 0, 1, 0]	[1, 1, 0, 0]
$[x_1 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_2 + x_0 * x_3, x_0 * x_1 + x_0 * x_3 + x_1 * x_3]$	[0, 1, 1, 1]	[1, 1, 0, 1]	[1, 0, 0, 0]
$[x_1 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_2 + x_0 * x_3, x_0 * x_1 + x_2 * x_3]$	[0, 1, 1, 0]	[1, 1, 0, 0]	[0, 1, 0, 1]
$[x_1 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_1 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3]$	[0, 1, 1, 1]	[1, 0, 0, 1]	[1, 1, 1, 1]
$[x_1 * x_3, x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_2 + x_1 * x_3, x_0 * x_1 + x_2 * x_3]$	[0, 1, 0, 0]	[1, 0, 1, 0]	[0, 1, 0, 1]
$[x_1 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_2, x_0 * x_1 + x_0 * x_3 + x_2 * x_3]$	[0, 1, 0, 1]	[1, 1, 0, 0]	[1, 1, 0, 1]
$[x_0 * x_3 + x_1 * x_3, x_1 * x_2, x_0 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_1 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3]$	[1, 0, 1, 1]	[1, 0, 0, 1]	[0, 0, 1, 1]
$[x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_1 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3]$	[1, 1, 1, 1]	[1, 0, 1, 1]	[1, 0, 0, 1]
$[x_0 * x_3, x_1 * x_2 + x_1 * x_3, x_0 * x_2 + x_1 * x_3, x_0 * x_1 + x_1 * x_3 + x_2 * x_3]$	[1, 0, 0, 0]	[0, 1, 1, 1]	[0, 0, 0, 1]
$[x_0 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_1 + x_0 * x_3 + x_2 * x_3]$	[1, 0, 1, 0]	[0, 0, 1, 1]	[0, 0, 1, 1]
$[x_1 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_1 + x_2 * x_3]$	[0, 1, 1, 0]	[0, 0, 1, 0]	[1, 0, 1, 1]
$[x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_2 * x_3]$	[1, 1, 0, 0]	[1, 1, 1, 0]	[1, 1, 1, 1]
$[x_0 * x_3 + x_1 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_1 + x_1 * x_3 + x_2 * x_3]$	[1, 1, 1, 0]	[1, 1, 0, 1]	[0, 1, 1, 1]
$[x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_0 * x_3 + x_1 * x_3]$	[0, 1, 1, 1]	[0, 0, 1, 1]	[1, 1, 1, 0]
$[x_0 * x_3 + x_1 * x_3, x_1 * x_2, x_0 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_1 + x_2 * x_3]$	[1, 0, 1, 0]	[1, 0, 0, 0]	[0, 0, 1, 1]
$[x_0 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_2 + x_1 * x_3, x_0 * x_1 + x_1 * x_3 + x_2 * x_3]$	[1, 1, 0, 0]	[0, 1, 1, 1]	[0, 1, 0, 1]
$[x_0 * x_3 + x_1 * x_3, x_1 * x_2 + x_1 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3]$	[1, 0, 1, 1]	[1, 1, 1, 1]	[0, 0, 1, 1]
$[x_1 * x_3, x_1 * x_2 + x_0 * x_3, x_0 * x_2 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_0 * x_3 + x_2 * x_3]$	[0, 1, 0, 1]	[1, 0, 1, 0]	[0, 0, 1, 1]
$[x_0 * x_3, x_1 * x_2 + x_2 * x_3, x_0 * x_2 + x_1 * x_3, x_0 * x_1 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3]$	[1, 0, 0, 1]	[0, 0, 1, 1]	[0, 1, 0, 1]
$[x_0 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_0 * x_3 + x_1 * x_3]$	[1, 1, 1, 1]	[0, 0, 1, 1]	[1, 1, 1, 0]
$[x_1 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_2 + x_1 * x_3, x_0 * x_1 + x_1 * x_3 + x_2 * x_3]$	[0, 1, 0, 0]	[1, 1, 1, 1]	[0, 1, 0, 1]
$[x_1 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_2 + x_1 * x_3, x_0 * x_1 + x_0 * x_3 + x_2 * x_3]$	[0, 1, 0, 1]	[1, 1, 1, 0]	[0, 0, 0, 1]
$[x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_2 + x_1 * x_3, x_0 * x_1]$	[0, 1, 0, 0]	[0, 1, 1, 0]	[1, 1, 0, 0]
$[x_0 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_2 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_0 * x_3 + x_2 * x_3]$	[1, 1, 0, 1]	[0, 0, 1, 1]	[0, 0, 1, 1]
$[x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_1 * x_2, x_0 * x_2 + x_0 * x_3, x_0 * x_1 + x_1 * x_3 + x_2 * x_3]$	[1, 0, 1, 1]	[1, 0, 0, 1]	[1, 0, 0, 0]
$[x_0 * x_3 + x_2 * x_3, x_1 * x_2 + x_1 * x_3 + x_2 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_1 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3]$	[1, 0, 1, 0]	[1, 1, 0, 0]	[1, 1, 0, 0]
$[x_0 * x_3 + x_1 * x_3, x_1 * x_2, x_0 * x_2 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_2 * x_3]$	[1, 0, 0, 0]	[1, 0, 1, 0]	[0, 0, 1, 1]
$[x_0 * x_3 + x_1 * x_3, x_1 * x_2 + x_2 * x_3, x_0 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_1 + x_1 * x_3 + x_2 * x_3]$	[1, 0, 1, 0]	[1, 0, 0, 1]	[0, 1, 1, 1]
$[x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_0 * x_3 + x_1 * x_3]$	[1, 1, 1, 1]	[1, 0, 1, 1]	[1, 1, 1, 0]
$[x_0 * x_3 + x_2 * x_3, x_1 * x_2, x_0 * x_2 + x_1 * x_3, x_0 * x_1 + x_2 * x_3]$	[1, 0, 0, 0]	[0, 0, 1, 1]	[1, 0, 0, 1]
$[x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_2 + x_0 * x_3, x_0 * x_1 + x_2 * x_3]$	[1, 1, 0, 1]	[1, 0, 0, 0]	[1, 1, 1, 0]
$[x_1 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_2 + x_0 * x_3, x_0 * x_1 + x_0 * x_3 + x_1 * x_3]$	[0, 1, 1, 1]	[1, 1, 0, 1]	[0, 0, 0, 1]
$[x_0 * x_3 + x_2 * x_3, x_1 * x_2 + x_2 * x_3, x_0 * x_2 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_0 * x_3 + x_2 * x_3]$	[0, 1, 1, 0]	[0, 0, 1, 0]	[1, 1, 1, 1]
$[x_2 * x_3, x_1 * x_2 + x_0 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_0 * x_3]$	[0, 1, 1, 1]	[0, 0, 1, 0]	[1, 0, 1, 0]
$[x_1 * x_3, x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_2, x_0 * x_1 + x_2 * x_3]$	[0, 1, 0, 0]	[1, 0, 0, 0]	[0, 1, 0, 1]
$[x_0 * x_3 + x_2 * x_3, x_1 * x_2, x_0 * x_2 + x_1 * x_3, x_0 * x_1 + x_0 * x_3 + x_1 * x_3]$	[1, 0, 0, 1]	[0, 0, 1, 1]	[1, 0, 0, 0]
$[x_0 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_0 * x_3 + x_1 * x_3]$	[1, 1, 1, 1]	[0, 1, 1, 1]	[1, 1, 0, 0]
$[x_0 * x_3 + x_1 * x_3, x_1 * x_2 + x_2 * x_3, x_0 * x_2 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_2 * x_3]$	[1, 0, 1, 0]	[1, 0, 1, 1]	[0, 1, 1, 1]
$[x_1 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_2 + x_2 * x_3, x_0 * x_1 + x_1 * x_3 + x_2 * x_3]$	[0, 1, 0, 0]	[1, 1, 0, 1]	[0, 1, 1, 1]
$[x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_1]$	[0, 1, 1, 0]	[0, 0, 1, 0]	[1, 1, 1, 0]
$[x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_1 * x_3]$	[1, 1, 1, 0]	[1, 0, 1, 1]	[1, 0, 1, 0]
$[x_0 * x_3 + x_2 * x_3, x_1 * x_2 + x_1 * x_3 + x_2 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_1 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3]$	[1, 0, 1, 1]	[0, 1, 1, 1]	[1, 1, 0, 1]
$[x_0 * x_3 + x_2 * x_3, x_1 * x_2, x_0 * x_2 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_2 * x_3]$	[1, 0, 0, 0]	[0, 0, 1, 0]	[1, 0, 1, 1]
$[x_1 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_2 + x_2 * x_3, x_0 * x_1 + x_2 * x_3]$	[0, 1, 0, 0]	[1, 0, 0, 1]	[1, 1, 1, 1]
$[x_1 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_1 * x_3]$	[0, 1, 1, 0]	[1, 1, 1, 1]	[1, 1, 1, 0]

Table A.4: From a single $g(X) = [x_1 * x_2, x_1 * x_2, x_0 * x_2, x_0 * x_1]$, randomly selected 50 results of extension from $g(X) : F_{2^3} \to F_{2^4}$ to $f(C) : F_{2^4} \to F_{2^4}$

$f(X)$	A_0	A_1	A_2
$[x_1 * x_2, x_1 * x_2 + x_0 * x_3, x_0 * x_2 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_2 * x_3]$	[0, 1, 0, 0]	[0, 0, 1, 0]	[0, 0, 1, 1]
$[x_1 * x_2 + x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_2 + x_1 * x_3, x_0 * x_1 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3]$	[0, 1, 0, 1]	[0, 1, 1, 1]	[1, 0, 0, 1]
$[x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_2 + x_1 * x_3, x_0 * x_1 + x_2 * x_3]$	[1, 1, 0, 0]	[0, 1, 1, 0]	[1, 1, 0, 1]
$[x_1 * x_2 + x_0 * x_3 + x_1 * x_3, x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_2 + x_1 * x_3, x_0 * x_1 + x_1 * x_3 + x_2 * x_3]$	[1, 0, 0, 0]	[1, 0, 1, 1]	[0, 0, 0, 1]
$[x_1 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_2 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_1 * x_3 + x_2 * x_3]$	[1, 1, 0, 0]	[1, 1, 1, 1]	[1, 0, 1, 1]
$[x_1 * x_2 + x_0 * x_3 + x_1 * x_3, x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_2 + x_1 * x_3, x_0 * x_1]$	[1, 1, 0, 0]	[1, 0, 1, 0]	[0, 1, 0, 0]
$[x_1 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3, x_0 * x_2, x_0 * x_1 + x_0 * x_3 + x_2 * x_3]$	[1, 1, 0, 1]	[1, 0, 0, 0]	[1, 0, 0, 1]
$[x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_2 + x_1 * x_3, x_0 * x_1 + x_1 * x_3 + x_2 * x_3]$	[1, 1, 0, 0]	[0, 1, 1, 1]	[1, 1, 0, 1]
$[x_1 * x_2, x_1 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_2 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3]$	[0, 1, 0, 1]	[0, 1, 1, 1]	[0, 0, 1, 1]
$[x_1 * x_2 + x_2 * x_3, x_1 * x_2 + x_0 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_1 * x_3]$	[0, 1, 1, 0]	[0, 0, 1, 1]	[1, 0, 1, 0]
$[x_1 * x_2 + x_1 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_2 * x_3]$	[0, 1, 1, 0]	[1, 1, 1, 0]	[0, 1, 1, 1]
$[x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_1 * x_2, x_0 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_1 + x_2 * x_3]$	[1, 0, 1, 0]	[0, 0, 1, 0]	[1, 0, 0, 1]
$[x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_1]$	[1, 1, 1, 0]	[0, 0, 1, 0]	[1, 0, 0, 0]
$[x_1 * x_2 + x_0 * x_3, x_1 * x_2, x_0 * x_2 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_0 * x_3 + x_2 * x_3]$	[1, 0, 0, 1]	[0, 0, 1, 0]	[0, 0, 1, 1]
$[x_1 * x_2 + x_0 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_1 + x_0 * x_3 + x_2 * x_3]$	[1, 1, 1, 1]	[0, 1, 0, 0]	[0, 0, 1, 1]
$[x_1 * x_2 + x_0 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_1 + x_0 * x_3 + x_2 * x_3]$	[1, 1, 1, 0]	[0, 0, 1, 0]	[1, 0, 0, 1]
$[x_1 * x_2 + x_0 * x_3, x_1 * x_2 + x_1 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_1 + x_0 * x_3 + x_2 * x_3]$	[1, 0, 1, 1]	[0, 1, 1, 0]	[0, 0, 0, 1]
$[x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_1 * x_2 + x_1 * x_3 + x_2 * x_3, x_0 * x_2 + x_2 * x_3, x_0 * x_1 + x_0 * x_3 + x_2 * x_3]$	[1, 0, 0, 0]	[0, 1, 0, 1]	[1, 1, 1, 1]
$[x_1 * x_2 + x_1 * x_3, x_1 * x_2 + x_0 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_1 * x_3 + x_2 * x_3]$	[0, 1, 1, 0]	[1, 0, 1, 1]	[0, 0, 0, 1]
$[x_1 * x_2 + x_0 * x_3 + x_1 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_2 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_2 * x_3]$	[1, 1, 0, 0]	[1, 1, 1, 0]	[0, 1, 1, 1]
$[x_1 * x_2 + x_0 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_1 + x_2 * x_3]$	[1, 1, 1, 0]	[0, 0, 1, 1]	[0, 1, 1, 1]
$[x_1 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_1 * x_3 + x_2 * x_3]$	[1, 1, 1, 0]	[1, 0, 1, 1]	[1, 1, 0, 1]
$[x_1 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3, x_0 * x_2 + x_1 * x_3, x_0 * x_1 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3]$	[1, 1, 0, 1]	[1, 0, 1, 1]	[1, 0, 0, 1]
$[x_1 * x_2 + x_0 * x_3, x_1 * x_2 + x_1 * x_3, x_0 * x_2, x_0 * x_1 + x_0 * x_3 + x_2 * x_3]$	[1, 0, 0, 1]	[0, 1, 0, 0]	[0, 0, 0, 1]
$[x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_2 * x_3]$	[1, 1, 1, 0]	[0, 1, 1, 0]	[1, 1, 1, 1]
$[x_1 * x_2 + x_0 * x_3 + x_1 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_2 + x_1 * x_3, x_0 * x_1 + x_0 * x_3]$	[1, 1, 0, 1]	[1, 1, 1, 0]	[0, 1, 0, 0]
$[x_1 * x_2 + x_0 * x_3 + x_1 * x_3, x_1 * x_2 + x_0 * x_3, x_0 * x_2 + x_0 * x_3, x_0 * x_1 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3]$	[1, 1, 1, 1]	[1, 0, 0, 1]	[0, 0, 0, 1]
$[x_1 * x_2, x_1 * x_2 + x_0 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_2 * x_3]$	[0, 1, 1, 0]	[0, 0, 1, 0]	[0, 0, 1, 1]
$[x_1 * x_2 + x_0 * x_3 + x_1 * x_3, x_1 * x_2 + x_1 * x_3, x_0 * x_2 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_2 * x_3]$	[1, 0, 0, 0]	[1, 1, 1, 0]	[0, 0, 1, 1]
$[x_1 * x_2 + x_0 * x_3 + x_1 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_2 + x_1 * x_3, x_0 * x_1 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3]$	[1, 1, 0, 1]	[1, 1, 1, 1]	[1, 0, 1, 0]
$[x_1 * x_2 + x_2 * x_3, x_1 * x_2 + x_0 * x_3, x_0 * x_2 + x_0 * x_3, x_0 * x_1 + x_1 * x_3]$	[0, 1, 1, 0]	[0, 1, 0, 1]	[1, 0, 0, 0]
$[x_1 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_0 * x_3]$	[1, 0, 1, 0]	[1, 1, 1, 0]	[1, 0, 1, 0]
$[x_1 * x_2 + x_0 * x_3 + x_1 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_2 + x_1 * x_3, x_0 * x_1 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3]$	[1, 1, 0, 1]	[1, 1, 1, 1]	[0, 1, 0, 1]
$[x_1 * x_2 + x_0 * x_3 + x_1 * x_3, x_1 * x_2, x_0 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_1 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3]$	[1, 0, 1, 0]	[0, 0, 1, 0]	[0, 0, 1, 1]
$[x_1 * x_2 + x_0 * x_3 + x_1 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_1 + x_1 * x_3 + x_2 * x_3]$	[1, 1, 0, 0]	[1, 1, 1, 1]	[1, 0, 0, 1]
$[x_1 * x_2 + x_0 * x_3 + x_1 * x_3, x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_2 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_2 * x_3]$	[1, 1, 0, 0]	[1, 0, 1, 0]	[0, 1, 1, 1]
$[x_1 * x_2 + x_1 * x_3, x_1 * x_2 + x_0 * x_3, x_0 * x_2, x_0 * x_1 + x_1 * x_3 + x_2 * x_3]$	[0, 1, 0, 0]	[1, 0, 0, 1]	[0, 0, 0, 1]
$[x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_1 * x_2, x_0 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_1 * x_3 + x_2 * x_3]$	[1, 0, 1, 0]	[0, 0, 1, 1]	[1, 0, 1, 1]
$[x_1 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_1 * x_2 + x_1 * x_3, x_0 * x_2 + x_1 * x_3, x_0 * x_1 + x_2 * x_3]$	[1, 0, 0, 0]	[1, 1, 1, 0]	[1, 0, 0, 1]
$[x_1 * x_2 + x_1 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_1 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3]$	[0, 1, 1, 1]	[1, 0, 1, 1]	[1, 0, 0, 1]
$[x_1 * x_2 + x_1 * x_3, x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_2 + x_2 * x_3, x_0 * x_1 + x_1 * x_3 + x_2 * x_3]$	[0, 1, 0, 0]	[1, 0, 0, 1]	[0, 1, 1, 1]

Table A.5: From a single $g(X) = [x_0 * x_2 + x_1 * x_2, x_1 * x_2, 0, x_0 * x_1 + x_0 * x_2]$, randomly selected 50 results of extension from $g(X) : F_{2^3} \to F_{2^4}$ to $f(x) : F_{2^4} \to F_{2^4}$

$f(X)$	A_0	A_1	A_2
$[x_0 * x_2 + x_1 * x_2 + x_1 * x_3, x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_3 + x_1 * x_3, x_0 * x_1 + x_0 * x_2 + x_0 * x_3]$	$[0, 1, 1, 1]$	$[1, 0, 1, 0]$	$[0, 1, 0, 0]$
$[x_0 * x_2 + x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_0 * x_2 + x_0 * x_3 + x_1 * x_3]$	$[1, 1, 1, 1]$	$[0, 1, 1, 1]$	$[1, 0, 1, 0]$
$[x_0 * x_2 + x_1 * x_2 + x_0 * x_3, x_1 * x_2 + x_0 * x_3, x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_0 * x_2 + x_0 * x_3 + x_1 * x_3]$	$[1, 1, 0, 1]$	$[0, 0, 1, 1]$	$[0, 0, 1, 0]$
$[x_0 * x_2 + x_1 * x_2 + x_1 * x_3, x_1 * x_2 + x_0 * x_3, x_2 * x_3, x_0 * x_1 + x_0 * x_2 + x_0 * x_3]$	$[0, 1, 0, 1]$	$[1, 0, 0, 0]$	$[0, 0, 1, 0]$
$[x_0 * x_2 + x_1 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3, x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_0 * x_2 + x_2 * x_3]$	$[1, 1, 0, 0]$	$[1, 1, 1, 0]$	$[1, 0, 1, 1]$
$[x_0 * x_2 + x_1 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_1 * x_2, x_0 * x_3, x_0 * x_1 + x_0 * x_2 + x_1 * x_3]$	$[1, 0, 1, 0]$	$[1, 0, 0, 1]$	$[1, 0, 0, 0]$
$[x_0 * x_2 + x_1 * x_2 + x_0 * x_3, x_1 * x_2, x_0 * x_3 + x_1 * x_3, x_0 * x_1 + x_0 * x_2 + x_2 * x_3]$	$[1, 0, 1, 0]$	$[0, 0, 1, 0]$	$[0, 0, 0, 1]$
$[x_0 * x_2 + x_1 * x_2 + x_1 * x_3, x_1 * x_2 + x_0 * x_3, x_0 * x_3 + x_1 * x_3, x_0 * x_1 + x_0 * x_2 + x_0 * x_3 + x_2 * x_3]$	$[0, 1, 1, 1]$	$[1, 0, 1, 0]$	$[0, 0, 0, 1]$
$[x_0 * x_2 + x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_1 + x_0 * x_2 + x_0 * x_3]$	$[1, 1, 1, 1]$	$[0, 1, 1, 0]$	$[1, 0, 0, 0]$
$[x_0 * x_2 + x_1 * x_2 + x_0 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_3 + x_1 * x_3, x_0 * x_1 + x_0 * x_2]$	$[1, 1, 0, 0]$	$[0, 1, 1, 0]$	$[0, 1, 0, 0]$
$[x_0 * x_2 + x_1 * x_2, x_1 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_1 * x_3, x_0 * x_1 + x_0 * x_2 + x_0 * x_3 + x_1 * x_3]$	$[0, 1, 0, 1]$	$[0, 1, 1, 1]$	$[0, 1, 0, 0]$
$[x_0 * x_2 + x_1 * x_2 + x_1 * x_3, x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_2 * x_3, x_0 * x_1 + x_0 * x_2 + x_2 * x_3]$	$[0, 1, 0, 0]$	$[1, 0, 0, 0]$	$[0, 1, 1, 1]$
$[x_0 * x_2 + x_1 * x_2 + x_1 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3, x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_0 * x_2 + x_1 * x_3]$	$[0, 1, 0, 0]$	$[1, 1, 1, 1]$	$[1, 0, 1, 0]$
$[x_0 * x_2 + x_1 * x_2 + x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_1 * x_3, x_0 * x_1 + x_0 * x_2 + x_0 * x_3 + x_2 * x_3]$	$[0, 1, 0, 1]$	$[0, 1, 1, 0]$	$[1, 1, 0, 1]$
$[x_0 * x_2 + x_1 * x_2 + x_1 * x_3, x_1 * x_2 + x_2 * x_3, x_0 * x_3 + x_1 * x_3, x_0 * x_1 + x_0 * x_2]$	$[0, 0, 1, 0]$	$[1, 0, 1, 0]$	$[0, 1, 0, 0]$
$[x_0 * x_2 + x_1 * x_2 + x_1 * x_3, x_1 * x_2 + x_1 * x_3 + x_2 * x_3, x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_0 * x_2 + x_0 * x_3 + x_1 * x_3]$	$[0, 1, 1, 1]$	$[1, 1, 1, 1]$	$[0, 0, 1, 0]$
$[x_0 * x_2 + x_1 * x_2 + x_0 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3, x_2 * x_3, x_0 * x_1 + x_0 * x_2 + x_0 * x_3 + x_1 * x_3]$	$[1, 1, 0, 1]$	$[0, 1, 0, 1]$	$[0, 0, 1, 0]$
$[x_0 * x_2 + x_1 * x_2, x_1 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_1 * x_3, x_0 * x_1 + x_0 * x_2 + x_0 * x_3 + x_1 * x_3]$	$[0, 1, 1, 1]$	$[0, 1, 0, 1]$	$[0, 1, 1, 0]$
$[x_0 * x_2 + x_1 * x_2 + x_1 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_3 + x_2 * x_3, x_0 * x_1 + x_0 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3]$	$[0, 1, 1, 1]$	$[1, 0, 0, 1]$	$[1, 1, 1, 1]$
$[x_0 * x_2 + x_1 * x_2 + x_1 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_3 + x_1 * x_3, x_0 * x_1 + x_0 * x_2 + x_0 * x_3 + x_2 * x_3]$	$[0, 1, 0, 1]$	$[1, 0, 1, 0]$	$[1, 1, 0, 1]$
$[x_0 * x_2 + x_1 * x_2 + x_0 * x_3, x_1 * x_2 + x_1 * x_3 + x_2 * x_3, x_0 * x_3 + x_1 * x_3, x_0 * x_1 + x_0 * x_2 + x_1 * x_3]$	$[1, 0, 1, 0]$	$[0, 1, 1, 1]$	$[0, 1, 0, 0]$
$[x_0 * x_2 + x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_3 + x_2 * x_3, x_0 * x_1 + x_0 * x_2 + x_0 * x_3]$	$[1, 1, 1, 1]$	$[0, 1, 0, 0]$	$[1, 0, 1, 0]$
$[x_0 * x_2 + x_1 * x_2 + x_0 * x_3, x_1 * x_2 + x_1 * x_3 + x_2 * x_3, x_0 * x_3 + x_1 * x_3, x_0 * x_1 + x_0 * x_2 + x_1 * x_3]$	$[1, 0, 1, 0]$	$[0, 1, 1, 1]$	$[1, 0, 1, 0]$
$[x_0 * x_2 + x_1 * x_2 + x_1 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_2 * x_3, x_0 * x_1 + x_0 * x_2 + x_1 * x_3]$	$[0, 0, 1, 0]$	$[1, 0, 1, 1]$	$[1, 1, 1, 0]$
$[x_0 * x_2 + x_1 * x_2 + x_0 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3, x_2 * x_3, x_0 * x_1 + x_0 * x_2 + x_0 * x_3 + x_1 * x_3]$	$[1, 1, 0, 1]$	$[0, 1, 0, 1]$	$[0, 0, 1, 0]$
$[x_0 * x_2 + x_1 * x_2 + x_0 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3, x_2 * x_3, x_0 * x_1 + x_0 * x_2 + x_0 * x_3 + x_1 * x_3]$	$[0, 1, 1, 1]$	$[0, 1, 0, 1]$	$[0, 1, 1, 0]$
$[x_0 * x_2 + x_1 * x_2, x_1 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_1 * x_3, x_0 * x_1 + x_0 * x_2 + x_0 * x_3 + x_1 * x_3]$	$[0, 1, 1, 1]$	$[1, 0, 0, 1]$	$[1, 1, 1, 1]$
$[x_0 * x_2 + x_1 * x_2 + x_1 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_2 * x_3, x_0 * x_1 + x_0 * x_2 + x_1 * x_3]$	$[0, 1, 0, 1]$	$[1, 0, 1, 0]$	$[1, 1, 0, 1]$
$[x_0 * x_2 + x_1 * x_2 + x_0 * x_3, x_1 * x_2 + x_1 * x_3 + x_2 * x_3, x_0 * x_3 + x_1 * x_3, x_0 * x_1 + x_0 * x_2 + x_0 * x_3 + x_1 * x_3]$	$[1, 1, 0, 1]$	$[0, 1, 0, 1]$	$[0, 0, 1, 0]$
$[x_0 * x_2 + x_1 * x_2 + x_0 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3, x_2 * x_3, x_0 * x_1 + x_0 * x_2 + x_0 * x_3 + x_1 * x_3]$	$[0, 1, 1, 1]$	$[0, 1, 0, 1]$	$[0, 1, 1, 0]$
$[x_0 * x_2 + x_1 * x_2 + x_1 * x_3, x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_2 * x_3, x_0 * x_1 + x_0 * x_2 + x_1 * x_3]$	$[0, 1, 1, 1]$	$[1, 0, 0, 1]$	$[1, 1, 1, 1]$
$[x_0 * x_2 + x_1 * x_2 + x_1 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_2 * x_3, x_0 * x_1 + x_0 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3]$	$[0, 1, 1, 1]$	$[1, 0, 0, 1]$	$[1, 1, 1, 1]$
$[x_0 * x_2 + x_1 * x_2 + x_1 * x_3, x_1 * x_2 + x_2 * x_3, x_0 * x_3 + x_1 * x_3, x_0 * x_1 + x_0 * x_2]$	$[0, 0, 1, 0]$	$[1, 0, 0, 1]$	$[0, 1, 0, 0]$
$[x_0 * x_2 + x_1 * x_2 + x_0 * x_3 + x_1 * x_3, x_1 * x_2 + x_0 * x_3, x_1 * x_3, x_0 * x_1 + x_0 * x_2 + x_0 * x_3]$	$[1, 1, 0, 1]$	$[1, 0, 1, 1]$	$[0, 0, 0, 1]$

Table A.6: From a single $g(X) = [x_0 * x_2, x_1 * x_2, x_0 * x_2, x_0 * x_1]$, randomly selected 50 results of extension from $g(X) : F_{2^3} \to F_{2^4}$ to $f(X) : F_{2^4} \to F_{2^4}$

$f(X)$	A_0	A_1	A_2
$[x_0 * x_2 + x_0 * x_3 + x_1 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_2 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3]$	$[1, 1, 0, 1]$	$[1, 1, 1, 1]$	$[0, 0, 1, 1]$
$[x_0 * x_2, x_1 * x_2 + x_0 * x_3, x_0 * x_2 + x_1 * x_3, x_0 * x_1 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3]$	$[0, 1, 0, 1]$	$[0, 0, 1, 1]$	$[0, 0, 0, 1]$
$[x_0 * x_2 + x_1 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_1 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3]$	$[0, 1, 1, 1]$	$[1, 1, 1, 1]$	$[0, 0, 0, 1]$
$[x_0 * x_2 + x_1 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_0 * x_3 + x_2 * x_3]$	$[0, 1, 1, 1]$	$[1, 1, 1, 0]$	$[1, 1, 1, 1]$
$[x_0 * x_2 + x_1 * x_3 + x_2 * x_3, x_1 * x_2, x_0 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_1 + x_2 * x_3]$	$[0, 0, 1, 0]$	$[1, 0, 0, 0]$	$[1, 0, 1, 1]$
$[x_0 * x_2 + x_1 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_2 + x_1 * x_3, x_0 * x_1 + x_2 * x_3]$	$[0, 1, 0, 0]$	$[1, 1, 1, 0]$	$[1, 0, 0, 1]$
$[x_0 * x_2, x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_0 * x_3 + x_1 * x_3]$	$[0, 1, 1, 1]$	$[0, 0, 1, 1]$	$[0, 1, 1, 0]$
$[x_0 * x_2 + x_1 * x_3 + x_2 * x_3, x_1 * x_2 + x_1 * x_3 + x_2 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_1 + x_1 * x_3]$	$[0, 0, 1, 0]$	$[1, 1, 1, 1]$	$[1, 1, 0, 0]$
$[x_0 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_1 * x_2, x_0 * x_2 + x_1 * x_3, x_0 * x_1 + x_0 * x_3 + x_1 * x_3]$	$[1, 0, 0, 1]$	$[1, 0, 1, 1]$	$[1, 0, 0, 0]$
$[x_0 * x_2 + x_1 * x_3, x_1 * x_2 + x_1 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_0 * x_3]$	$[0, 0, 1, 1]$	$[1, 1, 1, 0]$	$[0, 0, 1, 0]$
$[x_0 * x_2 + x_0 * x_3 + x_2 * x_3, x_1 * x_2 + x_2 * x_3, x_0 * x_2 + x_1 * x_3, x_0 * x_1 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3]$	$[1, 0, 0, 1]$	$[0, 0, 1, 1]$	$[1, 1, 0, 1]$
$[x_0 * x_2 + x_0 * x_3 + x_1 * x_3, x_1 * x_2, x_0 * x_2 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_2 * x_3]$	$[1, 0, 0, 0]$	$[1, 0, 1, 0]$	$[0, 0, 1, 1]$
$[x_0 * x_2 + x_1 * x_3, x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_1 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3]$	$[0, 1, 1, 1]$	$[1, 0, 1, 1]$	$[0, 1, 0, 1]$
$[x_0 * x_2 + x_0 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_2 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_2 * x_3]$	$[1, 1, 0, 0]$	$[0, 1, 1, 0]$	$[0, 1, 1, 1]$
$[x_0 * x_2 + x_0 * x_3 + x_2 * x_3, x_1 * x_2 + x_2 * x_3, x_0 * x_2 + x_1 * x_3, x_0 * x_1]$	$[1, 0, 0, 0]$	$[0, 0, 1, 0]$	$[1, 1, 0, 0]$
$[x_0 * x_2, x_1 * x_2 + x_1 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_0 * x_3 + x_1 * x_3]$	$[0, 0, 1, 1]$	$[0, 1, 1, 1]$	$[0, 0, 1, 0]$
$[x_0 * x_2, x_1 * x_2 + x_0 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_1 + x_1 * x_3 + x_2 * x_3]$	$[0, 1, 1, 0]$	$[0, 0, 1, 1]$	$[0, 0, 0, 1]$
$[x_0 * x_2 + x_0 * x_3 + x_1 * x_3, x_1 * x_2 + x_0 * x_3, x_0 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_1 + x_1 * x_3 + x_2 * x_3]$	$[1, 1, 1, 0]$	$[1, 0, 0, 1]$	$[1, 1, 0, 1]$
$[x_0 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_1 + x_1 * x_3]$	$[1, 1, 1, 0]$	$[1, 0, 1, 1]$	$[1, 1, 0, 0]$
$[x_0 * x_2, x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_2 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_1 * x_3]$	$[0, 1, 0, 0]$	$[0, 0, 1, 1]$	$[0, 1, 1, 0]$
$[x_0 * x_2 + x_1 * x_3, x_1 * x_2 + x_0 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_0 * x_3 + x_1 * x_3]$	$[1, 0, 1, 1]$	$[0, 1, 1, 1]$	$[0, 0, 1, 0]$
$[x_0 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_2 + x_1 * x_3, x_0 * x_1 + x_0 * x_3 + x_2 * x_3]$	$[1, 0, 0, 1]$	$[1, 1, 1, 1]$	$[0, 0, 1, 0]$
$[x_0 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_1 * x_2 + x_1 * x_3, x_0 * x_2, x_0 * x_1 + x_0 * x_3 + x_1 * x_3]$	$[1, 0, 0, 1]$	$[1, 1, 0, 1]$	$[1, 0, 0, 0]$
$[x_0 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_0 * x_3 + x_2 * x_3]$	$[1, 1, 1, 1]$	$[1, 1, 0, 0]$	$[1, 0, 1, 1]$
$[x_0 * x_2 + x_0 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_2 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_0 * x_3 + x_1 * x_3]$	$[1, 1, 0, 1]$	$[0, 1, 1, 1]$	$[0, 1, 1, 0]$
$[x_0 * x_2 + x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_2 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_2 * x_3]$	$[0, 1, 0, 0]$	$[0, 1, 1, 0]$	$[1, 1, 1, 1]$
$[x_0 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_1 + x_1 * x_3 + x_2 * x_3]$	$[1, 1, 1, 0]$	$[1, 0, 1, 1]$	$[1, 1, 0, 1]$
$[x_0 * x_2 + x_1 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_1 + x_0 * x_3 + x_1 * x_3]$	$[0, 1, 1, 1]$	$[1, 0, 1, 1]$	$[0, 1, 1, 0]$
$[x_0 * x_2 + x_0 * x_3 + x_1 * x_3, x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_2 + x_1 * x_3, x_0 * x_1 + x_0 * x_3 + x_2 * x_3]$	$[1, 1, 0, 1]$	$[0, 1, 1, 1]$	$[0, 0, 1, 1]$
$[x_0 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_2 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3]$	$[1, 1, 0, 1]$	$[1, 1, 1, 0]$	$[1, 0, 1, 1]$
$[x_0 * x_2, x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_2 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_0 * x_3 + x_2 * x_3]$	$[0, 1, 0, 1]$	$[0, 1, 1, 0]$	$[0, 1, 1, 1]$
$[x_0 * x_2 + x_0 * x_3 + x_1 * x_3, x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_1 + x_1 * x_3 + x_2 * x_3]$	$[1, 1, 1, 1]$	$[1, 0, 1, 1]$	$[0, 0, 1, 1]$
$[x_0 * x_2 + x_1 * x_3, x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_1 + x_1 * x_3 + x_2 * x_3]$	$[0, 1, 1, 0]$	$[1, 0, 1, 1]$	$[0, 1, 0, 1]$
$[x_0 * x_2 + x_1 * x_3, x_1 * x_2 + x_0 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_1 + x_0 * x_3 + x_2 * x_3]$	$[0, 1, 1, 1]$	$[1, 1, 1, 1]$	$[0, 0, 0, 1]$
$[x_0 * x_2 + x_1 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3, x_0 * x_2 + x_1 * x_3, x_0 * x_1 + x_0 * x_3 + x_2 * x_3]$	$[1, 1, 1, 0]$	$[1, 0, 1, 1]$	$[1, 1, 0, 1]$
$[x_0 * x_2 + x_1 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_1 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3]$	$[0, 1, 1, 1]$	$[1, 1, 0, 1]$	$[1, 0, 0, 0]$
$[x_0 * x_2 + x_0 * x_3 + x_1 * x_3, x_1 * x_2 + x_1 * x_3, x_0 * x_2 + x_0 * x_3, x_0 * x_1 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3]$	$[1, 0, 0, 1]$	$[1, 0, 1, 1]$	$[0, 0, 1, 1]$
$[x_0 * x_2 + x_0 * x_3 + x_1 * x_3, x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_2 + x_1 * x_3, x_0 * x_1 + x_0 * x_3 + x_1 * x_3]$	$[1, 1, 0, 0]$	$[1, 0, 0, 1]$	$[1, 0, 0, 0]$
$[x_0 * x_2 + x_0 * x_3 + x_1 * x_3, x_1 * x_2 + x_0 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_1 + x_1 * x_3 + x_2 * x_3]$	$[1, 1, 1, 0]$	$[0, 1, 1, 1]$	$[0, 0, 1, 1]$
$[x_0 * x_2 + x_1 * x_3, x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_1 + x_0 * x_3 + x_2 * x_3]$	$[0, 1, 1, 1]$	$[1, 1, 0, 1]$	$[1, 0, 0, 1]$
$[x_0 * x_2 + x_0 * x_3 + x_1 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_1 + x_1 * x_3 + x_2 * x_3]$	$[1, 1, 0, 1]$	$[1, 1, 1, 0]$	$[1, 0, 1, 1]$
$[x_0 * x_2 + x_0 * x_3 + x_1 * x_3, x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_2 + x_1 * x_3, x_0 * x_1 + x_0 * x_3]$	$[1, 1, 0, 0]$	$[1, 0, 0, 1]$	$[1, 0, 0, 0]$
$[x_0 * x_2 + x_0 * x_3 + x_1 * x_3, x_1 * x_2 + x_2 * x_3, x_0 * x_2 + x_1 * x_3 + x_2 * x_3, x_0 * x_1]$	$[1, 0, 0, 0]$	$[1, 0, 1, 0]$	$[0, 1, 1, 0]$
$[x_0 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3, x_0 * x_2, x_0 * x_1 + x_1 * x_3]$	$[1, 1, 0, 0]$	$[1, 0, 0, 1]$	$[1, 0, 0, 0]$
$[x_0 * x_2 + x_0 * x_3 + x_1 * x_3, x_1 * x_2 + x_0 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_1 * x_3]$	$[1, 1, 1, 0]$	$[1, 0, 1, 1]$	$[0, 0, 1, 0]$
$[x_0 * x_2 + x_1 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_1 + x_0 * x_3]$	$[0, 1, 1, 0]$	$[1, 1, 0, 1]$	$[1, 0, 0, 0]$
$[x_0 * x_2 + x_0 * x_3 + x_1 * x_3, x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_1 + x_2 * x_3]$	$[1, 1, 1, 0]$	$[1, 0, 1, 0]$	$[0, 1, 1, 1]$
$[x_0 * x_2 + x_0 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3, x_0 * x_1 + x_0 * x_3 + x_2 * x_3]$	$[1, 1, 1, 1]$	$[0, 1, 1, 0]$	$[1, 0, 1, 1]$
$[x_0 * x_2 + x_1 * x_3, x_1 * x_2, x_0 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_1 + x_0 * x_3]$	$[0, 0, 1, 1]$	$[1, 0, 0, 0]$	$[0, 0, 1, 0]$
$[x_0 * x_2 + x_0 * x_3 + x_2 * x_3, x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_1 + x_0 * x_3 + x_1 * x_3 + x_2 * x_3]$	$[1, 1, 1, 1]$	$[0, 0, 1, 1]$	$[1, 1, 0, 1]$
$[x_0 * x_2, x_1 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_2 + x_1 * x_3, x_0 * x_1 + x_0 * x_3 + x_2 * x_3]$	$[0, 1, 0, 1]$	$[0, 1, 1, 0]$	$[0, 0, 0, 1]$
$[x_0 * x_2 + x_1 * x_3, x_1 * x_2 + x_0 * x_3 + x_2 * x_3, x_0 * x_2 + x_0 * x_3 + x_1 * x_3, x_0 * x_1 + x_1 * x_3 + x_2 * x_3]$	$[0, 1, 1, 0]$	$[1, 0, 1, 1]$	$[0, 1, 0, 1]$