

Malek Safieh

Algorithms and Architectures for Cryptography and Source Coding in Non-Volatile Flash Memories



Springer Vieweg

Schriftenreihe der Institute für Systemdynamik (ISD) und optische Systeme (IOS)

Chefredakteure

Jürgen Freudenberger, Institut für Systemdynamik, Hochschule Konstanz (HTWG), Konstanz, Baden-Württemberg, Deutschland

Johannes Reuter, Institut für Systemdynamik, Hochschule Konstanz (HTWG), Konstanz, Baden-Württemberg, Deutschland

Matthias Franz, Institut für Optische Systeme, Hochschule Konstanz (HTWG), Konstanz, Baden-Württemberg, Deutschland

Georg Umlauf, Institut für Optische Systeme, Hochschule Konstanz (HTWG), Konstanz, Baden-Württemberg, Deutschland

Die „Schriftenreihe der Institute für Systemdynamik (ISD) und optische Systeme (IOS)“ deckt ein breites Themenspektrum ab: von angewandter Informatik bis zu Ingenieurwissenschaften. Die Institute für Systemdynamik und optische Systeme bilden gemeinsam einen Forschungsschwerpunkt der Hochschule Konstanz. Die Forschungsprogramme der beiden Institute umfassen informations- und regelungstechnische Fragestellungen sowie kognitive und bildgebende Systeme. Das Bindeglied ist dabei der Systemgedanke mit systemtechnischer Herangehensweise und damit verbunden die Suche nach Methoden zur Lösung interdisziplinärer, komplexer Probleme. In der Schriftenreihe werden Forschungsergebnisse in Form von Dissertationen veröffentlicht.

The “Series of the institutes of System Dynamics (ISD) and Optical Systems (IOS)” covers a broad range of topics: from applied computer science to engineering. The institutes of System Dynamics and Optical Systems form a research focus of the HTWG Konstanz. The research programs of both institutes cover problems in information technology and control engineering as well as cognitive and imaging systems. The connective link is the system concept and the systems engineering approach, i.e. the search for methods and solutions of interdisciplinary, complex problems. The series publishes research results in the form of dissertations.

Weitere Bände in der Reihe <http://www.springer.com/series/16265>

Malek Safieh

Algorithms and Architectures for Cryptography and Source Coding in Non-Volatile Flash Memories



Springer Vieweg

Malek Safieh
University of Ulm
Munich, Germany

Dissertation, Ulm University, 2021

In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of the university of Ulm's or Springer Vieweg's products or services.

ISSN 2661-8087 ISSN 2661-8095 (electronic)
Schriftenreihe der Institute für Systemdynamik (ISD) und optische Systeme (IOS)
ISBN 978-3-658-34458-0 ISBN 978-3-658-34459-7 (eBook)
<https://doi.org/10.1007/978-3-658-34459-7>

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer
Fachmedien Wiesbaden GmbH, part of Springer Nature 2021

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Responsible Editor: Stefanie Eggert

This Springer Vieweg imprint is published by the registered company Springer Fachmedien Wiesbaden GmbH part of Springer Nature.

The registered company address is: Abraham-Lincoln-Str. 46, 65189 Wiesbaden, Germany

Acknowledgements

First and foremost, I would like to thank my doctoral supervisor Prof. Dr. Jürgen Freudenberger for giving me the opportunity to join the research group at the Institute of System Dynamics (ISD) at the University of Applied Sciences (HTWG) Konstanz, Germany. Due to his efforts, it became possible for me to work on my Ph.D. for the last three and a half years. I appreciate the valuableness, the time, and efforts he offered for the long and fruitful discussions we had. I am very grateful for his advices and supports that contributed to the improvement of my research skills.

I would also like to thank Prof. Dr. Martin Bossert for reading my thesis and being my second supervisor. I am very thankful to all the research members at the ISD. It would be truly unfair to mention only a few people since everyone made my time there a pleasure. The conversations during the coffee breaks in the kitchen, after eating in the canteen, as well as during the after-work beer were always fun, kept me motivated and with a fresh mindset.

I would like to express my gratitude to my family and friends for all their help and contribution during my study and the research period. Further thanks goes to everyone who supported me during my doctoral research.

Abstract

Flash memory is an important non-volatile storage medium. Reliable and secure data storage in flash memories requires sophisticated coding and signal processing techniques. Although error-correcting codes are applied in practically all flash storage systems, coding techniques for cryptography and data compression are less developed.

Due to the limited computational performance of the flash controller, many flash storage systems rely on symmetric cryptography for the message authentication. Asymmetric cryptography like elliptic curve cryptographic (ECC) systems offer additional functionality such as digital signatures and key exchange methods, which allows a verification of the integrity and authenticity. In this work, we demonstrate that ECC systems over Gaussian integers are very efficient. Gaussian integers are a subset of complex numbers with integers as real and imaginary parts. Since many Gaussian integer fields are isomorphic to prime fields, this arithmetic is suitable for ECC systems. Implementations of cryptographic algorithms are prone to side channel attacks. We show that using Gaussian integers can reduce the complexity and memory requirements for hardware implementations which are protected against such attacks.

However, determining the modulo reduction over Gaussian integers is extremely expensive. To reduce the complexity, we derive a Montgomery modular arithmetic over Gaussian integers. Moreover, we develop a hardware architecture optimized for the ECC operations targeting low area. The proposed ECC processor for Gaussian integers is a competitive solution for applications in flash memories and other resource-constrained embedded systems.

Data compression provides several advantages for flash memory controllers, e.g. improving the lifetime and storage capacity. In this work, we focus on the

dictionary-based Lempel-Ziv-Welch (LZW) algorithm. A fast and compact implementation of this universal data compression procedure is a challenging task due to the recursive structure of the LZW dictionary. To speed up the encoding, we present an architecture that applies multiple dictionaries in parallel. Two dictionary partitioning techniques are introduced that improve the compression rate and reduce the memory size of this parallel dictionary LZW algorithm.

Contents

1	Introduction	1
1.1	Problem Statement and Motivation	1
1.2	Structure of the Thesis	3
2	Elliptic Curve Cryptography	5
2.1	Cryptography for Flash Memory Controllers	5
2.2	Applications of Elliptic Curve Cryptographic (ECC) Systems	7
2.3	Elliptic Curve Point Multiplication	14
2.4	Elliptic Curve Geometry and Group Laws	17
2.5	Reducing the Number of Field Inversions for Elliptic Curves over Prime Fields	20
2.6	Discussion	22
3	Elliptic Curve Cryptography over Gaussian Integers	25
3.1	Gaussian Integer Rings and Fields	26
3.2	Point Multiplication over Gaussian Integers	29
3.3	Resistance Against side Channel Attacks using Gaussian Integers	33
3.4	Discussion	41
4	Montgomery Arithmetic over Gaussian Integers	43
4.1	Montgomery Arithmetic	44
4.2	Reduction over Gaussian Integers using the Absolute Value	46
4.3	Reduction over Gaussian integers using the Manhattan Weight	51
4.4	Simplifying the Reduction based on the Manhattan Weight	57
4.5	Discussion	59

5	Architecture of the ECC Coprocessor for Gaussian Integers	61
5.1	Coprocessor Architecture for Gaussian Integers	62
5.2	ECC Coprocessor Architecture for Prime Fields	70
5.3	Implementation Results	76
5.4	Discussion	81
6	Compact Architecture of the ECC Coprocessor for Binary	
	Extension Fields	83
6.1	Group Laws and Projective Coordinates for Binary Extension Fields	84
6.2	ECC Coprocessor Architecture	87
6.3	Results and Discussion	94
7	The Parallel Dictionary LZW Algorithm for Flash Memory	
	Controllers	97
7.1	Data Compression for Flash Memory Devices	99
7.2	Parallel Dictionary LZW (PDLZW) Algorithm	107
7.3	Address Space Partitioning for the PDLZW	112
7.4	Reducing the Memory Requirements of the PDLZW	116
7.5	Compression and Implementation Results	124
7.6	Discussion and Comparison with Other Data Compression Schemes	128
8	Conclusion	131
	Bibliography	133

Acronyms

ADD	point addition
AES	advanced encryption standard
ANSI	American National Standards Institute
ASCII	American standard code for information interchange
ASIC	application-specific integrated circuit
ASIP	application-specific instruction set processor
BCH	Bose Chaudhuri Hocqenhem
BMH	BWT-MTF-Huffman
BWT	Burrows-Wheeler transformation
CAM	content-addressable memories
DBL	point doubling
DPA	differential power analysis
DP-RAM	dual-port RAM
DSP	digital signal processor
ECC	elliptic curve cryptographic
FF	flip-flops
FPGA	field programmable gate array
FTL	flash translation layer
GF	Galois field
HDD	hard disc drives
HMAC	keyed-hash message authentication code
KB	Kilobyte
LUT	lookup tables
LZW	Lempel-Ziv-Welch
MAC	message authentication codes
MH	MTF-Huffman

MLC	multi-level cell
MTF	move-to-front
NIST	National Institute of Standards and Technology
P/E	program/erase
PDLZW	parallel dictionary LZW
PM	elliptic curve point multiplication
RAM	random-access memory
RIP	randomized initial point
RPA	refined power analysis
RSA	Rivest-Shamir-Adleman
SADD	special point addition
SCA	side channel attacks
SD	secure digital
SPA	simple power analysis
SSD	solid-state drives
TA	timing attacks
USB	universal serial bus
WER	word error rate
ZPA	zero-value point attacks

List of Figures

Figure 2.1	Examples for elliptic curve over real numbers	9
Figure 2.2	Principal of digital signatures	12
Figure 2.3	Hierarchy of calculating the point multiplication for prime fields	15
Figure 2.4	Point addition (ADD) example	18
Figure 2.5	Point doubling (DBL) example	19
Figure 3.1	The set of Gaussian integers for $\pi = 4 + i$	28
Figure 4.1	Elements of the Gaussian integer field $\mathcal{G}_{29}$ with $\pi = 5 + 2i$	46
Figure 4.2	Illustration of the the Montgomery domain for $p = 29$ and $\pi = 5 + 2i$	50
Figure 4.3	Illustration of the the Montgomery domain for $p = 53$ and $\pi = 7 + 2i$	56
Figure 5.1	Block diagram of the ECC processor	65
Figure 5.2	Structure of the Karatsuba multiplier	69
Figure 5.3	Structure of the arithmetic unit (AU)	75
Figure 6.1	Block diagram of the ECC processor	88
Figure 6.2	Block diagram of the parallelized least significant bit-first multiplier	93
Figure 7.1	Example of the garbage collection procedure	100
Figure 7.2	Codeword structure with and without the combination of data compression techniques and an error-correcting code	102
Figure 7.3	Illustrating the lifetime improvement using word error rates with different data compression algorithms for the Canterbury corpus according to [4]	104

Figure 7.4	Example of the PDLZW encoding	109
Figure 7.5	Illustration of a Markov chain for the PDLZW encoding	112
Figure 7.6	Stationary probability distribution for Calgary corpus for an address space with 1024 entries	114
Figure 7.7	Comparing the PDLZW with the recursive PDLZW	117
Figure 7.8	Word partitioning principal	119
Figure 7.9	Distribution of the number of unique entries for the different dictionaries	121
Figure 7.10	Structure of the PDLZW encoding with word partitioning	123
Figure 7.11	Compression results for the Calgary corpus with different input block lengths and different number of entries (address space)	125
Figure 7.12	Compression results for the Canterbury corpus with a partitioning derived from the Calgary corpus and $q \leq 0.5$ for different input block lengths and different number of entries (address space)	126

List of Tables

Table 2.1	Comparison key sizes of ECC and RSA cryptographic systems for equivalent security levels in bits according to [14]	7
Table 3.1	Examples for primes of the form $p = a^2 + b^2$ with $b = 1$ or $b = a - 1$	29
Table 3.2	Number of multiplications and squarings using Jacobian coordinates according to [14, 78]	38
Table 3.3	Number of point operations required for the precomputations and an iteration of the point multiplication with different τ	39
Table 3.4	Performance results for different τ -adic expansions in comparison with [31] for a binary key length $r = 163$	40
Table 4.1	Examples of primes of the form $p = a^2 + b^2$ suitable for ECC applications and the percentage of the number of offset reduction steps required (red.)	58
Table 5.1	Definition of the instruction set for the Gaussian integers arithmetic	66
Table 5.2	Definition of the instruction set	73
Table 5.3	Comparison of the hardware resources with other area-efficient implementations	79
Table 5.4	Latencies comparison with other area-efficient implementations	80
Table 6.1	Definition of the instruction set	89
Table 6.2	Number of arithmetic operations for DBL and ADD over $GF(2^m)$	90
Table 6.3	Required resources for the proposed architecture	94

Table 6.4	Comparison with other area-efficient implementations	95
Table 7.1	Determination of dictionary sizes using the probability distribution	115
Table 7.2	Proposed dictionary partitioning for different address spaces	115
Table 7.3	Mean compressed block length of LZW for different block lengths	116
Table 7.4	Hardware size for the register-based implementation	127
Table 7.5	Hardware size for the RAM-based implementation	127
Table 7.6	Mean block size for the PDLZW with word partitioning (4 dict.)	128
Table 7.7	Comparison with other data compression schemes	129