QUEEN'S UNIVERSITY BELFAST

DOCTOR OF PHILOSOPHY

Practical lattice-based cryptography over structured lattices

McCarthy, Sarah

Award date:
2020

Awarding institution:
Queen's University Belfast

Link to publication

Terms of use
All those accessing thesis content in Queen's University Belfast Research Portal are subject to the following terms and conditions of use

- Copyright is subject to the Copyright, Designs and Patent Act 1988, or as modified by any successor legislation
- Copyright and moral rights for thesis content are retained by the author and/or other copyright owners
- A copy of a thesis may be downloaded for personal non-commercial research/study without the need for permission or charge
- Distribution or reproduction of thesis content in any format is not permitted without the permission of the copyright holder
- When citing this work, full bibliographic details should be supplied, including the author, title, awarding institution and date of thesis

Take down policy
A thesis can be removed from the Research Portal if there has been a breach of copyright, or a similarly robust reason.
If you believe this document breaches copyright, or there is sufficient cause to take down, please contact us, citing details. Email: openaccess@qub.ac.uk

Supplementary materials
Where possible, we endeavour to provide supplementary materials to theses. This may include video, audio and other types of files. We endeavour to capture all content and upload as part of the Pure record for each thesis.
Note, it may not be possible in all instances to convert analogue formats to usable digital formats for some supplementary materials. We exercise best efforts on our behalf and, in such instances, encourage the individual to consult the physical thesis for further information.

Download date: 24. Apr. 2024

Practical Lattice-based Cryptography over Structured Lattices

Sarah Caroline McCarthy MSci

Supervisor: Prof. Máire O'Neill

Dr Ciara Rafferty

Dr Elizabeth O'Sullivan

School of Electronics, Electrical Engineering, and Computer Science,

Queen's University Belfast.

This dissertation is submitted for the degree of

Doctor of Philosophy
June 2020

Throughout the period of writing this thesis, I have both loved and lost. Within a week of starting my PhD, I lost my Grandma Kennedy, who taught me to pursue my dreams, and smuggled me multiple £20 notes behind my mother's back to help me fund my escapades. Just over a year later, I lost my Granny McCarthy, who I'm convinced I got my stubbornness from...the number one attribute required to finish a PhD. In December 2017, we lost our family dog Lucy, who had been my study buddy throughout school and university, and without whom I would have got lonely during my late night study sessions and wouldn't have made it to where I am today.

I learned what heartbreak was, which also taught me resilience, staying true to oneself, when to quit and when not to, and discovering who your true friends are.

My best friends: Emily, Sam, Tori, Emma, Jemma, Dr. Laura Boyle (we did it!), Sophia, Jennie, Nicola and Lee, who I've known the shortest yet it feels like the longest and is my number one supporter.

My family: Mum and Dad for always putting me first and doing my ironing during thesis write-up, and "big bro" Christopher who I'll always look up to, even if I am more qualified than him.

My horses Tara and Pippa, cats Humbug and Wotsit and crazy dog Rita (who I believe rescued me more than I rescued her), all of whom taught me what love and partnership is and can make my heart explode without saying a word.

There were many occasions when I thought I wouldn't finish this PhD, so in the words of Taylor Swift:

But I got smarter, I got harder in the nick of time

Honey, I rose up from the dead, I do it all the time

Declaration

I hereby declare that, except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 80,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Sarah Caroline McCarthy MSci

June 2020

Acknowledgements

I like lists, and so this seems an appropriate place for one....

· My colleague Dr. Domhnall Carlin for his careful thesis-reading and suggestions

· Ms Jill Doherty for assisting with all my travel plans

· Mrs Emma McLarnon for her constant encouragement and TLC

· Mr Seamus Brannigan, for his jokes which sometimes required some thought, and his help with “all that computer stuff" I didn’t understand

· Dr. James Howe for collaborations, giving my papers the “Howe" factor, and hitting up the vegan buffets at events

· Dr. Ayesha Khalid for (sometimes) non-sensical discussions and for being my cheerleader and conference buddy

· Dr. Neil Smyth for his out-of-this-world programming skills

· Dr. Thomas Prest, the P in DLP and the wings of Falcon-sampling

· My advisors in The National Cyber Security Centre, Peter C and Michael G

· Dr. Neil Hanley for putting up with my blonde moments throughout the LATTE implementation

· Dr. Tony Barnett from Thales UK for his on-call out-of-hours attitude to my coding issues

· My former supervisor Dr. Elizabeth O’Sullivan, who started me on my PhD journey

· My second-supervisor, and good friend, Dr. Ciara Rafferty

· My supervisor, role model and pioneer of her field, Prof. Máire O’Neill.

Abstract

Alongside the growth of technology is the increasing demand for security of data and communication channels. As attacking power grows, devices become smaller and data expands, the security needs to be stronger yet less resource-consuming than ever before.

Public key cryptography (PKC) is used in the majority of today's communications systems. The security of public-key cryptographic schemes, such as RSA and elliptic curve cryptography (ECC), is based on underlying mathematical problems such as factoring large integers or solving the discrete log problem. However, the discovery of Shor's Algorithm in 1994 insinuates that these schemes will be insecure when a full-scale quantum computer is developed, as these problems will be solvable in polynomial time. Therefore attention has moved towards post-quantum, or quantum-safe, cryptography (QSC).

In 2016, the National Institute of Standards and Technology (NIST) announced the beginning of their efforts to migrate to QSC, a standardisation process which has involved the cryptographic community worldwide. There are five main types of QSC being considered for standardisation; lattice-based, code-based, multivariate-based, hash-based and isogeny-based.

Lattice-based cryptography (LBC) offers a promising alternative to today's classical approaches. Hard problems based on lattices are believed to be unsolvable, even by a quantum-computer, in a feasible amount of time. In 1995, Ajtai showed that the average case complexity of these problems is at least as hard as solving the problems in the worst case. Furthermore, lattice-based schemes are the only remaining contender in the NIST process to provide the important primitives of encryption, key encapsulation and digital signature schemes (DSSs).

A well-recognised open problem surrounding quantum-safe cryptography is its increase in bandwidth and computational complexity in comparison to existing cryptography. However, in comparison to other quantum-safe schemes, LBC has been shown to be relatively efficient. In this thesis, novel software designs of lattice-based schemes are presented and shown to be competitive with existing PKC. Additionally, physical attacks are considered and countermeasures are incorporated into the designs.

DSSs are required for authentication, integrity and non-repudiation of data. This thesis provides an evaluation of three leading signature schemes, BLISS-B, Dilithium and Falcon, which vary in terms of their key features, such as the sampling processes used, and the structure of the lattices. Physical attacks on the three schemes are considered and the effect on performance of appropriate attack countermeasures is evaluated. A novel fault attack on the NIST candidate Falcon is also proposed.

A further advantage of LBC is the ability to support cryptographic primitives with extended capabilities to facilitate certificate management and compute on encrypted data. The complexity of today's technology advances mean applications require more sophisticated primitives, which contrastingly are required to run on even smaller platforms. This thesis presents novel quantum-safe advanced cryptographic schemes, specifically lattice-based identity-based encryption (IBE) and hierarchical identity-

based encryption (HIBE). IBE removes the need for certificate management, as a user's public key depends on their public identity. IBE is currently used in vital applications such as emergency services communication and so the development of a high-performance quantum-safe alternative is imperative.

This thesis also introduces the first scalable HIBE scheme based on lattices. The first HIBE scheme based on structured lattices, Latte, was proposed by the National Centre for Cyber Security (NCSC) in 2017. However, the public key and ciphertext sizes increased with each delegation and hence it was not feasible to scale the algorithm. Skinny Latte, proposed in this thesis, remedies this issue by fixing the dimension of the lattice, thereby making public key and ciphertext sizes constant. This research also shows that it is practical for real-world deployment.

Overall, this thesis demonstrates that lattice-based cryptography is practical for real-world applications. It considers both theoretical and physical attacks, and presents novel designs of advanced quantum-safe cryptographic primitives, which have only yet been realised by lattices.

Nomenclature

5G 5th generation mobile network.

ABE Attribute Based Encryption.

AKE Authenticated Key Exchange.

BDD Bounded Distance Decoding.

BKZ Block Korkine-Zolotarev lattice reduction algorithm.

BLISS The bimodal lattice-based digital signature scheme of Ducas et al. [2013a].

CA Certificate Authority.

CDF Cumulative Distribution Function.

CDT Cumulative Distribution Table.

CRT Chinese Remainder Theorem.

CSPRNG Cryptographically Secure Pseudo-Random Number Generator.

CVP the Closest Vector Problem.

DSA Digital Signature Algorithm.

DSS Digital Signature Scheme.

ECC Elliptic-Curve Cryptography.

ECDSA Elliptic-Curve Digital Signature Algorithm.

ECIES Elliptic Curve Integrated Encryption Scheme.

ETSI European Telecommunications Standards Institute.

FFT Fast Fourier Transform.

FHE Fully Homomorphic Encryption.

GapCVP The approximate decision version of the Closest Vector Problem.

GapSVP The approximate decision version of the Shortest Vector Problem.

GCD Greatest Common Divisor.

GMP GNU Multiple Precision Arithmetic Library.

GS-norm Gram-Schmidt Norm.

GSB GS-norm Bound.

GSO Gram-Schmidt Orthogonalisation.

HIBE Hierarchical Identity Based Encryption.

HSVP Hermite Shortest Vector Problem.

IBE Identity Based Encryption.

iNTT Inverse Number Theoretic Transform.

IoT Internet of Things.

KDF Key Derivation Function.

KEM Key Exchange Mechanism.

KMS Key Management Service.

LBC Lattice-Based Cryptography.

LHS Left Hand Side.

LLL The Lenstra–Lenstra–Lovász lattice reduction algorithm.

LWE the Learning With Errors problem.

MPC Multi-Party Computation.

NCSC UK National Cyber Security Centre.

NIST National Institute of Standards and Technology.

NP Non-deterministic Polynomial-time.

NTL Number Theoretic Library.

NTT Number Theoretic Transform.

PKC Public Key Cryptography.

PKE Public Key Encryption.

PKI Public Key Infrastructure.

PoC Proof of Concept.

PPT Probabilistic Polynomial Time.

PQC Post-Quantum Cryptography.

Q-ROM Quantum Random Oracle Model.

QKD Quantum Key Distribution.

QSC Quantum-Safe Cryptography.

R-LWE the Ring variant of the Learning With Errors (LWE) problem.

R-SIS the Ring variant of the Short Integer Solution (SIS) problem.

RHS Right Hand Side.

RNG Random Number Generator.

ROM Random Oracle Model.

RSA the cryptosystem named after its inventors Rivest, Shamir and Adleman [Rivest et al., 1978].

SBP The Smallest Basis Problem.

SCA Side-Channel Analysis.

SIMD Single Instruction, Multiple Data.

SIS the Short Integer Solution problem.

SIVP The Shortest Independent Vector Problem.

SVP the Shortest Vector Problem.

TETRA TErrestrial Trunked RAdio.

uSVP The unique Shortest Vector Problem.

XOF eXtendable-Output hash Function.

Table of contents

Nomenclature **viii**

List of figures **xix**

List of tables **xxi**

1 Introduction **1**

1.1 Motivation 1

1.2 Objectives 7

1.3 Novel Contributions 9

1.4 Organisation of Thesis 11

1.5 Collaborative contributions 13

2 Background and Theory **14**

2.1 Introduction 14

2.2 Preliminaries 15

2.2.1 Mathematics and Notation 15

2.2.2 Cryptographic and Security Concepts 17

2.3 NIST Standardisation Process 19

2.3.1 Introduction 19

2.3.2 Submissions 19
2.3.3 Round 1 20
2.3.4 Round 2 21
2.3.5 Lattice-based NIST candidates 22
2.3.6 Next steps to standardisation 23
2.4 Lattice-based Cryptography 23
2.4.1 Lattices 23
2.4.2 More on NTRU Lattices 28
2.4.3 Hard Lattice Problems 30
2.4.4 Basic Lattice-based Primitives 33
2.4.5 Advanced Lattice-based Primitives 48
2.5 Mathematical building blocks of LBC primitives 50
2.5.1 Hash functions 50
2.5.2 Gaussian Sampling 52
2.5.3 Polynomial Arithmetic and the Number Theoretic Transform (NTT) 53
2.5.4 Gram-Schmidt Orthogonalisation (GSO) 55
2.5.5 Trapdoor Sampling 55
2.6 Basis Reduction Methods 57
2.6.1 LLL 57
2.6.2 BKZ 60
2.6.3 BKZ 2.0 61
2.7 Existing practical constructions and implementations 62
2.7.1 H2020 SAFEcrypto project 67
2.8 Discussion 69

3 The Impact of Physical Attack Countermeasures on the Performance of Lattice-based Digital Signature Schemes 71

3.1 Introduction 71

3.2 Physical attacks on LBC DSS 74

3.2.1 BLISS-B 78

3.2.2 Dilithium 79

3.2.3 Falcon 81

3.3 Bearz attack Falcon 85

3.3.1 Recursion in Falcon 87

3.3.2 FFT: Merging and Splitting 88

3.3.3 Method of attack 89

3.3.4 Post-attack processing 93

3.3.5 Attack results 94

3.3.6 Fault Model and Feasibility of Attack 95

3.4 SCA Countermeasures 98

3.4.1 Double computation and verify-after-sign 99

3.4.2 Sample Discard 99

3.4.3 Shuffling 99

3.4.4 Constant-time 100

3.4.5 Zero-check 102

3.5 Results and evaluation 102

3.5.1 BLISS-B with Countermeasures 103

3.5.2 Dilithium with Countermeasures 104

3.5.3 Falcon with Countermeasures 107

3.5.4 Performance Evaluation 109

3.6 Conclusions 110

4 Practical Identity-Based Encryption over NTRU Lattices 114

4.1 Introduction 114

4.2 Identity-Based Encryption 116

4.2.1 Current IBE schemes 120

4.2.2 Lattice-based IBE schemes 121

4.3 The DLP-IBE Scheme 122

4.3.1 Description of the DLP-IBE Scheme 123

4.3.2 Optimised Software Design of the DLP-IBE Scheme 133

4.4 Performance results and discussion 143

4.5 The Falcon-IBE Scheme 146

4.5.1 Description of the Falcon-IBE Scheme 146

4.6 Conclusions 146

5 Skinny Latte: A Novel Scalable Hierarchical Identity-Based Encryption Scheme 150

5.1 Introduction 150

5.2 Hierarchical Identity-based Encryption 152

5.2.1 Existing lattice-based (H)IBE schemes 155

5.3 Latte 157

5.3.1 Theoretical Analysis 162

5.3.2 Performance 165

5.4 Skinny Latte 166

5.4.1 Skinny Latte description 166

5.4.2 Latte to Skinny Latte 170

5.4.3 Security Analysis and Parameter Derivation 172
5.5 Conclusions 194

6 Conclusion and Future Work 195

6.1 Conclusion Summary 195
6.1.1 Chapter 3 196
6.1.2 Chapter 4 198
6.1.3 Chapter 5 199
6.2 Topics For Future Research 199
6.2.1 QSC + QKD 200
6.2.2 Further advanced primitives 200
6.2.3 Gaussian Sampling over the lattice 201
6.2.4 Tightening security 202
6.2.5 Further NIST evaluation 202

References 204

Appendix A Reference code and algorithms 225

A.1 Random oracles used in DLP-IBE design 225
A.2 FFT Sampler [Prest et al., 2017] 226
A.3 Merge FFT 228

Appendix B Falcon family 230

Appendix C Additional cryptographic algorithms 236

C.1 LWE PKE [Peikert et al., 2008] 236
C.2 R-LWE DSS [Güneysu et al., 2012a] 237

C.3 Kyber KEM [Schwabe et al., 2017] 237

Appendix D Author's Publications and Contributions 242

List of figures

2.1 The distribution of NIST candidate types 20

2.2 A 2-dimensional lattice [Tourloupis, 2012] 24

2.3 Reductions between lattice problems 34

2.4 The workings of a PKE scheme 36

2.5 The workings of a DSS scheme 39

2.6 The evolution of lattice-based digital signature schemes (* indicates further relations between schemes, explained in text) 41

2.7 The workings of a KEM 47

2.8 A Gaussian distribution curve 53

3.1 The recursion tree of the FFT Sampler and Bearz attack points $(n = N)$ 88

3.2 The strategies of the Bearz and DLP attacks 90

3.3 Protected Dilithium (+Double comp. and Verify-after-sign) vs. protected Falcon (+zero-check) 110

3.4 Performance analysis of the proposed SCA countermeasures for the three signature schemes. 111

4.1 The workings of an IBE scheme 118

4.2 Babai's algorithm vs. the GPV sampler 130

5.1 A 2-level HIBE scheme 153

5.2 Basis changes during Skinny Latte delegation 169

B.1 The components of the DLP IBE scheme 231

B.2 The components of the Falcon Signature scheme 232

B.3 The components of the Falcon IBE scheme 233

B.4 The components of the Latte HIBE scheme 234

B.5 The components of the Skinny Latte HIBE scheme 235

List of tables

2.1 The security levels specified by NIST in their call for submissions . . 20

2.2 Schemes which merged to create new Round 2 candidates 22

2.3 The lattice-based schemes involved in the NIST standardisation process Round 1 (Round 2: **selected**/*merged*) 23

2.4 The three types of lattices used in cryptography 27

2.5 Adversarial security concepts 37

2.6 Adversarial DSS security concepts [Goldwasser et al., 1988] 40

2.7 Comparison of trapdoor samplers [Prest et al., 2017] 57

2.8 Hermite Factors reached by lattice reduction algorithms 61

2.9 Lattice-based software libraries 64

2.10 Implementation options for LBC components(IC-implementor choice) 65

2.11 A taxonomy of the schemes in this thesis 66

3.1 A taxonomy of the lattice-based signature schemes. 77

3.2 Timings (in seconds) to run the BKZ algorithm on l faulty signatures to obtain the basis polynomial F running on a single core of an Intel Core i7-6700HQ CPU at 2.60 GHz. 95

3.3 Corresponding attacks and countermeasures 98

3.4 Performance results (ops/sec) for BLISS-B with SCA countermeasures, on Intel E5-1620@ 3.7GHz unless otherwise stated. KeyGen figures are not given in the reference papers. 105
3.5 Performance results (ops/sec) for Dilithium with SCA countermeasures, on Intel E5-1620@ 3.7GHz unless otherwise stated. 106
3.6 Performance results (ops/sec) for Falcon with SCA countermeasures, including % decrease in affected components, on Intel E5-1620@ 3.7GHz unless otherwise stated. 108

4.1 Algorithm summary 123
4.2 Comparison with hardware design of DLP-IBE 139
4.3 IBE Scheme original parameters 142
4.4 Proposed IBE Scheme parameters 142
4.5 DLP-IBE key and encrypted message bit sizes. All values are in bits. . 142
4.6 DLP-IBE performance in terms of KeyGen processing time in seconds and Extract, Encrypt and Verify operations per second on an Intel Core i7 6700 @ 4 GHz with SHA-3, Cumulative Distribution Table (CDT) Gaussian Sampling and AES CTR-DRBG 143
4.7 Prest's NTL implementation results comparison, operations per sec [Prest, 2016] 144
4.8 Performance comparison of conventional schemes (results from [Cheng et al., 2011], scaled to 4GHz) and the DLP-IBE scheme (this work). . . 145
4.9 Hardware performance results of DLP-IBE scheme (cycles) [O'Neill et al., 2016] 149

5.1 Latte Algorithm summary 158

5.2 Latte Parameters 162
5.3 Latte Public Key and Ciphertext Sizes (bytes) 162
5.4 Latte Estimated cost of delegated key recovery 163
5.5 Latte Estimated cost of user key recovery 163
5.6 LATTE performance results (op/s) 164
5.7 Timings in ms for the Gentry-Silverberg HIBE scheme (2-level) on Intel Core i5 4GB Mac OS X 10.6.7 using Java v1.6 [Luzhnica, 2011] . 166
5.8 Summary of Skinny Latte components 168
5.9 Skinny Latte Public Key and Ciphertext Sizes bytes 176
5.10 LATTE Master Key Recovery Estimated Costs in bits Campbell and Groves [2017] 185
5.11 Skinny Latte Estimated cost of delegated key recovery 186
5.12 Skinny Latte Estimated cost of user key recovery 187
5.13 Skinny Latte $(l = 1)$ Message Recovery via Primal-CVP Estimated Costs 187
5.14 Skinny Latte $(l = 1)$ Estimated cost of message recovery Dual-SVP attack 188
5.15 Proposed Skinny Latte Parameters 189

CHAPTER 1

Introduction

1.1 Motivation

Quantum computers will bring much benefit in fields such as medicine and engineering with the ability to quickly compute on large amounts of data and and solve complex algorithms, they will be detrimental to computer security. As of 2019, the current qubit record, which is the fundamental unit of quantum information, is 72 qubits due to the work of Google[1]. The interest of large-scale conglomerates such as D-Wave and Google into quantum computing provides a driving force to the research area, with full-scale machines expected within the next 30 years [Mosca, 2018]. In 2014, Edward Snowden revealed that the NSA was funding a $79.7M research program to build a "cryptologically useful quantum computer"[2]. However, their true capability in this area is unknown. Events such as this highlight the requirement for more advanced and secure cryptography.

[1] https://ai.googleblog.com/2018/03/a-preview-of-bristlecone-googles-new.html

[2] https://www.theguardian.com/world/interactive/2013/nov/01/snowden-nsa-files-surveillance-revelations-decoded

The expansion of the Internet of Things (IoT) creates as many vulnerabilities as it does opportunities, and the risks are increasingly complex with the expanded attack surface. This demands strong cryptography to protect against hacks and attacks, and to protect the privacy and integrity of confidential data. More advanced applications offered by the IoT, smart cities and global communications call for more advanced primitives. Scenarios have arisen where it is required that data is unreadable by an outsider, but where a user may wish to allow computation on encrypted data or be able to expose certain parts of the data to those with specific credentials. The extended lifespan of many products and data also calls for systems which can remain protected against future unknown attacks. Furthermore, these efforts have to start *today* for applications to be protected in the future. For instance, a car produced today has a lifespan of around 12 years and so its central processing unit needs to be secure for this length of time, yet it takes up to 10 years of research and development before the required cryptography can be deployed into the cars. Therefore, attacks in 20/30 years' time need to be mitigated today.

In 1997, Peter Shor developed an algorithm that would initiate a revolution in modern cryptography. When run on a quantum computer, Shor's algorithm [Shor, 1997] can break existing public-key cryptographic protocols in polynomial time, a feasible amount of time for an attacker to act. It can solve the underlying mathematical problem of factorising products of large primes (used in the cryptosystem named after its inventors Rivest, Shamir and Adleman [Rivest et al., 1978] (RSA)) and the discrete logarithm problem (used in Elliptic-Curve Cryptography (ECC)). As these are the backbone of all cryptographic protocols used today, attention has turned to post-quantum, or quantum safe, cryptography. These algorithms can run on classical computers used today, yet remain secure against known quantum computing attacks.

Therefore the research community has turned to more powerful primitives to secure our data. University research groups, governments worldwide, standards bodies such as the National Institute of Standards and Technology (NIST) and the European Telecommunications Standards Institute (ETSI) and companies like Google, IBM and many others are currently researching Post-Quantum Cryptography (PQC) for this purpose. This thesis focusses on lattice-based cryptography, specifically those schemes which have advanced capabilities, and shows that they can be made efficient for use in real world applications, even on constrained devices.

There are five strands of PQC currently being examined by the research community.

- *Lattice-based*: Arguably the most popular of the post-quantum contenders, lattices are characterised by their associated worst-case hardness problems, such as finding short vectors in a lattice, upon which both basic, such as Public Key Encryption (PKE) [Regev, 2010], Key Exchange Mechanism (KEM)s [Schwabe et al., 2017], Digital Signature Scheme (DSS)s [Lyubashevsky et al., 2017, Prest et al., 2017, Ducas, 2014b], and more advanced cryptographic primitives such as Fully Homomorphic Encryption (FHE) [Gentry et al., 2013], Identity Based Encryption (IBE) [Ducas et al., 2014a] and e-voting [Chillotti et al., 2016] can be built. A popular lattice problem is Learning With Errors (LWE) (discussed in §2.4.1), and many LWE-based schemes have proven to be just as, if not more, efficient than existing comparable primitives. An acknowledged challenge of Lattice-Based Cryptography (LBC) is large key sizes, which can be addressed through adding structure to the lattice (at the expense of a stronger security assumption). The reader is referred to Chapter 2 for further details on LBC.

- *Code-based*: In this family, the underlying one-way function is an error-correcting code. Code-based cryptography can provide PKE [McEliece, 1978, Niederreiter, 1986], KEMs [Aragon et al., 2017a], DSSs [Courtois et al., 2001, Kabatianskii et al., 1997], identification schemes [Stern, 1993], random number generators [Fischer and Stern, 1996, Gaborit et al., 2007] and hash functions [Augot et al., 2005]. Similarly to LBC, a challenge surrounding code-based systems is their large key sizes (greater than 1 MB for McEliece). However, they can be efficient and there is a lot of confidence in the McEliece scheme (and its variants), which has remained secure for 40 years.

- *Multivariate-based*: The security of multivariate based systems is based on the hard problem of solving non-linear multivariate equations over a finite field. They have been more successfully used to build signature schemes ([Ding et al., 2017, Samardjiska et al., 2017]) than PKE. Multivariate signatures have the advantage of being fast and having short signature sizes, however key sizes are large (up to 100kB) and security proofs are lacking, with security estimates solely based on known attacks.

- *Hash-based*: The security of hash-based signatures relies on the collision resistance of the underlying hash function. Examples of hash-based schemes are Merkle [Merkle, 1989] and XMSS [Buchmann et al., 2011]. Their security is well understood, and promising performance and small signature sizes mean hash-based signatures are a promising quantum-safe alternative. One of the challenges is creating practical stateless hash-based schemes. Stateful signature schemes require keeping track of the signing keys to en-

sure they are never reused. SPHINCS$^+$ [Hulsing et al., 2017] is an example of a practical stateless hash-based signature.

- *Isogeny-based*: The most recent approach to quantum-safe cryptography is based on isogenies over elliptic curves [Jao and De Feo, 2011]. Isogeny-based schemes are popular due to their short key sizes, for example only 564 bytes for SIDH [Jao et al., 2017]. However, further cryptanalysis is required to increase confidence in the security margins, as some attacks have been proposed in recent years [Galbraith et al., 2016, Petit, 2017]. The isogeny-based Diffie-Hellman-type scheme CSIDH [Castryck et al., 2018] has undergone enlightening cryptanalysis in [Peikert, 2019]. The original proposal did not consider the cost of a C-sieve attack and the authors of [Peikert, 2019] found that the complexity of key recovery was reduced to only about 2^{16} quantum evaluations using 240 bits of quantumly accessible classical memory when considering quantum C-sieving.

Lattice-based schemes are currently the most popular (12 out of the 26 Round 2 candidates in the NIST process, followed by 7 code-based schemes, see Section 2.3). Advantages of lattice-based schemes over the other families include:

- lattice problems have been shown to have a worst-case to average-case hardness property [Ajtai, 1996], meaning an average instance of the problem is as hard as the hardest instance

- lattices can provide PKE, DSS, and KEM and in Round 2 of the NIST process (discussed in §2.3), are represented in all categories, whereas hash-based approaches can only provide signatures and there are no code-based signatures in Round 2

- lattices have the ability to provide more advanced cryptographic primitives, such as IBE ([Boneh and Boyen, 2004], Chapter 4), Attribute Based Encryption (ABE) [Goyal et al., 2006, Boyen, 2013], FHE [Gentry, 2009], privacy-preserving encryption [Abdallah and Shen, 2016] and Multi-Party Computation (MPC) [Damgård et al., 2012, López-Alt et al., 2012] which have not yet been realised by other types

- performance of lattice-based primitives has been shown to be competitive with existing schemes such as RSA and ECC

The underlying lattice problems the Closest Vector Problem (CVP) and the Shortest Vector Problem (SVP) are assumed to be of Non-deterministic Polynomial-time (NP)-hardness, which means they cannot be solved in polynomial time, [Ajtai, 1996, Micciancio and Goldwasser, 2012] and are resistant to quantum computing attacks. Indeed, many schemes are based on the hardness of approximating the solution of these problems to within polynomial or super-polynomial factors. It is common for cryptographic primitives to base their security on learning problems which have been proven to be at least as hard as the core lattice problems.

Public keys associated with lattice-based systems can be over 15kB [Naehrig et al., 2017], which can restrict performance. However, it is common to add structure to the lattices to reduce this by a square root factor. To date, this is not known to introduce any vulnerability to the security, while allowing great benefits in efficiency. Not only are the key sizes reduced, but the speed of underlying operations can be increased due to this new structure.

There is a requirement for security primitives to keep up with rapid developments in technology as a whole. For instance the range and number of devices connected

to the internet is growing more rapidly than ever before, and as lattices are the only quantum-safe cryptographic family to offer advanced primitives, they can offer adaptable, versatile security solutions for such applications.

Lattices were first used in cryptanalysis [Hinek, 2004] but a seminal paper by Ajtai [1996] led to them being considered for cryptography in 1996. Although as a field it is still relatively young, the main underlying problems have sustained their hardness guarantees. Ongoing cryptanalysis is important but lattices have arguably, along with codes, the most robust security claims. The standards bodies ETSI and NIST have a strong interest in LBC, Google trialled the lattice-based quantum-safe scheme NewHope in its Chrome browser in 2016 [Braithwaite, 2016] and the lattice-based DSS BLISS-B has been integrated into the StrongSwan IPSec implementation[3].

All of these aspects provide the motivation for this thesis. It progresses the research into lattice-based cryptography as a quantum-safe alternative to today's primitives, by demonstrating they can be efficient, and improving confidence in their security, both in terms of cryptanalysis and physical attacks without sacrificing performance. This research also aims to provide the first practical evaluation of advanced lattice-based primitives, some of which have not been previously explored in this regard even in the non-quantum-safe setting.

1.2 Objectives

The central objective of this thesis is to demonstrate the versatility of lattice-based cryptography and show it can be practical for everyday applications without sacrificing security. This is achieved by developing algorithmic improvements and optimal

[3]https://github.com/strongswan/strongswan

software designs to reduce bandwidth consumption and memory usage of the schemes, whilst also considering the underlying security principles (cryptanalysis and physical attacks) and required security levels. More sophisticated cryptographic primitives are created in order to reduce the public key infrastructure overheads of conventional PKE schemes by removing the need for certificate management.

A common application considered throughout this thesis is the IoT. With the expected widespread deployment of smart devices and infrastructures connected to the 5th generation mobile network (5G), there is a requirement for new low-cost yet versatile cryptographic solutions. IoT devices tend to operate in low-power, constrained environments that would struggle to even support today's cryptography and so the requirement for lightweight quantum-safe cryptography is imperative.

In 2016, NIST announced their intentions to standardise quantum-safe cryptography [NIST, 2016]. This has turned into a global effort to build cryptographic primitives in which users can be confident with regards to their security, and that will be usable for the technology of the future. The research in this thesis is strongly related to the NIST candidates Dilithium and Falcon, over half of which were lattice-based in Round 1. The digital signature scheme evaluation in Chapter 3 aims to help progress the understanding of these two candidates, and the advanced cryptographic primitives studied in Chapters 4 and 5 are closely related and serve to illustrate the versatility of LBC, in particular the NIST candidate Falcon.

Physical attack security is a popular area of research in relation to existing cryptography, however the uniqueness of lattice-based cryptography means that research is largely starting from scratch when it comes to this aspect. This thesis considers side-channel attacks against lattice-based schemes and proposes countermeasures to protect against them. All of the implementations presented here have incorporated

constant-time components where possible and attempt to mitigate other timing and fault attacks. This provides a fairer comparison when comparing the NIST candidate schemes, something which was deemed important by NIST from the beginning of the process.

Prior to this research, there were no performance figures for advanced quantum-safe cryptographic primitives to indicate that they would potentially be deployable for low-cost devices. This thesis fills this gap by developing schemes and software designs which demonstrate that quantum-safe algorithms can run in a reasonable amount of time and be sufficiently lightweight for constrained devices. The implementations are developed in portable ANSI C code which make it straightforward to port to a range of devices as required.

1.3 Novel Contributions

The contributions of this thesis are detailed as follows:

1. The first analysis of the physical attack security of the Falcon signature scheme is undertaken [McCarthy et al., 2019]. This includes the first proposed fault attack of this DSS. The attack is simulated in software, together with post-processing of the faulty signatures to derive the secret key. Countermeasures to protect against it are proposed together with an evaluation of the effect on performance of these countermeasures.

2. A comprehensive survey of known attacks on the digital signature schemes Dilithium and BLISS-B and proposed countermeasures for these attacks are presented. An evaluation of the effect on performance of each scheme with its

associated attack countermeasures applied is given and provides a fair comparison of the performance overhead of countermeasures on the schemes.

3. A review of the state of the art with regards to *advanced* quantum-safe cryptographic schemes is presented. At the beginning of this research, only basic lattice-based schemes had been the focus of the research community, however with more sophisticated use cases such as in IoT, it became clear that this gap needed to be filled. The aim of this thesis is to initiate interest in using lattices for such advanced schemes and show they have the potential for standardisation and practical deployment.

4. The first practical software design of a quantum-safe IBE scheme is presented. Algorithmic optimisations such as the use of the Number Theoretic Transform (NTT), CDT sampler and Barrett reduction and implementation concepts such as memory optimisation and efficient sampling techniques are utilised to show the scheme proposed in [Ducas et al., 2014a] can be practical. Parameter recommendations are given and side channel attack resistance is implemented through the use of constant-time components, such as a constant-time CDT Gaussian sampler.

5. The IBE lattice-based scheme is shown to be competitive in comparison to conventional IBE schemes used today in terms of running time and memory consumption. The performance of the lattice-based IBE scheme is compared with the state of the art and shown to be two orders of magnitude faster than conventional pairings-based schemes. This design also outperforms existing quantum-safe implementations of IBE, with an improvement of 180% on the proof of concept non-optimised implementation for user key extraction.

6. A novel, scalable quantum-safe HIBE scheme named Skinny Latte is proposed together with theoretical analysis of its provable and practical security. Parameter recommendations for Skinny Latte are given. The existing Latte HIBE scheme [Campbell and Groves, 2017] is reviewed including a theoretical security evaluation and the first published software design and performance results of this scheme are given. The two variants are compared and evaluated.

1.4 Organisation of Thesis

This thesis is organised as follows. Chapter 1 has described the motivation and objectives of the research, and has outlined the novel contributions.

Chapter 2 presents the technical knowledge the reader requires for this thesis. This includes a comprehensive explanation of lattice-based cryptography. It covers the mathematical knowledge needed to understand algorithms explored in this thesis and details the cryptographic concepts researched. Current efforts in post-quantum cryptography within the research community are described, in particular the NIST standardisation process. High-level descriptions of both basic and advanced cryptographic primitives are given here.

Chapter 3 is a software analysis of three lattice-based digital signature schemes, together with an analysis of physical attacks and relevant countermeasures. The main focus here is the Falcon signature scheme, and the first proposed fault attack targeting its sampler. The attack targets the recursive call of the sampler, and suggests multiple methods of terminating the recursion early in order to reveal the secret key. Software simulation of the attack is performed, followed by the post-processing and lattice reduction steps. Countermeasures for this attack and for other known timing attacks

are proposed and incorporated into the SCA-resistant ANSI C-based software design. Two other lattice-based signature schemes are also considered, Dilithium and BLISS-B, and countermeasures are proposed for reported physical attacks against them. A thorough performance benchmarking of the three schemes with these countermeasures in place is provided, with the intention of providing a fair comparison of the schemes.

Chapter 4 presents a practical implementation of the lattice-based IBE scheme DLP-IBE. These algorithms are closely related to the Falcon signature scheme, as they share the same underlying polynomial ring and key generation components. This chapter provides the first performance results of a quantum-safe IBE scheme, and shows it is practical for constrained devices, such as those used in the IoT. These results are achieved through algorithmic and design optimisations.

Chapter 5 extends these schemes to a Hierarchical Identity Based Encryption (HIBE) scheme. The HIBE scheme Latte [Campbell and Groves, 2017] was proposed in 2017 and was based on the DLP-IBE scheme. However, it is shown to be unscalable due to the increase in lattice dimension during delegation. To address this, a novel fixed-dimension HIBE scheme named "Skinny Latte" is constructed. Its security is proved in the random oracle model and its practical security is also analysed in order to derive suitable parameters. Initial re†sults show that it is suitable for real-world applications, e.g. for instantaneous communication between low power devices.

The thesis is concluded in Chapter 6, where the main outcomes of each chapter are revisited and future related research directions are proposed.

1.5 Collaborative contributions

The research presented in this thesis was carried out in collaboration with academic, government and industry partners. The software designs of the lattice-based signature (Chapter 3) and IBE schemes (Chapter 4) contributed to the H2020 SAFEcrypto project, as discussed in §2.7.1. The DLP-IBE scheme design in Chapter 4 formed a basis for the IBE demonstrator, an outcome of SAFEcrypto, and provided the UK National Cyber Security Centre (NCSC) (with whom the author completed an 8-week placement) with confidence and motivation to consider this scheme as a basis for their HIBE construction Latte. The research in Chapter 5 was carried out in consultation with Thomas Prest and researchers at NCSC. The implementation work in Chapter 5 is joint work with Thales UK. The Latte HIBE scheme is currently being considered for standardisation by ETSI and the author of this thesis has contributed to the technical report, with the performance results presented in this thesis to be included.

CHAPTER 2

Background and Theory

2.1 Introduction

In this chapter, descriptions and definitions of the key concepts that underpin the thesis contributions are given. Standard cryptographic primitives, namely PKE, DSS and KEM are introduced. Lattices are formally defined alongside associated hard problems underpinning the security of LBC. The state of the art of practical constructions is covered, alongside ongoing efforts within the community towards standardisation and deployment of Quantum-Safe Cryptography (QSC). The chapter concludes with acknowledged open problems surrounding LBC, a portion of which this thesis aims to address.

Lattice-based schemes are the most popular construction for quantum-safe cryptographic primitives and they account for almost half of proposals in Round 2 of the NIST standardisation process [NIST, 2016] as discussed in §2.3. The reasons for this include the diversity of primitives they support including advanced cryptographic primitives, as well as their performance relative to the other candidates. This chapter introduces

lattices from a mathematical perspective, and describes how the computationally hard problems associated with lattices can be extended to cryptographic primitives. Also introduced are the different types of lattices, which ultimately offer a trade-off between efficiency and security in the end primitive.

2.2 Preliminaries

2.2.1 Mathematics and Notation

Throughout this thesis, vectors will be denoted by lower case bold letters $\mathbf{v} = (v_0, v_1, ..., v_N)$, with coefficients v_i. The ith scalar component of $\mathbf{v}$ is denoted by v_i. Occasionally, to maintain consistency with existing literature, vectors may also be denoted by V. Matrices are denoted by upper case bold letters $\mathbf{M} = (\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, ..., \mathbf{v}^{(N)})$. N is a power-of-two integer, and q is a prime modulus.

Where the schemes are based on ideal lattices, the computations are performed over a polynomial ring. A polynomial ring over the integers is defined as $\mathcal{R}_q = \mathbb{Z}_q[x] \big/ \langle f(x) \rangle$, where deg $f(x) = N$. This is the ring whose elements can be represented by polynomials of degree $N-1$ over a modulus q with coefficients in the range $\big[-\frac{q-1}{2}, \frac{q-1}{2}\big]$. An element $\mathbf{v} \mod q$ has been reduced with respect to this modulus. Ideal lattice vectors can be considered as ring polynomials. The identity matrix of dimension N is denoted $\mathbf{I_N}$ and the null matrix of dimension N is denoted $\mathbf{0_N}$. The dot product of two vectors $\mathbf{f}, \mathbf{g}$ is denoted $\langle \mathbf{f}, \mathbf{g} \rangle$ and the multiplication operation of scalars and/or polynomials is represented by $\mathbf{f} \cdot \mathbf{g}$. The notations $\lceil v \rceil$, $\lfloor v \rfloor$ and $\lfloor v \rceil$ are used for rounding up, down or to the nearest integer respectively, meaning coefficient-wise

when applied to a vector or polynomial. The anti-circulant matrix of polynomial $\mathbf{f} = (f_1, f_2, ..., f_N)$ is denoted

$$\mathcal{A}(\mathbf{f}) = \begin{pmatrix} f_0 & f_1 & \cdots & f_{N-1} \\ -f_{n-1} & f_0 & \cdots & f_{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ -f_1 & -f_2 & \cdots & f_0 \end{pmatrix}$$

For a NTRU lattice defined by polynomials $\mathbf{h} = \mathbf{g} \cdot \mathbf{f^{-1}}$, the public matrix is

$$\mathbf{B}_{pub} = \begin{pmatrix} -\mathcal{A}(\mathbf{h}) & \mathbf{I_N} \\ q\mathbf{I_N} & \mathbf{0_N} \end{pmatrix}$$

and secret matrix is

$$\mathbf{B}_{priv} = \begin{pmatrix} \mathcal{A}(\mathbf{g}) & -\mathcal{A}(\mathbf{f}) \\ \mathcal{A}(G) & -\mathcal{A}(F) \end{pmatrix}$$

where F, G are the NTRU polynomials (see §2.4.2) associated with $\mathbf{f}, \mathbf{g}$. Note that for conciseness, an NTRU-type basis will simply be denoted by the polynomials, so that

$$\begin{pmatrix} \mathbf{g} & -\mathbf{f} \\ G & -F \end{pmatrix} := \begin{pmatrix} \mathcal{A}(\mathbf{g}) & -\mathcal{A}(\mathbf{f}) \\ \mathcal{A}(G) & -\mathcal{A}(F) \end{pmatrix}$$

All logarithms used are to the base 2, i.e. $\log_2(x)$ will simply be denoted as $\log(x)$.The natural logarithm function is denoted ln. The Gram-Schmidt orthogonalisation of a vector $\mathbf{v}$ will be denoted as $\tilde{\mathbf{v}}$. The Euclidean norm of a vector $\mathbf{v}$ will be denoted $||\mathbf{v}||$

and the Gram-Schmidt norm of a matrix $\mathbf{B}$ will be denoted $||\mathbf{B}||_{GS}$. Where an element is sampled uniformly at random from a distribution X, this will be denoted $x \leftarrow X$. The expectation of a random variable X is denoted $E[X]$. The Gaussian distribution over the vector space $\mathbb{R}^N$ with centre 0 and standard deviation σ is denoted $\mathcal{D}_{N,\sigma}$ and is further defined in §2.8.

The projection $Proj(\mathbf{x}, \mathbf{y})$ of a vector $\mathbf{x}$ onto a vector $\mathbf{y}$ is the orthogonal projection of $\mathbf{x}$ onto a straight line parallel to $\mathbf{y}$. It can be considered as the "amount" of $\mathbf{x}$ which lies in the same direction as $\mathbf{y}$. The span $Span(\mathbf{Y})$ of a set of vectors $\{\mathbf{y_0}, \mathbf{y_1}, ..., \mathbf{y_n}\}$ is the n-dimensional space containing vectors which can be expressed as a linear combination of vectors in $\mathbf{Y}$.

2.2.2 Cryptographic and Security Concepts

There are many different types of cryptographic primitives covered in this thesis. The thesis begins with basic primitives: public key encryption (PKE), digital signature schemes (DSSs) and key encapsulation mechanisms (KEMs). It then moves onto advanced schemes, namely identity based encryption (IBE) and its hierarchical extension. All of these schemes are described in more detail with respect to lattices in Section 2.4. The aim of encryption is to preserve the confidentiality and integrity of the data and digital signature schemes additionally ensure authenticity and non-repudiation. KEMs are primarily used to encrypt and transport a symmetric key for encryption.

Provable security refers to the security of a cryptosystem A which can be theoretically proved to depend on the hardness of an underlying mathematical problem B, for example a lattice-based problem. This implies that breaking the cryptosystem is at least as hard as solving the underlying problem B, therefore, if it is assumed that the

problem B is too hard to solve within a feasible amount of time, the cryptosystem is secure. Security proofs may assume the existence of a Random Oracle Model (ROM) which is a theoretical black-box outputting a truly random, uniformly distributed value. For a given input, the output value will remain consistent. A ROM may be susceptible to attacks by a quantum computer. Therefore, the stronger concept of Quantum Random Oracle Model (Q-ROM) was introduced, that is one which is secure even if the attacker is equipped with a quantum computer. In reality, (Q-)ROMs are replaced by a hash function whose outputs behave as pseudo-random. In contrast to a proof in the ROM, a security reduction in the standard model does not require as strict conditions on the properties of the hash function. A scheme which is secure on the ROM model is only secure in the standard model if the attacker is not able to exploit any vulnerabilities in the hash function. The security reduction may be tight or loose. If solving the hard problem B takes much longer, with a lower success probability than breaking A, then the reduction is considered loose. In the case of a loose security reduction, the parameters chosen for A will need to be larger than those for the equivalent problem B, as it is only known that A takes at least as much as *a fraction* of the effort required to break B.

Knowing parameters which make the underlying problem hard allow the user to choose suitable security parameters for the cryptosystem. However, this usually renders the scheme very slow and inefficient. Therefore, most constructions use known attacks to evaluate the practical security of the scheme and assign practical parameters. A further consideration is that of Side-Channel Analysis (SCA) or physical attacks. Although the mathematics of the scheme may be secure, when executed in a real-life application, the implementation may introduce a vulnerability. Physical attacks are discussed further in Chapter 3, in relation to lattice-based DSSs.

2.3 NIST Standardisation Process

2.3.1 Introduction

In 2016, NIST announced a post-quantum cryptographic standardisation process [NIST, 2016]. This invited the research community to submit candidate proposals, which would be publicly assessed throughout several iterations, with a view to choosing several for standardisation by 2022 [Moody, 2019]. An online forum[1] was set up to encourage collaborative discussion. NIST aims to choose a diverse range of primitives which can be practical and realised with confidence in security.

2.3.2 Submissions

There were a total of 82 submissions to the process, in the form of PKE, DSS and KEM schemes. The requirements and evaluation criteria specified five security levels, as shown in Table 2.1. Those in red are considered for high-security use-cases, e.g. government, military, long-term data storage etc. NIST included properties such as drop-in replacements, perfect forward secrecy, resistance to side-channel attacks, simplicity and flexibility and misuse resistance as important in considering which submissions would be successful. Performance on various classical platforms would also be evaluated. The submissions spanned the five main quantum-safe families as described in Section 1.1, along with some miscellaneous and classical cryptography bases. Lattice-based primitives accounted for over a third of the 69 proposals in the first round, and 12 out of 26 proposals in the second round. Figure 2.1 shows the distribution of the candidates in each round according to type.

[1] https://groups.google.com/a/list.nist.gov/forum/#!forum/pqc-forum

Table 2.1 The security levels specified by NIST in their call for submissions

Security Level	Description: at least as difficult to break as...
1	AES128
2	SHA256
3	AES192
4	SHA384
5	AES256

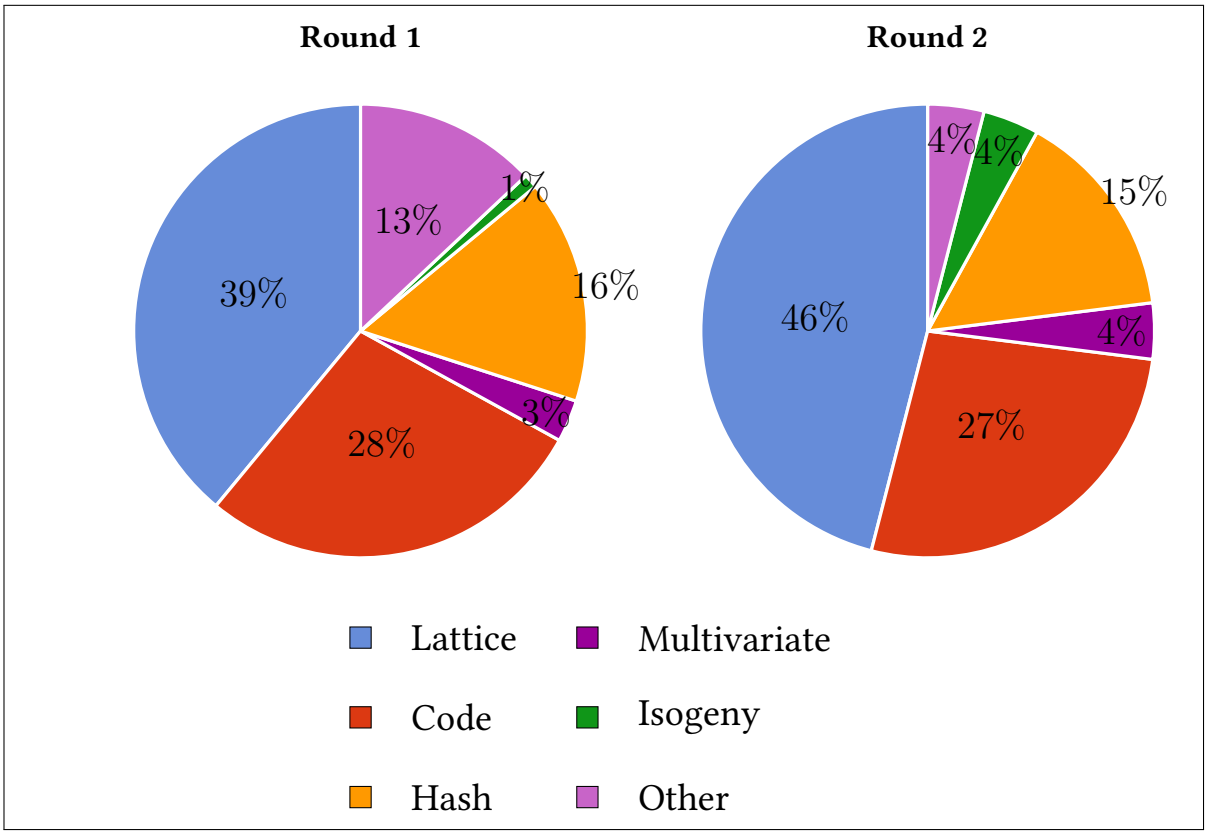


Fig. 2.1 The distribution of NIST candidate types

2.3.3 Round 1

A total of 69 of the original 82 submissions made it through to Round 1 following initial inspection, which involved checking they conformed to the submission criteria. Within 3 weeks of publication, 12 of the 69 schemes were significantly attacked or broken. Of

these, 5 schemes (RankSign [Aragon et al., 2017b], Edon-K [Gligoroski and Gjosteen, 2017], SRTPI [Peretz and Granot, 2017] , RVB [Roellgen and Brands, 2017], HK17 [Hecht and Kamlofsky, 2017]) were considered beyond repair and were therefore withdrawn. Others (e.g. DAGS [Banegas et al., 2017], pqNTRUSign [Zhang et al., 2017b], Walnut DSA [Atkins et al., 2017]) were attacked and patched. Scrutinisation of the first round candidates took place from publication in December 2017 until January 2019. The first NIST PQC Standardisation Workshop was held in Fort Lauderdale in April 2018[2] and encouraged cryptanalysis and performance benchmarking of the schemes from the community as a whole. Open-source software libraries focussing on NIST candidates include libSAFEcrypto [SAFEcrypto, 2018], OpenQuantumSafe [Stebila and Mosca, 2016] and LatticeCrypto [Microsoft, 2016]. A comparison of these is given in Table 2.9, later in this chapter. There is additionally a benchmarking package SUPERCOP [VAMPIRELab, 2019] which allows existing implementations of schemes by various authors to be dropped in and fairly compared by using the same underlying components (hash functions, random number generators etc) and platform, which may aid NIST in deciding between algorithms based on performance/efficiency trade-offs.

2.3.4 Round 2

NIST used factors such as a candidate's track record in security vulnerabilities and potential feasibility of the schemes in useful applications to narrow down the selection for Round 2. They reported a lack of full confidence in the security of some schemes [Moody, 2019] such as WalnutDSA [Atkins et al., 2017] and Compact-LWE [Liu et al., 2017], and others such as PQRSA [Bernstein et al., 2017] and DualModeMS

[2]https://csrc.nist.gov/events/2018/first-pqc-standardization-conference

Table 2.2 Schemes which merged to create new Round 2 candidates

Original schemes	Merged scheme
Lake Locker Ouroboros-R	ROLLO
LEDApkc LEDAkem	LEDAcrypt
NTRUencrypt NTRU-HRSS-KEM	NTRU
Hila5 Round2	Round5

[Faugere et al., 2017] were deemed too inefficient and did not proceed to the second round. NIST aimed to maintain diversity of schemes for Round 2, and to facilitate this process, encouraged teams with similar schemes to merge. This created 4 new schemes outlined in Table 2.2. All authors were allowed to tweak and update their scheme specifications following announcement of the Round 2 candidates. The second NIST PQC Standardisation Workshop was co-located with Crypto[3] in August 2019.

2.3.5 Lattice-based NIST candidates

The lattice-based candidates in the NIST process are shown in Table 2.3. There are submissions based on standard [Naehrig et al., 2017], ideal [Bindel et al., 2017], NTRU [Prest et al., 2017] and module [Lyubashevsky et al., 2017, Schwabe et al., 2017] lattices. Lattices are the only family to have submissions in all cryptographic classes (PKE, KEM, DSS) in Round 2. They are a popular primitive due to confidence in security (well-studied, no major breaks, cryptanalysis reaching a lower bound [Lyubashevsky, 2018]), empirical efficiency and potential for real-world deployment. However, well-recognised challenges include vulnerability to SCA (in particular the

[3]https://crypto.iacr.org/2019/

Table 2.3 The lattice-based schemes involved in the NIST standardisation process Round 1 (Round 2: **selected**/*merged*)

PKE/KEM		***Signature***
Compact LWE	Lizard	**Dilithium**
Ding Key Exchange	Lotus	DRS
(R) Emblem	Mersenne-756839	**Falcon**
FrodoKEM	**NewHope**	pqNTRUsign
Giophantus	**NTRUPrime**	**qTESLA**
HILA5	*NTRU-HRSS-KEM*	
KCL	*NTRUEncrypt*	
KINDI	Odd Manhattan	
Kyber	*Round2*	
LAC	**Saber**	
Lepton	**ThreeBears**	
Lima	Titanium	

Gaussian sampling stage) and their large key sizes, particularly for unstructured lattices. There are still strong security assumptions regarding the use of structured lattices.

2.3.6 Next steps to standardisation

Round 2 is expected to continue for a further 6-12 months following the August workshop, and there will be a potential Round 3 stage. NIST anticipate draft standards by 2022 [Moody, 2019].

2.4 Lattice-based Cryptography

2.4.1 Lattices

Lattices are discrete additive subgroups of $\mathbb{R}^n$, defined by a collection of vectors $\mathbf{b}_i$ known as its basis $\mathbf{B}$. Every vector in the lattice can be expressed as a linear

combination of the basis vectors, with integer coefficients. Formally, a lattice Λ with basis $\mathbf{B}$ and dimension N can be defined as:

$$\Lambda(\mathbf{B}) = \left\{ \sum_i a_i b_i \mid a_i \in \mathbb{Z}, b_i \in \mathbf{B}, 1 \leq i \leq N \right\}$$

Informally, lattices can be thought of as an infinite arrangement of regularly spaced points. Figure 2.2 shows an example of a 2-dimensional lattice.

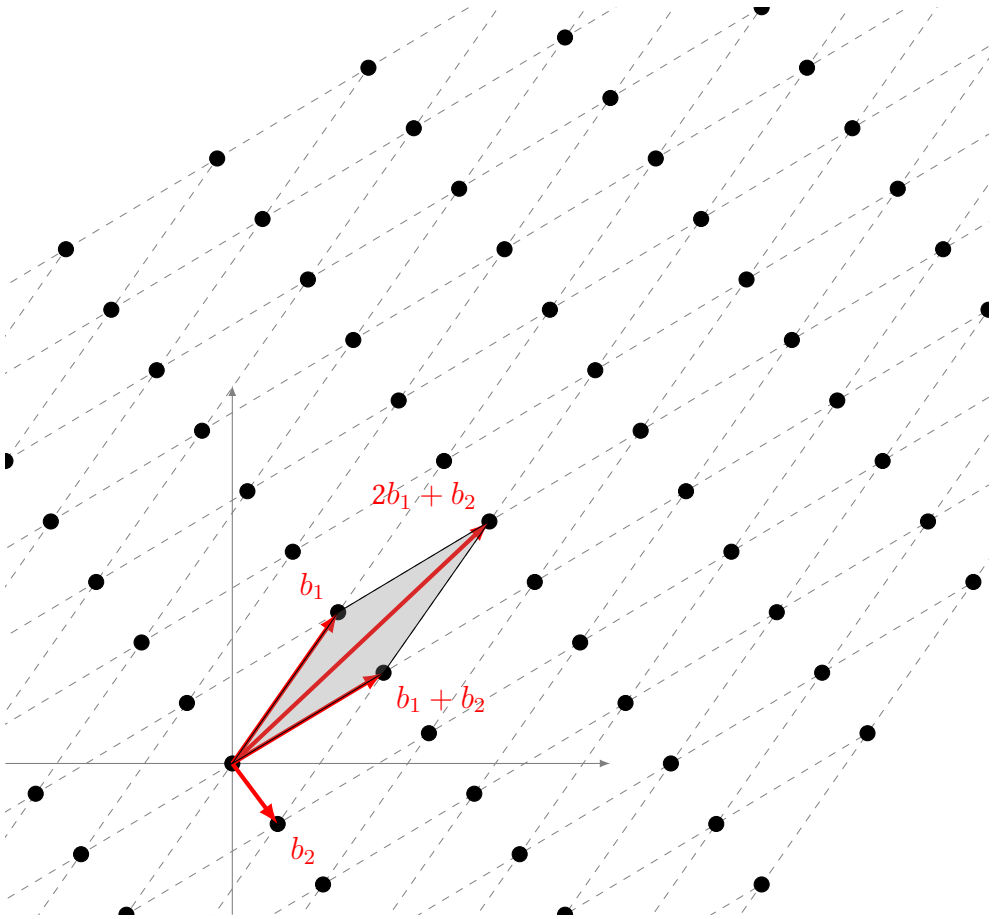


Fig. 2.2 A 2-dimensional lattice [Tourloupis, 2012]

The smoothing parameter $\eta_\epsilon(\Lambda)$ of a lattice is the smallest s such that

$$\rho_{0,1/s\sqrt{2\pi}}\big(\Lambda^*/0\big) \leq \epsilon$$

where $\rho_{0,1/s\sqrt{2\pi}}$ is the Gaussian distribution with centre 0 and standard deviation $1/s\sqrt{2\pi}$, which will be formally introduced in §2.8. The bounds for smoothing parameters of $\mathbb{Z}$ and $\mathbb{Z}^N$ used throughout this thesis are

$$\eta_\epsilon(\mathbb{Z}) \leq \frac{1}{\pi}\sqrt{\frac{1}{2}ln\left(2\Big(1+\frac{1}{\epsilon}\Big)\right)}$$

$$\eta_\epsilon(\mathbb{Z}^N) \leq \frac{1}{\pi}\sqrt{\frac{1}{2}ln\left(2N\Big(1+\frac{1}{\epsilon}\Big)\right)}$$

These definitions follow that of Prest [2017], which scales the smoothing parameter by a factor of $\sqrt{2\pi}$ for conciseness.

Whilst standard lattices are defined by the matrix of basis vectors, **ideal lattices**, introduced by Lyubashevsky et al. [2010a] can be defined by a single vector, as the rest of the matrix can be generated by cyclically shifting the single basis vector. This improves efficiency of the lattice based schemes by reducing both storage and computations by a factor of N. To date, this extra structure has not been shown to introduce any security vulnerabilities. Ideal lattices can be modelled as a polynomial ring $\mathcal{R}$ due to the canonical mapping.

NTRU lattices [Hoffstein et al., 1998b] are a specific type of ideal lattice whose private basis is defined by two short polynomials $\mathbf{f}, \mathbf{g}$, and public basis is defined

by the polynomial $\mathbf{h} = \mathbf{g} \cdot \mathbf{f}^{-1}$. The underlying polynomial ring in this instance is $\mathcal{R} = \mathbb{Z}[x]/\langle x^N - 1 \rangle$.

Module lattices, as described by Langlois and Stehlé [2014], can be considered as somewhere in-between standard and ideal lattices. The coefficients can be considered as polynomials. Just as an ideal lattice is isomorphic to a polynomial ring-ideal, a module lattice is isomorphic to a module of the ring. A ring-module can be considered as a vector space over a ring, so the coefficients of the vectors are ring elements, in this case polynomials. Table 2.4 summarises key points about, and examples of, the three types of lattices.

Table 2.4 The three types of lattices used in cryptography

Lattice type	**Basis structure**	**Defining features**	**Scheme examples**
Standard	Uniform	Security understood	Round5 variant [Garcia-Morchon et al., 2019] FrodoKEM [Naehrig et al., 2017]
Module	Coefficients can be considered as polynomials	Security “happy medium"	Dilithium [Lyubashevsky et al., 2017] Three Bears [Hamburg, 2017]
Ideal	Cyclic	Strong security assumption Compact basis Fast polynomial multiplication	NTRU-based [Zhang et al., 2017b,a] NewHope [Poppelmann et al., 2017] Falcon [Prest et al., 2017]

A q-**ary lattice** is one which satisfies $q\mathbb{Z}^N \subseteq \Lambda \subseteq \mathbb{Z}^N$. For q-ary lattices, it is known that the Learning With Errors problem (LWE) and the Short Integer Solution problem (SIS) are as hard as the worst-case lattice problems and so these are often used in LBC. The **dual** of a lattice Λ is the set of all points whose inner product with any of the points in Λ is an integer. A lattice can be defined in an alternative way. If a lattice is defined as:

$$\Lambda_q(\mathbf{B}) = \{y \in \mathbb{Z}^N : y = \mathbf{B}^T s \mod q \text{ for some } s \in \mathbb{Z}^N\}$$

then its dual can be defined as

$$\Lambda_q^{\perp}(\mathbf{B}) = \{y \in \mathbb{Z}^N : \mathbf{B}y = 0 \mod q\}$$

2.4.2 More on NTRU Lattices

Adding structure and pattern to the lattice basis improves the practicality of the cryptographic primitive. The memory and efficiency can be improved by reducing the amount of basis information that needs to be stored or transported to recover the lattice. The DLP-IBE scheme [Ducas et al., 2014a] examined in Chapter 4 uses NTRU lattices for this reason as NTRU lattice bases have a convolutional, modular structure. The trapdoors in this scheme are the polynomials $\mathbf{f}, \mathbf{g}$, which allow the user to generate a "nice" basis $\mathbf{B}_{\text{priv}}$ whilst the public only have access to the lattice through a "bad" basis $\mathbf{B}_{\text{pub}}$ defined by polynomial $\mathbf{h}$.

$$\mathbf{B}_{\text{pub}} = \begin{pmatrix} h_0 & h_1 & \dots & h_{N-1} & 1 & 0 & \dots & 0 \\ -h_{n-1} & h_0 & \dots & h_{N-2} & 0 & 1 & \dots & 0 \\ \vdots & \ddots & & \vdots & \vdots & \ddots & & \vdots \\ -h_1 & -h_2 & \dots & h_0 & 0 & 0 & \dots & 1 \\ q & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & q & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \ddots & & \vdots & \vdots & \ddots & & \vdots \\ 0 & 0 & \dots & q & 0 & 0 & \dots & 0 \end{pmatrix} = \begin{pmatrix} -\mathcal{A}(\mathbf{h}) & \mathbf{I}_N \\ \mathbf{I}_q & \mathbf{0}_n \end{pmatrix}$$

$$\mathbf{B}_{\text{priv}} = \begin{pmatrix} \mathcal{A}(\mathbf{g}) & -\mathcal{A}(\mathbf{f}) \\ \mathcal{A}(G) & -\mathcal{A}(F) \end{pmatrix},$$

where each $\mathcal{A}(\cdot)$ is an anti-circulant matrix as defined in §2.4.1 dependent on a polynomial $\mathbf{f}, \mathbf{g}, F$ and G which satisfy the NTRU equation $\mathbf{f}G - \mathbf{g}F = q \mod x^N - 1$.

The NTRU lattice assumption is that it is a hard problem to recover polynomials $\mathbf{f}, \mathbf{g}$ from $\mathbf{h}$, where $\mathbf{h} = \mathbf{g}/\mathbf{f}$, i.e. it is hard to obtain $\mathbf{B}_{\text{priv}}$ from $\mathbf{B}_{\text{pub}}$. The original NTRU system used the polynomial ring $\mathbb{Z}_q[x]/\langle x^N - 1\rangle$, however the schemes focussed on in this thesis use the NTRU distribution over the polynomial ring $\mathbb{Z}_q[x]/\langle x^N + 1\rangle$, as proposed by Stehlé and Steinfeld [2011]. This choice of ring allows the schemes to be provably secure under the assumption that R-LWE is hard to solve in the worst-case for this family of ideal lattices. However, there is a gap between theoretic security and practise. Choosing practical parameters requires assessing computational hardness, and reducing the variance of the error distribution to a level where it is practical but still secure.

To generate an NTRU lattice basis pair, first $\mathbf{f}$ and $\mathbf{g}$ are sampled according to a Gaussian distribution over the polynomial ring. This immediately determines the public key $\mathbf{h} = \mathbf{g} \cdot \mathbf{f}^{-1}$. Then the NTRU equation must be solved to find F and G. The classical approach to solving the NTRU equation, given in Hoffstein et al. [2003], consists of evoking a Greatest Common Divisor (GCD) solver over both polynomials

and integers, whilst keeping track of the intermediate coefficients. However, the resulting coefficient of F and G will be very large, and so much be reduced by multiples of $\mathbf{f}$ and $\mathbf{g}$ using Babai's reduction method. Additionally, it can be noted that the choice of F and G are not unique, but it is sufficient for them to satisfy the NTRU equation and be small.

Solving the NTRU equation is recognised bottleneck of the NTRU key generation algorithm. This is largely due to the large coefficients of the intermediate F and G and it may use up to 3.3MB per function call. Efforts towards accelerating this component have been presented in [Prest et al., 2017, Pornin and Prest, 2019].

2.4.3 Hard Lattice Problems

Lattice-based cryptographic primitives base their security on computationally hard problems, of which the most common will now be defined.

- The shortest vector problem (SVP) states that given a lattice basis $\mathbf{B}$, find the shortest vector $\mathbf{v}$ in that lattice, i.e. with $||\mathbf{v}|| = \Lambda_1(\mathbf{B})$. Minkowski's first constant, which is the length of the shortest vector within the lattice, is denoted Λ_1.

- The approximate decision version of the Shortest Vector Problem (GapSVP)$_\gamma$ is where, given a lattice basis $\mathbf{B}$ and a positive integer d, distinguish between the cases $\Lambda_1(\Lambda(\mathbf{B})) \leq d$ and $\Lambda_1(\Lambda(\mathbf{B})) > \gamma \cdot d$.

- The unique Shortest Vector Problem (uSVP)$_\gamma$ states that given a lattice basis $\mathbf{B}$, find the shortest vector $\mathbf{v}$ in that lattice, i.e. with $||\mathbf{v}|| = \Lambda_1(\mathbf{B})$, given that $\Lambda_2 \geq \gamma \cdot \Lambda_1$, for some $1 \geq \gamma \geq poly(N)$.

- The Hermite Shortest Vector Problem (HSVP)$_\gamma$ states that given a lattice basis $\mathbf{B}$, find the shortest vector $\mathbf{v}$ in that lattice, i.e. with $0 \leq ||\mathbf{v}|| = \gamma \cdot det(\Lambda)^{1/d} = \Lambda_1$, where $det(\Lambda)$ is the determinant of the lattice.

- The closest vector problem (CVP) states that given a lattice basis $\mathbf{B}$ and target vector $\mathbf{t}$, find $\mathbf{v} \in \Lambda(\mathbf{B})$ such that $dist(\mathbf{v}, \mathbf{t})$ is minimum.

- The approximate decision version of the Closest Vector Problem (GapCVP)$_\gamma$ states that given a lattice basis $\mathbf{B}$, target vector $\mathbf{t}$ and a positive integer d, distinguish between the cases $dist(\mathbf{t}, \Lambda(\mathbf{B})) \leq d$ and $dist(\mathbf{t}, \Lambda(\mathbf{B})) > \gamma \cdot d$.

- The Shortest Independent Vector Problem (SIVP) states that given a lattice basis $\mathbf{B}$, find n linearly independent vectors $(s_1, s_2, ..., s_n)$ such that $||s_i||$ is minimised for each i.

- The Smallest Basis Problem (SBP)$_\gamma$ states that given a lattice basis $\mathbf{B}$, find a basis $\mathbf{B}'$ such that $\Lambda(\mathbf{B}) = \Lambda(\mathbf{B}')$ and $||\mathbf{B}'||_{GS}$ is minimised.

- Bounded Distance Decoding (BDD) states that given a lattice basis $\mathbf{B}$, a positive integer d and a target vector $\mathbf{t}$ such that $dist(\mathbf{t}, \Lambda(\mathbf{B})) \leq d \cdot \Lambda_1(\Lambda)$, find a $\mathbf{y}$ such that $dist(\mathbf{y}, \mathbf{t}) = dist(\mathbf{t}, \Lambda(\mathbf{B}))$.

Two important learning problems, which can be considered as lattice problems, are now defined.

The **Learning With Errors** (LWE) problem [Regev, 2005, 2009] is as follows:

Let n, q and m be positive integers, χ be a probability distribution on the set of integers modulo q, $\mathbb{Z}_q$, and s be a secret vector in the corresponding vector field $\mathbb{Z}_q^n$. Denote $\mathcal{A}_{\mathbf{s},\chi}$ to be the probability distribution on $\mathbb{Z}_q^n \times \mathbb{Z}_q$ by choosing $\mathbf{a^i} \in \mathbb{Z}_q^n$ (where $i \in \mathbb{Z}$, $1 \leq i \leq m$) uniformly at random and error element $e_i \in \mathbb{Z}_q$ according to χ, and returning:

$$(\mathbf{a^i}, b_i) = (\mathbf{a^i}, \langle \mathbf{a^i}, s \rangle + e_i).$$

This can be expressed as:

$$\mathbf{As} + \mathbf{e} = \mathbf{b}.$$

The **Search-LWE** states given $(\mathbf{A}, \mathbf{b})$, find $\mathbf{s}$. The **Decision-LWE** problem is that of distinguishing $\mathcal{A}_{\mathbf{s},\chi}$ from a uniform distribution.

The Search-LWE Problem can be thought of as a system of m noisy equations with n unknowns. If there was no noise, it could be easily solved by Gaussian elimination. The fact that there is noise, and that this noise comes from a Gaussian distribution, adds complexity to the problem. Alternatively, it can be considered as a case of the BDD problem over lattices. The LWE problem over ideal lattices is called the Ring variant of the Learning With Errors (LWE) problem (R-LWE) [Lyubashevsky et al., 2010b].

The **Short Integer Solution** (SIS$_q$) problem [Ajtai, 1996] states that given a matrix $\mathbf{A} \mod q$ and integer $\mathbf{d} \leq q$, find $\mathbf{y} \in \mathbb{Z}^n$ such that $\mathbf{Ay} = 0 \mod q$ and $||\mathbf{y}|| \leq d$.

Similarly, the SIS problem over ideal lattices is called the Ring variant of the Short Integer Solution (SIS) problem (R-SIS), introduced by Lyubashevsky et al. [2010b].

The hard lattice problems are linked by reductions. If problem A can be reduced to problem B, that means that a solution for problem B will automatically provide a solution to problem A, i.e. problem B is *at least* as hard a problem A. The premise behind the lattice-based problems is that if it can be assumed that SVP/CVP are NP-hard, and a new problem is constructed to which SVP/CVP can be reduced, then this new problem can also be assumed to be NP-hard. Figure 2.3 shows reductions between common lattice problems. An arrow from A to B indicates that A can be reduced to B. An unfilled double arrow head indicates a quantum reduction, which means a quantum algorithm is utilised in the reduction. CVP was proven to be NP-hard to approximate within any constant factor in [Arora et al., 1997, Dinur et al., 1998] and SVP was reduced to CVP in [Henk, 1997, Goldreich et al., 1999]. Therefore these two problems are the core of most lattice-based cryptographic primitives. LWE can be shown to be as hard as GapSVP. The ring variants of various schemes are omitted from the diagram, but these can each be trivially reduced to its standard version.

2.4.4 Basic Lattice-based Primitives

In this section, basic cryptographic primitives are outlined and lattice-based schemes discussed.

Public Key Encryption (PKE)

PKE allows both sender and receiver to exchange messages without sharing a common key. Currently, this is generally done with RSA or ECC. The receiver has access to a public key k_{pub} which they can use to encrypt the message, and only the sender

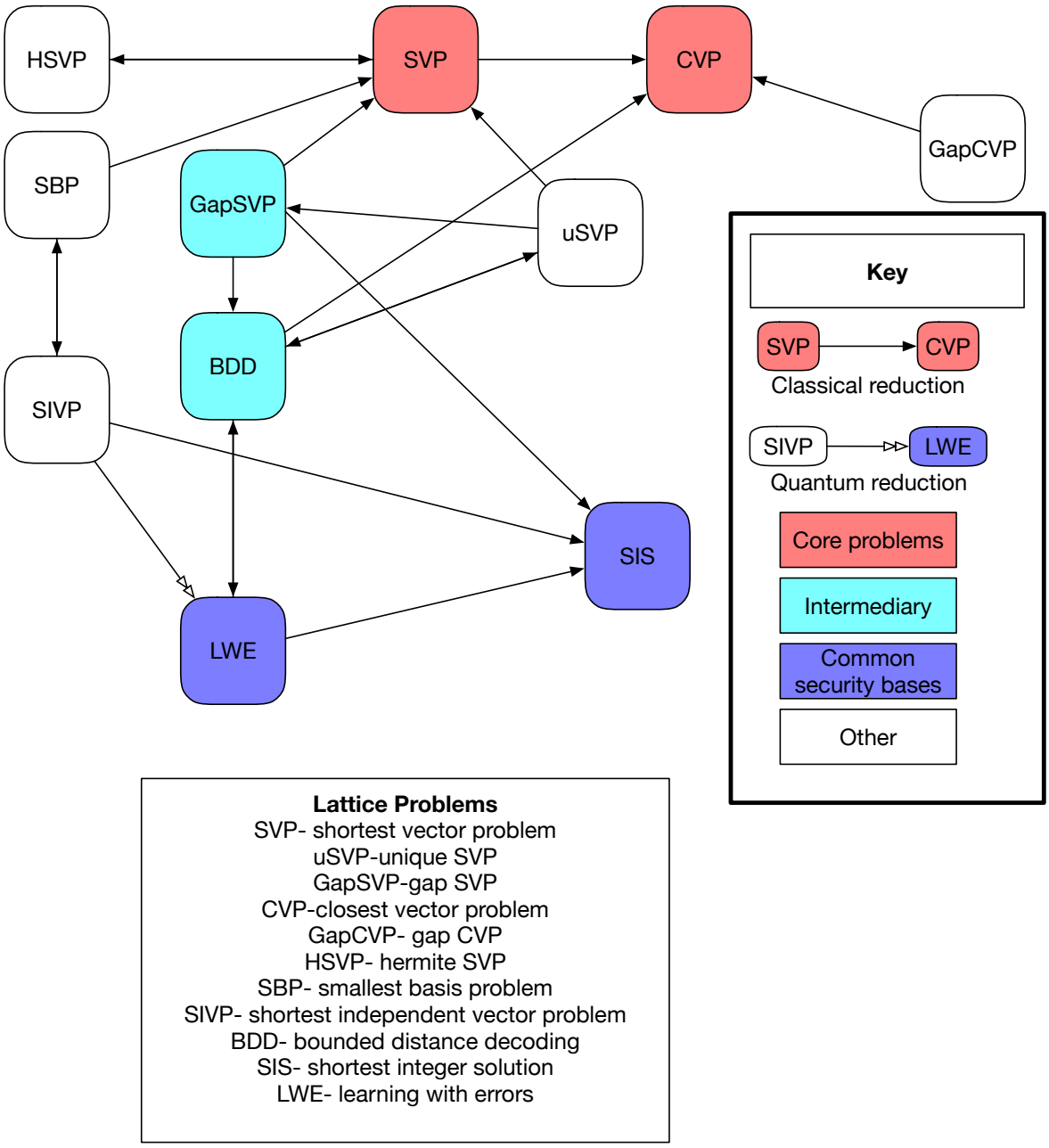


Fig. 2.3 Reductions between lattice problems

knows the private decryption key k_{priv}. These keys are related so that when a message $m \in M$ is encrypted using the public key to give a ciphertext $c \in C$, $Enc_{k_{pub}} : M \to C$ and then decrypted using the private key, $Dec_{k_{priv}} : C \to M$, the property

$$Dec_{k_{priv}}\left(Enc_{k_{pub}}(m)\right) = m$$

is satisfied. It can formally be defined as a system with three algorithms $\{KeyGen, Enc, Dec\}$ satisfying the properties:

- $KeyGen(*)$ outputs a pair of keys (k_{pub}, k_{priv}); the public and private keys respectively
- $Enc(k_{pub}, m)$ encrypts a message m under public key k_{pub}.
- $Dec(k_{priv}, c)$ decrypts a ciphertext c under secret key k_{priv}.

This is visualised in Figure 2.4. The role of encryption is to provide confidentiality of the data, and for this to be perfectly secret no information about the plaintext should be inferred from the ciphertext without the key. Security notions are used to describe the security of a PKE scheme by stating information available to the attacker, and information they are able to derive. The most common examples of these are given in Table 2.5. They can be defined as follows:

- chosen-plaintext attacks (CPA) assume the attacker can choose plaintexts to be encrypted, and receive the corresponding ciphertexts
- chosen-ciphertext attacks (CCA) assume the attacker can choose ciphertexts to be decrypted, and receive the corresponding plaintexts

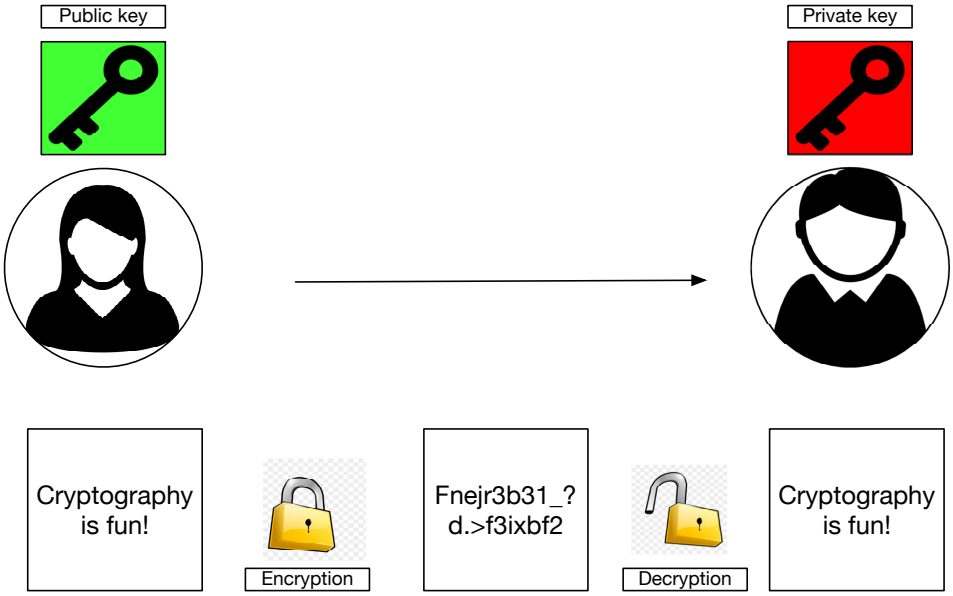


Fig. 2.4 The workings of a PKE scheme

- indistinguishability (IND) assumes the attacker is unable to distinguish between pairs of ciphertexts depending on the message they encrypt
- semantic security (SS) means only negligible information about the plaintext can be derived from the ciphertext
- non-malleability (NM) assumes the ciphertext can not be transformed into another valid ciphertext

Combining these leads to six security notions: IND-CPA, IND-CCA, SS-CPA, SS-CCA, NM-CPA, NM-CCA.

Numerous lattice-based PKEs have been proposed [Ajtai and Dwork, 1997, Goldreich et al., 1996, Hoffstein et al., 1998a, Regev, 2010]. The algorithms for a generic LWE-based PKE proposed by Regev [2005] and refined by Peikert et al. [2008] are given in Appendix C as Algorithms 36 to 38. The main security parameter N is the dimension of the lattice. Additional parameters such as the modulus q and standard

Table 2.5 Adversarial security concepts

Adversarial models
chosen-plaintext attacks (CPA) [Goldwasser and Micali, 1984] chosen-ciphertext attacks (CCA) [Rackoff and Simon, 1991]
Adversarial Goals
indistinguishability (IND) [Goldwasser and Micali, 1984] semantic security (SS) [Goldwasser and Micali, 1984] non-malleability (NM) [Dolev et al., 2003]

deviation σ of the Gaussian sampler also contribute to the security of the scheme. It is the complex interaction of the multitude of parameters which leads to the widely acknowledged difficulty in selecting the best in terms of security and efficiency for lattice-based schemes. The key generation of the LWE-based PKE scheme involves generating a vector $\mathbf{s}$ as the private key, and creating an LWE instance $(\mathbf{A}, \mathbf{b})$ as the public key, where $\mathbf{A}$ is a uniform matrix, $\mathbf{b} = \mathbf{A}s + \mathbf{e}$ and $\mathbf{e}$ is a Gaussian-sampled noise vector. To encrypt a message $\mathbf{v}$, a vector $\mathbf{a}$ is chosen uniformly at random and the ciphertext is output as $\mathbf{u} = \mathbf{A^T a}, \mathbf{c} = \mathbf{ba} + f(\mathbf{v})$, where f is the function which maps the vector $\mathbf{v}$ to the message space. To decrypt, the private key $\mathbf{s}$ is used to compute $f^{-1}(\mathbf{c} - \mathbf{s^T u})$.

Digital Signature Schemes (DSS)

In a DSS, the private key k_{priv} is used to sign the message by $Sign_{k_{priv}}$, and the public key k_{pub} is used to verify the signature with $Verify_{k_{pub}}$. The correctness of a DSS scheme is defined as

$$Verify_{k_{pub}}\left(Sign_{k_{priv}}(m)\right) = Accept$$

if and only if k_{priv} was correct. This ensures integrity, authenticity and non-repudiation. It can formally be defined as a system $\{KeyGen, Sign, Verify\}$ satisfying the properties:

- $KeyGen(*)$ outputs a pair of keys (k_{pub}, k_{priv}); the public and private keys respectively
- $Sign_{k_{priv}}(m)$ signs a message m under private key k_{priv}.
- $Verify_{k_{pub}}(m, sig)$ verifies the signature sig of a message m using k_{pub}.

The signature scheme's correctness is satisfied if, for all messages and key pairs, the signature produced by the signing algorithm is accepted by the verifying algorithm. Figure 2.5 shows how the components of a digital signature scheme interact. Real world use cases of digital signature schemes include authenticating software updates and ensuring data has not been tampered with during transit. Non-quantum-safe schemes currently in use include DSA [FIPS 186-4], EIGamal [ElGamal, 1985] and ECDSA [Vanstone, 1992]. Lattice-based DSSs are detailed in the next section.

An adversary may try to attack through key recovery or signature forgery methods and this forms the basis for security evaluation. There are multiple security notions relating to DSSs defined by Goldwasser et al. [1988] and given in Table 2.6. The main notions used in this thesis are EUF-CMA and sEUF-CMA .

- key only attacks assume the attacker only has access to the public key
- known message attacks assume the attacker has access to arbitrary signed messages
- chosen message attacks (CMA) assume the attack has access to signed messages of their choosing

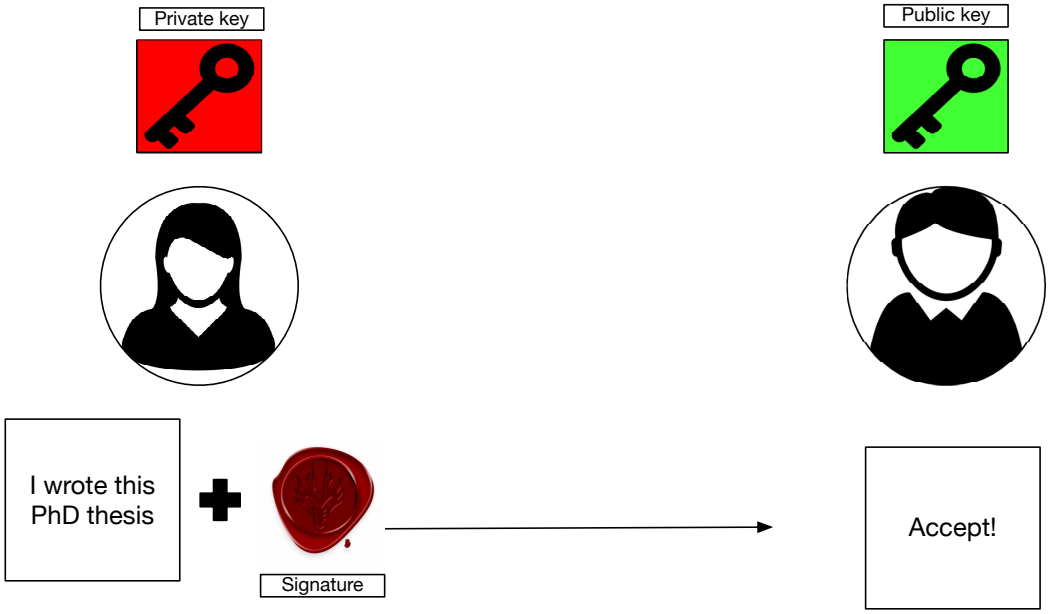


Fig. 2.5 The workings of a DSS scheme

- existential unforgeability (EUF) assumes the attacker cannot construct a valid (message, signature) pair for a message not previously signed
- strong existential unforgeability (sEUF) assumes the attacker cannot construct a valid (message, signature) pair such that the (message, signature) pair was not previously given
- selective forgery (SF) assumes the attacker can construct a valid signature for a message of their choosing
- universal forgery (UF) assumes the attacker can construct a valid signature for any given message

Table 2.6 Adversarial DSS security concepts [Goldwasser et al., 1988]

Adversarial models
key only attack known message attacks chosen message attacks (CMA)
Adversarial Goals
existential unforgeability (EUF) strong existential unforgeability (sEUF) selective forgery (SF) universal forgery (UF)

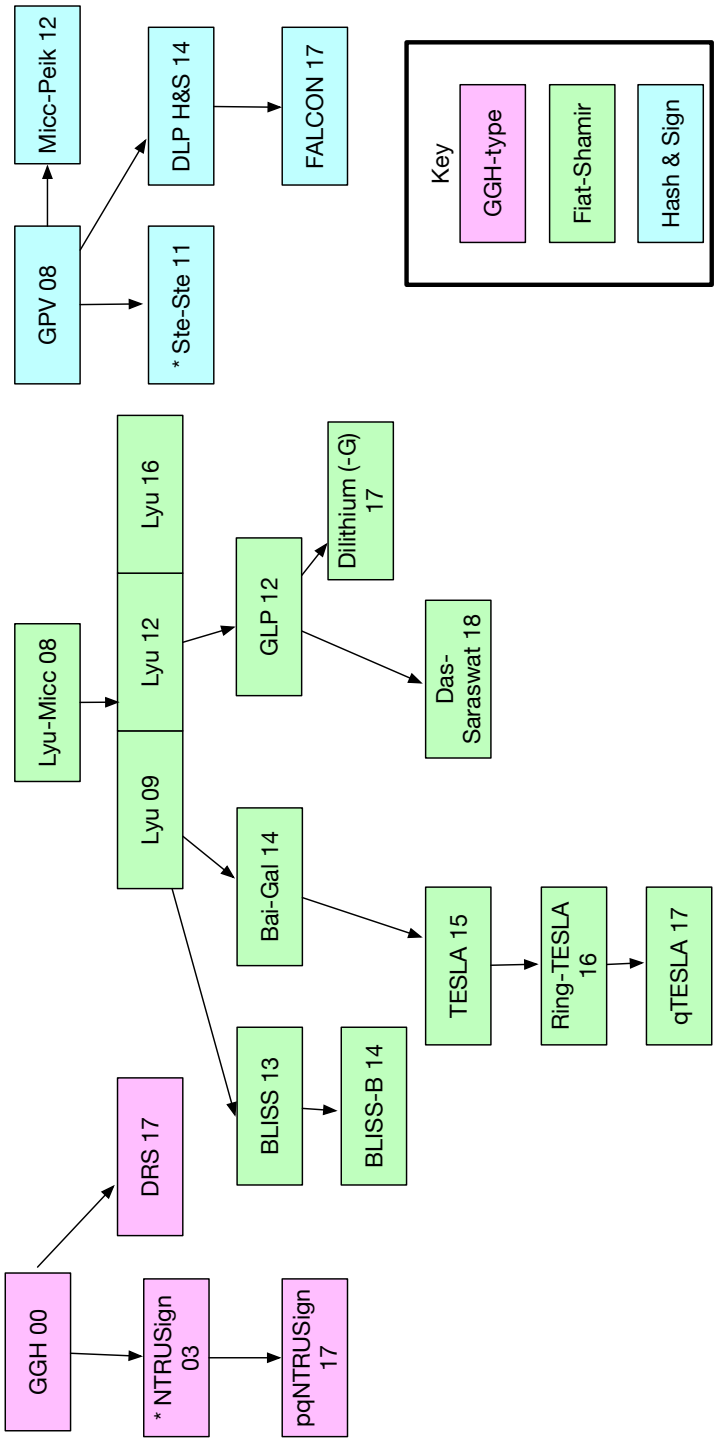


Fig. 2.6 The evolution of lattice-based digital signature schemes (* indicates further relations between schemes, explained in text)

The decision R-LWE based signature scheme proposed by Güneysu et al. [2012a], based on the construction by Lyubashevsky [2012], is given in Appendix C as Algorithms 39 to 41. This is known as the GLP scheme. The key generation component generates a random polynomial $(\mathbf{s_1}, \mathbf{s_2})$ as the secret key and $(\mathbf{a}, \mathbf{t})$ as the public key, where $\mathbf{a}$ is a uniformly sampled ring polynomial and $\mathbf{t} = \mathbf{a}\mathbf{s_1} + \mathbf{s_2}$. To sign a message $\mathbf{m}$, two masking polynomials $(\mathbf{y_1}, \mathbf{y_2})$ are uniformly sampled at random from the ring and the hash H of $(\mathbf{a}\mathbf{y_1}, \mathbf{y_2})$ is computed. The signature is then computed as $(\mathbf{z_1}, \mathbf{z_2}) = (\mathbf{s_1}\mathbf{c} + \mathbf{y_1}, \mathbf{s_2}\mathbf{c} + \mathbf{y_2})$, where $\mathbf{c}$ is a small hash output. The verification procedure constitutes distinguishing between a R-LWE instance and the uniform distribution over $\mathcal{R} \times \mathcal{R}$.

The evolution of lattice-based digital signature schemes

Lattice-based digital signature schemes can be separated into three categories, GGH-type, Fiat-Shamir, and Hash & Sign, and so they will be introduced in this manner. The timeline and connections between the schemes can be seen in Figure 2.6.

GGH-type The first lattice-based signature scheme, GGH, was proposed in 1996 by Goldreich et al., and its security was based on the CVP. The efficiency of GGH was later improved by using NTRU lattices [Hoffstein et al., 2003]. However, Nguyen and Regev exploited the fact that each signature leaks part of the secret key and were able to recover the NTRUSign-251 secret key using 400 signatures [Nguyen and Regev, 2009]. This meant that this first GGH-style of signature has no real practicality and no further published research has been carried out on it.

Other GGH-type schemes in the NIST contest include pqNTRUSign [Zhang et al., 2017b] and DRS [Plantard et al., 2017], neither of which progressed to the second round. A statistical attack for DRS was given in [Yu and Ducas, 2018], which recovered

the secret key with 100,000 signatures, reducing its 128-bit security to 70-bit. These can also be classified as hash and sign type signatures.

Fiat-Shamir In 2008, Lyubashevsky and Micciancio were the first to develop a lattice-based signature scheme [Lyubashevsky and Micciancio, 2008] along with a security proof in the standard model showing it is as hard as worst-case lattice problems, namely the approximate SVP over cyclic lattices. This was based on collision-resistant hash functions and had key and ciphertext sizes linear in dimension. This was a Fiat-Shamir type scheme, which uses the transform proposed in [Fiat and Shamir, 1986] to convert an identification scheme into an efficient signature scheme. Lyubashevsky [Lyubashevsky, 2009, 2012] improved upon this by introducing the notion of Fiat-Shamir with aborts. This allows the creation of strongly unforgeable signatures based on the SIS and LWE problems and their ring variants. Signatures are shortened via rejection sampling from the Gaussian distribution. Lyubashevsky's schemes were optimised for embedded systems in Güneysu et al. [2012b]. Alongside this, the authors proposed an improvement which reduces the signature size further by a factor of 2, using a new compression technique, whilst retaining security and in 2018, Das and Saraswat suggested even more compact signature and key sizes in order to make the scheme faster [Das and Saraswat, 2018]. They also show that forging the signature is harder than solving NTRU encryption.

Dilithium [Lyubashevsky et al., 2017] incorporated the strengths of previous schemes [Lyubashevsky, 2009, 2012, Güneysu et al., 2012a] and was submitted to the NIST call for cryptographic primitives. Based on Module-LWE, the public key size is compressed by removing low-order bits and replacing them with a "hint". It is secure in the Q-ROM. A Dilithium-G variant was presented in Ducas et al. [2017] which replaces the uniform sampler with a more efficient Gaussian sampler. However,

only the uniform-sampler version was submitted to NIST due to the cautiousness surrounding Gaussian samplers and their vulnerability to side channel attacks (as discussed in Chapter 3).

In Lyubashevsky [2016], a DSS based on the Ring-SIS problem is proposed which is shown to be secure for *any* $\mathbf{f}(x)$ such that the underlying ring is $\mathbb{Z}[x]\big/\mathbf{f}(x)$, as opposed to previous schemes which specified the polynomial $\mathbf{f}(x)$. The resulting scheme was as efficient as existing schemes over $\mathbb{Z}[x]\big/\langle x^N + 1\rangle$.

Bai and Galbraith built upon Lyubashevsky's compression techniques in 2014 to construct an efficient standard-LWE-based scheme [Bai and Galbraith, 2014]. This requires the signer to only prove knowledge of the secret rather than the error, and as a result the second half of the signature z_2 does not need to be sent. The security of the Bai-Galbraith scheme [Bai and Galbraith, 2014] is based on both LWE and SIS with a *loose* security reduction. In contrast, the authors of TESLA [Alkim et al., 2015] include a *tight* security reduction in the ROM from LWE over standard lattices alongside instantiations both against classical and quantum adversaries. Therefore the latter scheme is more desirable, as discussed in §2.2.2. TESLA has the further advantage of being deterministic and so does not rely on the security of a RNG as in [Bai and Galbraith, 2014]. Variants of TESLA include schemes by Akleylek et al. [2016], Barreto et al. [2016], Bindel et al. [2017], the latter of which (qTESLA) is a NIST candidate. qTESLA depends on the decision-R-LWE problem discussed in §2.3 and is a conservative scheme, provably secure in the Q-ROM. However, in 2019, concerns were raised over its security claims [Lyubashevsky, 2019].

BLISS, and its more efficient variant BLISS-B [Ducas et al., 2013a, Ducas, 2014b] are ring-based optimisations of the scheme in [Lyubashevsky, 2012]. Prior to the NIST

process, BLISS-B was considered the most popular and most efficient scheme, whose security is based on Ring-SIS over NTRU lattices.

Hash & Sign The third category of signatures is based on the framework of the GGH-type but introduces provable security. The GPV scheme [Gentry et al., 2008] can be seen as a theoretically secure version of the GGH scheme. It is based on pre-image samplable trapdoor functions and is the first to construct such a concept using lattices. It uses lattices known to have worst-case connections and replaces Babai's nearest plane procedure (discussed in Chapter 4) with random Gaussian sampling in order to hide the secret key. Based on standard lattices, it has complex key sizes and performance times are especially slow due to the expensive sampling procedure. Micciancio and Peikert proposed an efficient trapdoor sampler to improve its performance and to output shorter lattice vectors in [Micciancio and Peikert, 2012], but it was not until the introduction of NTRU-type lattices in [Ducas et al., 2014c] that practicality was achieved. This optimisation, along with a more compact public key and faster sampler, was included in the NIST submission Falcon [Prest et al., 2017] to achieve competitive performance. In [Stehlé and Steinfeld, 2011], the NTRUSign key generation and the GPV signing process are adapted to become provably secure in the ROM.

Key Encapsulation Mechanisms (KEM)

Rather than taking a user message as input as in a PKE, a KEM generates and transports a shared secret. Within a KEM, the public key k_{pub} is used to generate the shared secret and an encapsulation c using $Encap_{k_{pub}}$. The encapsulation is sent to the receiver, who decapsulates it using their private key $k_{priv}$with $Decap_{k_{priv}}$ and obtains the shared secret. In practice, this shared secret could be a key which can be used for more

efficient symmetric encryption schemes. A KEM can formally be defined as a system with three algorithms $(KeyGen, Encap, Decap)$ satisfying the properties:

- $KeyGen(*)$ outputs a pair of keys (k_{pub}, k_{priv}); the public and private keys respectively
- $Encap_{k_{pub}}(w)$ is used to encapsulate a secret key w to be shared as w'
- $Decap_{k_{priv}}(w')$ is used to decapsulate the encrypted secret key to obtain w

Figure 2.7 shows this process. Examples of lattice-based KEMs include [Naehrig et al., 2017, Schwabe et al., 2017, Alkim et al., 2016]. KEM security notions can be considered as similar to those of PKE, namely IND-CPA, IND-CCA, SS-CPA, SS-CCA, NM-CPA and NM-CCA. The lattice-based KEM Kyber [Schwabe et al., 2017] is presented in Appendix C by Algorithms 42 to 47 and is described here. The hardness of Kyber is based on the LWE PKE construction by Regev [2005], however its security depends on the hardness of solving the LWE problem over *module* lattices. The key generation component produces a public and secret key pair in a similar fashion to the LWE PKE described earlier in this section. A variant of the Fujisaki–Okamoto transform [Fujisaki and Okamoto, 1999] is applied so that the secret key includes the public key and a random value. The encapsulation component uses a Random Number Generator (RNG), two hash functions and a Key Derivation Function (KDF) to derive the key to be transported and encrypt the secret. The decapsulation attempts to regenerate this shared key by regenerating the random value by decrypting, checking by re-encryption and rerunning the KDF.

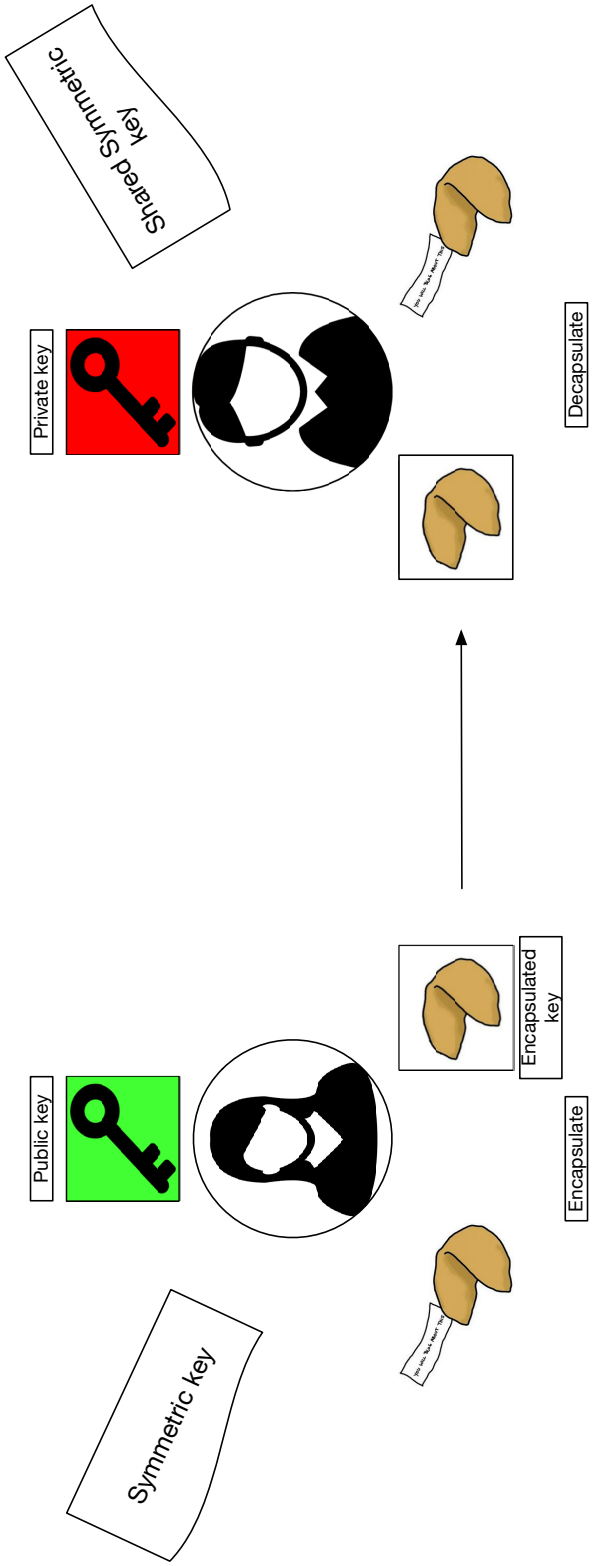


Fig. 2.7 The workings of a KEM

A note on Key Exchange vs. Key Encapsulation There is a subtle difference between a key exchange and a KEM. In a *key exchange*, an already established (usually symmetric) cryptographic key k is encrypted by a public key, sent and decrypted by the receiver's private key, so only they can access it. In an Authenticated Key Exchange (AKE) the user is additionally authenticated to ensure they are who they are expected to be. In contrast, and for improved efficiency, a KEM produces the cryptographic key k and encrypts, or encapsulates, and then sends it. The receiver can then decapsulate k using their private key.

2.4.5 Advanced Lattice-based Primitives

Lattices have the advantage of being able to provide more advanced primitives such as Fully Homomorphic Encryption (FHE) [Gentry, 2009], e-voting [Chillotti et al., 2016], IBE [Boneh and Boyen, 2004, Ducas et al., 2014a], ABE [Goyal et al., 2006, Boyen, 2013] and MPC [Damgård et al., 2012, López-Alt et al., 2012] and privacy-preserving encryption [Abdallah and Shen, 2016]. Conventional IBE schemes such as Boneh and Franklin [2001b] and Cocks [2001b] are based on mathematical problems over elliptic curves and quadratic residues which can be broken by Shor's algorithm [Shor, 1997]. LBC can be used to replace these. Recent research discussed in Chapter 4 has also shown they can be even more efficient than existing schemes [McCarthy et al., 2017].

An IBE scheme is one that uses the user's public identity as their public key. This mitigates the need for maintaining a database of public keys and certificates, and allows for further capabilities such as a client being able to send a message to a user without the user having prior registration. A typical IBE scheme consists of 4 components; master key generation, user key extraction, encryption and decryption.

The TETRA communications network [ETS, 2005], used for emergency services, currently implements the MIKEY SAKKE IBE scheme [Groves, 2012]. However, this is not quantum-secure as it is elliptic-curve based. Current practical quantum safe IBE schemes are based on structured lattices [Ducas et al., 2014a, Bert et al., 2018]. IBE schemes are covered in detail in Chapter 4.

A HIBE scheme is an extension of this concept which allows for finer grained control of key delegation abilities. It includes a sub-Key Management Service (KMS) along with a delegation function which allows the sub-KMS to extract user secret keys for the levels below. It can also delegate further sub-key pairs to the users below to enable them to become a sub-KMS in control of smaller groups of descendants. HIBE schemes are covered in detail in Chapter 5.

The notion of ABE was introduced by Sahai and Waters in 2005. It draws influence from IBE, but rather than encrypting based on a user identity, it encrypts based on a set of attributes of the user. There are two approaches to ABE. In key-policy attribute encryption, the trusted key generator encodes attribute policies within the secret keys it distributes, and ciphertexts can only be decrypted with a key that holds the required attributes and in ciphertext-policy attribute encryption, the encryptor specifies a set of attributes required by a receiver to enable them to decrypt the message. A user is only able to obtain the decryption key to decrypt the ciphertext if their attributes match the set. Examples of lattice-based ABE schemes are [Boneh et al., 2014, Dai et al., 2017].

Privacy-preserving encryption can be considered as a property of cryptographic schemes whereby a user's data is not revealed to any involved party. This can be realised through HE or MPC. Homomorphic encryption is a concept that allows computations on encrypted data without the need for prior decryption. Therefore

the confidentiality of the data is retained throughout the data processing lifecycle, which has clear advantages in applications such as cloud computing. Fully, somewhat, and partially homomorphic schemes differ with regards to the type and amount of computations which can be performed before the data becomes too noisy to recover after processing. These offer differing tradeoffs in efficiency and practicality. An open research area surrounding HE is the optimisation of performance to allow real-time computation on data. Lattice-based HE schemes to note are [Gentry, 2009, Gentry et al., 2013, Brakerski et al., 2014, Brakerski and Vaikuntanathan, 2014, Cheon et al., 2018].

MPC [Yao, 1982] is a concept whereby at least two parties are required to carry out a computation, without any of them learning anything about the computation other than the final output. Homomorphic encryption schemes have the potential to be developed into MPC schemes.

2.5 Mathematical building blocks of LBC primitives

2.5.1 Hash functions

A hash function is a bijective mapping H of an element from its domain to a random element in its image. The domain is the set of arbitrary length bit strings, denoted $\{0,1\}^*$, and its image is a set of fixed length l bit-strings. This can be formally written as $H : \{0,1\}^* \rightarrow \{0,1\}^l$. The hash function should have the properties of pre-image resistance, i.e. given y, it should be computationally infeasible to find x such that $H(x) = y$, second pre-image resistance, i.e. given x and $y = H(x)$, it should be

computationally infeasible to find x' such that $H(x') = y$, and collision resistance, i.e. it should be computationally infeasible to find $x \neq x'$ such that $H(x) = H(x')$.

In the context of LBC, a large message may be hashed into a smaller message before being signed, as in a hash-and-sign signature scheme (§2.4.4), to improve signature size. Hash functions are used throughout the thesis in several scenarios. The Falcon signature scheme introduced in Chapter 3 is of the aforementioned hash-and-sign type and so uses hash functions in this way. In the IBE (Chapter 4) and HIBE (Chapter 5) schemes, hash functions are used to extract the user key. The hash of the user identity is taken as its public key; this is a deterministic procedure and so still meets the requirements for a scheme to be identity-based. Furthermore, in the encryption procedure, an ephemeral key is sampled and hashed with the message to produce the ciphertext. This method follows the Fujusaki-Okamoto transform to protect against attacks utilising malformed ciphertexts. Throughout this thesis, the hash function maps into the relevant polynomial ring unless otherwise stated.

The specific choice of hash function is often left to the implementor. The software designs in this thesis use the following hash functions. The Falcon signature scheme uses FIPs-certified eXtendable-Output hash Function (XOF) SHAKE-256 and maps the output to a polynomial. The *libSAFEcrypto* design of SHAKE-256 given in this thesis uses SHA3-512 as the underlying hash function. For the (H)IBE schemes, the user identity is considered to be a MAC address, which can be expressed as a 48-bit char array. The methods in this design for mapping the user ID into a ring polynomial and where a random bit string is used to generate a one-time pad both use the hash function SHA-3.

2.5.2 Gaussian Sampling

A common feature of lattice-based constructions is the use of sampled noise to hide secret information. This is often Gaussian noise which allows for provable security of the schemes, unlike uniform noise [Regev, 2005, Applebaum et al., 2009]. Integers x can be sampled from a discrete Gaussian distribution, defined by:

$$\mathcal{D}_{\mu,\sigma} = \rho_\sigma(x) \big/ \rho_\sigma(\mathbb{Z})$$

$$\text{where } \rho_{c,\sigma}(x) = exp\Big(\frac{-||x-c||^2}{2\sigma^2}\Big)$$

A Gaussian distribution can be specified by its centre c, standard deviation σ and tail cut parameter τ, as shown in Figure 2.8. The Rényi divergence [Rényi et al., 1961] can be used to measure the distance between a sampled distribution and a true Gaussian. The definition given by Bai et al. [2018] is used in this thesis.

For any two discrete probability distributions P and Q such that $Supp(P) \subseteq Supp(Q)$ and $a \in (1, +\inf)$, the Rényi divergence of order a is

$$R_a(P||Q) = \left(\sum_{x \in Supp(P)} \frac{P(x)^a}{Q(x)^{a-1}} \right)^{\frac{1}{a-1}}$$

where $Supp(D) = \{x : D(x) \neq 0\}$, that is, the elements of the domain which are not mapped to 0.

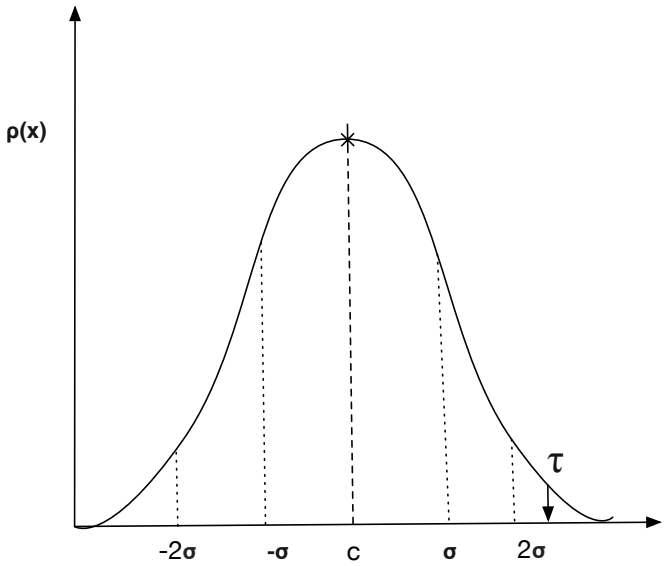


Fig. 2.8 A Gaussian distribution curve

Examples of commonly used Gaussian samplers in LBC include Bernoulli [Ducas et al., 2013a, Pöppelmann et al., 2014], Knuth-Yao [Knuth and Yao, 1976b], CDT [Peikert, 2010a] and Ziggurat [Buchmann et al., 2013a, Marsaglia et al., 2000].

2.5.3 Polynomial Arithmetic and the Number Theoretic Transform (NTT)

Standard lattice constructions involve computationally intensive matrix-vector multiplications. Ideal lattices reduce these to polynomial multiplications, which can be sped up by using the NTT in special cases. This mathematical process will now be described.

The NTT is a specialisation of the discrete Fast Fourier Transform (FFT) of integers modulo a prime p. Within the NTT domain, polynomials can be represented in point-value form, which consists of n different input/output pairs of the associated

polynomial equation. To translate a polynomial from the coefficient form to point-value form the NTT transform can be applied.

Within the NTT domain, multiplication of polynomials can be done point-wise, due to the properties of the complex number plane. This multiplicative method is termed negative wrapped convolution and the product $\mathbf{c}$ of $\mathbf{a}$ and $\mathbf{b}$ of length N under this method can be defined as:

$$c_i = \sum_{j=0}^{i} a_j b_{i-j} - \sum_{j=i+1}^{N-1} a_j b_{N+i-j}$$

for $0 \leq i \leq N-1$.

The NTT transform is only applicable when there exists a primitive root of unity ω, and for this it is required that $n = 2^k$, for some positive integer k, and that $q = 1 \mod 2n$ holds for modulus q.

For example, to multiply the polynomials

$$A = \sum_{0 \leq i \leq n-1} a_i \cdot x^i \text{ and } B = \sum_{0 \leq i \leq n-1} b_i \cdot x^i \tag{2.1}$$

the intuitive approach would be to use schoolbook multiplication, taking $O(n^2)$. Instead, the polynomials can be translated into the NTT domain, and then multiplication (convolution) can be performed point-wise, $c_i = a_i \cdot b_i$, in $O(n)$. Applying the inverse NTT function will output the polynomial $C = A * B$ in coefficient form. The NTT (and Inverse Number Theoretic Transform (iNTT)) algorithms can be implemented as per Cooley and Tukey [1965] with complexity $O(nlogn)$.

2.5.4 Gram-Schmidt Orthogonalisation (GSO)

A lattice basis is unique, however linear transformations can be applied to it, yet still define the same vector space ($\mathcal{B} \rightarrow \tilde{\mathcal{B}}$ and $Span(\mathcal{B}) = Span(\tilde{\mathcal{B}})$). A *good* basis is one which is short and orthogonal and will usually be the private key. A *bad* basis will ideally look uniform and can then be used as the public key. A good basis allows a holder to derive short vectors from the lattice, which is usually the secret information produced during an encryption or signing process in order to hide the message or create a signature. The Gram-Schmidt process is essentially a mechanism of linear transformations which shorten and orthogonalise the basis matrix. This process is used in both trapdoor sampling methods (Section 2.5.5) and cryptanalysis (Section 2.6) as the aim of both is to generate short lattice vectors. The GSO process works as follows; let $\mathbf{B} = \{b_1, b_2, ..., b_n\}$ be a basis, and the associated GSO-basis will be denoted as $\mathbf{B}^* = \{b_1^*, b_2^*, ..., b_n^*\}$. The GSO-basis is computed as

$$b_i^* = b_i - \sum_{1 \leq j \leq i-1} \frac{\langle b_i, b_j^* \rangle}{||b_j^*||^2} \cdot b_j^*$$

The Gram-Schmidt Norm (GS-norm) $||\mathbf{B}||_{GS}$ is a property of the basis $\mathbf{B}$. It is the maximum of the norms (moduli) of the vectors in the Gram-Schmidt orthogonalisation of the basis.

$$||\mathbf{B}||_{GS} = ||\tilde{\mathbf{B}}|| = \max_{i \in I} ||\tilde{b_i}||$$

2.5.5 Trapdoor Sampling

The hardness of many lattice-based schemes depends on finding short vectors in a lattice. These schemes use a trapdoor function, which is a function that is easy

to compute, but hard to invert unless one has possession of a trapdoor piece of information. This trapdoor is usually a short lattice basis which allows us to find short lattice vectors via a trapdoor sampler. The Gram-Schmidt Orthogonalisation (GSO) process would be used to generate a short lattice basis and then used to randomly select short lattice vectors. A range of different trapdoor samplers are available in the literature, differing in terms of their efficiency and the quality, or shortness, of the output lattice vectors. Popular samplers such as the GPV sampler [Gentry et al., 2008], are based on Klein's construction [Klein, 2000], which can be viewed as a randomised variant of Babai's Nearest Plane algorithm [Babai, 1986a] (explained in Chapter 4). The sampler proposed by Peikert [Peikert, 2010a] is a randomised variant of Babai's algorithm, and Micciancio-Peikert also proposed an efficient sampler in [Micciancio and Peikert, 2012]. Some samplers also make use of underlying lattice structures to improve performance [Ducas and Prest, 2016b]. Table 2.7 from [Prest et al., 2017] outlines key features of the available samplers.

The schemes in this thesis all make use of the GPV Sampler [Gentry et al., 2008]. Given a secret lattice basis $\mathbf{B}$ and standard deviation σ, the algorithm chooses a lattice vector at random under a Gaussian probability distribution. The value of standard deviation determines the usefulness of the sampled lattice vector; too small and the sampler will be subject to learning attacks, too large and the sampled vector will be too large to be of use in cryptography, and so σ is chosen in relation to $||\mathbf{B}||$. The output distribution of the sampler depends only on the length of the input basis and therefore the secret basis information is protected. The algorithm closely follows that of Babai's Nearest Plane, traditionally used in cryptanalysis for finding short vectors in the lattice, but instead introduces randomness from a Gaussian distribution over the integers. Whereas the original algorithm rounds a vector to the nearest lattice plane, the sampler

Table 2.7 Comparison of trapdoor samplers [Prest et al., 2017]

Sampler	**Fast**	**Short output**	**NTRU-compatible**
Klein/GPV [Klein, 2000, Gentry et al., 2008]	No	Yes	Yes
Peikert [Peikert, 2010a]	Yes	No	Yes
Micc-Peikert [Micciancio and Peikert, 2012]	Yes	Yes	No
Ducas-Prest [Ducas and Prest, 2016b]	Yes	Yes	Yes

randomly chooses a plane according to a Gaussian distribution, in order to reduce any dependence on the secret input basis. The GPV sampler is further dissected in §4.3.2, where algorithmic optimisations are introduced. The signature scheme Falcon uses a novel recursive form of the GPV sampler, called the $FFTSampler$, which makes use of the tower of rings structure of the cyclotomic ring and additionally operates within the FFT domain, so it is faster and more compact. The application of this version of the sampler to a HIBE scheme is proposed in Chapter 5.

2.6 Basis Reduction Methods

Most algorithms used for solving the LWE and Ring-LWE problems depend on basis reduction of the associated lattice and so this is the first step towards analysing the security properties of a design. Reducing a basis is essentially transforming the basis to become more orthogonal and shorter.

2.6.1 LLL

The Lenstra–Lenstra–Lovász lattice reduction algorithm (LLL) algorithm consists of two main steps. The first is **size-reduction** of the basis vectors. The size-reduction algorithm is based on the GSO (Section 2.5.4). However, the GSO cannot be used

directly for lattice basis reduction as the output vectors may not necessarily be in the lattice. The size-reduction method essentially lifts the vector back to the nearest lattice element after each projection. The subsequent vectors will then be projected onto the span of the previous vectors in their “lifted back" position and so the GSO coefficients have to be recalculated to account for the difference this makes. As the resulting vectors are not exactly orthogonal, the final GSO coefficients

$$\mu_{ij} = \frac{\langle \mathbf{b}_i, \mathbf{b}_j^* \rangle}{||\mathbf{b}_j^*||^2}$$

will not equal 0 or 1 as in the standard GSO method, but as long as the modulus $|\mu_{ij}| \leq \frac{1}{2}$, this basis is considered size-reduced.

The second step of the LLL-algorithm is the **lattice condition** step. This aims to reduce the norms of the ordered basis vectors, beginning with the first. This condition is derived from the fact that the GSO depends on the order of the basis vectors. If $\mathbf{b}_i$ and $\mathbf{b}_{i+1}$ are swapped for some $1 \leq i < m$, only $\mathbf{b}_i^*$ and $\mathbf{b}_{i+1}^*$ would change. For example, let

$$\mathbf{b}_i = \mathbf{x} \tag{2.2}$$

$$\mathbf{b}_{i+1} = \mathbf{y} \tag{2.3}$$

and the GSO output vectors:

$$\mathbf{b}_i^* = \mathbf{x}^* \tag{2.4}$$

$$\mathbf{b}_{i+1}^* = \mathbf{y}^* \tag{2.5}$$

Suppose $\mathbf{x}$ and y are swapped:

$$\mathbf{b}_i = \mathbf{y} \tag{2.6}$$

$$\mathbf{b}_{i+1} = \mathbf{x} \tag{2.7}$$

Denote:

$$new\ \mathbf{b}_i^* = \mathbf{y}^{*\prime} \tag{2.8}$$

$$new\ \mathbf{b}_{i+1}^* = \mathbf{x}^{*\prime} \tag{2.9}$$

Then

$$\mathbf{y}^{*\prime} = \mathbf{y}^* + \mu_{i+1,i} \cdot \mathbf{x} \tag{2.10}$$

where $\mu_{i+1,i}$ is the original GSO coefficient. When comparing the Right Hand Side (RHS) of equation 2.10 with the RHS of the Lovász' condition, which is defined as:

$$\delta \|\mathbf{b}_i^*\|^2 \leq \|\mathbf{b}_{i+1}^* + \mu_{i+1,1}\mathbf{b}_i^*\|^2 \tag{2.11}$$

Therefore, based on Lovász' condition, it can be stated that

$$\delta \cdot old \|\mathbf{b}_i^*\|^2 \leq new \|\mathbf{b}_i^*\|^2 \tag{2.12}$$

should hold. If this does not hold, i.e. if the existing norm of $\mathbf{b}_i^*$ is large, swap vectors. The new $\mathbf{b}_i^*$ will equal the RHS of Equation 2.12, i.e. as Lovász' does not hold, it will be smaller than $\delta\ old \|\mathbf{b}_i^*\|^2$ and so the norm of $\mathbf{b}_i^*$ will have been reduced. The best reduction will be achieved when $\delta = 1$. However, it is an open problem as to whether this can be achieved in polynomial time. For $\frac{1}{4} \leq \delta \leq \frac{3}{4}$, the algorithm can be shown to terminate in polynomial time.

The LLL is formally given in Algorithm 1.

Algorithm 1: The LLL algorithm [Deng]

Data: $\mathbf{b}_1, \mathbf{b}_2...., \mathbf{b}_N$
Result: The LLL reduced basis

```
for i = 1 → N do
    for j = (i − 1) → 1 do
        b_i = b_i − ⌊μ_ij⌉ b_j
for i = 1 → N − 1 do
    if δ‖b*_i‖² ≥ ‖b*_{i+1} + μ_{i+1,1} b*_i‖² then
        swap b_i and b_{i+1} and return to start
return LLL-reduced b_1, b_2...., b_N
```

2.6.2 BKZ

Schnorr and Euchner introduced and implemented the Block Korkine-Zolotarev lattice reduction algorithm (BKZ) [Schnorr and Euchner, 1994] in the Number Theoretic Library (NTL) [Shroup, 1996]. It is a generalisation of the LLL algorithm. Rather than satisfying the Lovásv condition, BKZ considers blocks of β vectors in the basis and finds the shortest vector in the lattice projected by these vectors in order to check the BKZ condition defined below. If this is violated, the vectors are swapped.

Suppose Λ a lattice of rank d, $\beta \in \mathbb{Z}$ s.t. $2 \leq \beta \leq d$, $\delta \in (\frac{1}{4}, 1)$. Then the lattice basis is BKZ-reduced if:

$$\delta \|\mathbf{b}_i^*\|^2 > \|\mathbf{b}_1^{new}\|^2 \, \forall 1 \leq i \leq d-1,$$

where $\mathbf{b}_i^{new}$ =shortest vector in the projected lattice of $\{\mathbf{b}_1, ..., \mathbf{b}_{min(n+\beta-1,d)}\}$.

The Hermite factor γ_N is a measure of how short a vector of a lattice of dimension N can be and is defined as $\gamma_N = \left(\frac{4}{3}\right)^{\frac{N-1}{2}}$ and a lower Hermite factor corresponds to a shorter, more orthogonal basis. The BKZ algorithm aims to get a lattice as short as

Table 2.8 Hermite Factors reached by lattice reduction algorithms

Algorithm	**BKZ**	**LLL**
Hermite factor	$\gamma_N(1+\epsilon)^{\frac{N-1}{2(\beta-1)}}$	$\gamma_N(1+\epsilon)^{\frac{N-1}{2}}$

possible, and therefore aims to reach as close to this lower bound as possible. The BKZ algorithm can reach lower Hermite factors (implying a better quality basis) than LLL (Table 2.8), however its complexity is at least super-polynomial in dimension N [Hanrot et al.]. In [Hoffstein et al., 2017], a step function is proposed to translate a given Hermite factor into a concrete security level. In [van de Pol and Smart, 2013], with $N = 131072$, a Hermite factor of 1.0055 was reached, when the algorithm was limited to 2^{256} operations.

2.6.3 BKZ 2.0

Chen and Nguyen made the following improvements to BKZ [Chen and Nguyen, 2011] and presented a simulation algorithm to obtain running times (again super-polynomial).

- Enumeration
- Extreme pruning
- Terminating after a fixed number of rounds
- Preprocessing of blocks

They simulated for block sizes as high as 286 to reach a Hermite factor of 1.005, the best result to date.

2.7 Existing practical constructions and implementations

Research demonstrating that lattice-based constructions are practical for real-world application is ongoing, with promising results. A high level comparison of open-source LBC software libraries is given in Table 2.9. These include libSAFEcrypto [SAFEcrypto, 2018], OpenQuantumSafe (OQS) [Stebila and Mosca, 2016], LatticeCrypto [Microsoft, 2016] and PALISADE [pal, 2018]. OQS additionally integrates the algorithms into protocols such as OpenSSL. LatticeCrypto includes fast multiplication and constant time components. PALISADE's main focus is on homomorphic encryption, but it includes core cryptographic components applicable to other lattice-based schemes. The software designs presented in this thesis form part of libSAFEcrypto [SAFEcrypto, 2018], discussed in §2.7.1. Furthermore, note that the results reviewed in this section are from a range of platforms and security targets.

The generic R-LWE encryption scheme has been shown to outperform traditional ECC-based PKE in software by at least one order of magnitude [De Clercq et al., 2015]. To compare, the authors state that the Elliptic Curve Integrated Encryption Scheme (ECIES) would take over 5 million cycles for encryption, whilst their R-LWE scheme takes 120,000 cycles. It took 1.67ms at 168MHz for a full run of the R-LWE scheme {Keygen, Encrypt, Decrypt}, on a 32-bit ARM Cortex-M4F microcontroller as the target platform. This was achieved through the use of the Knuth-Yao sampling algorithm to implement a fast discrete Gaussian sampler, and NTT optimisations from [Roy et al., 2014b] to speed up polynomial multiplication.

Recent implementation results of the lattice-based KEM NewHope for 229-bit classical security give a total primitive runtime of 0.3s on an Intel Core i7-4770K (Haswell) running at 3492MHz [Alkim et al., 2016]. In [An et al., 2018], the OpenQuantum-Safe Library implements NewHope and Kyber with runtimes of 0.23ms and 0.38ms respectively on an Intel(R) CPU i7-5500, with 16GB RAM. This is in stark contrast to the isogeny-based SIDH [Jao et al., 2019], which had a total runtime of 471ms on the same platform with comparable security level (NIST Level 5). For this security level, NewHope key sizes are: public key 1824 bytes, secret key 1792 bytes and ciphertext 2176 bytes (CPA variant). Furthermore, NewHope was successfully implemented in Google's Boring SSL within their Chrome browser as part of a bench-marking experiment [Braithwaite, 2016].

A full run of the reference implementation of the Dilithium signature scheme [Ducas et al., 2017, Lyubashevsky et al., 2017] {Keygen, Sign, Verify} on an Intel Core i7-6600U Skylake CPU took 0.96ms at 2.6GHz for the recommended 125-bit classical security parameter set. Parameters were chosen so as to allow the NTT to be utilised to speed up polynomial multiplication. The expansion of a seed into a matrix constitutes the majority of the signing and verifying computation time, and hence highly optimised SHAKE-128 and -256 hash function designs can greatly contribute to speedups of the scheme, as noted in a NIST pqcforum comment[4]. The size of the public key can be reduced by a factor of 2.5 at the expense of increasing the signature by around 150 bytes [Ducas et al., 2017, Lyubashevsky et al., 2017]. For the recommended intermediary security level, the scheme has 2.7KB signatures and 1.5KB public keys.

[4]https://groups.google.com/a/list.nist.gov/d/msg/pqc-forum/Tq3OqNMRr-s/chYGXIwoAgAJ

Table 2.9 Lattice-based software libraries

Library	**Lang.**	**Schemes supported**	**Main contributions**
libSAFEcrypto[SAFEcrypto, 2018]	C	BLISS-B, Dilithium / Dilithium-G, Ring-TESLA, Falcon, ENS-(sig/KEM), Kyber-(Enc/KEM), RLWE-Enc, DLP-(Sig/IBE)	LBC competitive with ECC Practical advanced primitives
OpenQuantumSafe [Stebila and Mosca, 2016]	C	FrodoKEM, Kyber, BCNS, Lima, NewHope, qTESLA, Three Bears, Dilithium, Saber, Titanium	Focus on NIST candidates
LatticeCrypto [Microsoft, 2016]	C	NewHope	Fast NTT
Palisade [pal, 2018]	C++	Range of HE Schemes e.g.FV, BGV,LTV, StSt	HE-based Core LBC components

Table 2.10 presents implementation options for LBC components, specially the NTRUSolve algorithm which generates the NTRU polynomials F, G such that the NTRU equation is satisfied (see §2.4.2), the Gaussian sampler and the hash functions used. Table 2.11 gives a taxonomy of advanced lattice-based schemes in this thesis. Implementation Choice (IC) denotes where the choice is not fixed in the scheme's specification and so the implementor has freedom of choice. The choice of building blocks available for each scheme allows for an interesting investigation into choosing optimally for each scheme. A diagramatic representation of this is given in Appendix B.

Table 2.10 Implementation options for LBC components(IC-implementor choice)

	Options
NTRUSolve	Classical TowerSolverI TowerSolverR
Gaussian	Bernoulli CDT Knuth-Yao Ziggurat
Hash	SHA2 SHA3 BLAKE2 Whirlpool

Table 2.11 A taxonomy of the schemes in this thesis

	KeyGen		Extract/Sign		Delegate		Encrypt/Decrypt
	Sampler	NTRUSolve	Hash	Short lattice vector sampler	NTRUSolve	Short lattice vector sampler	
DLP	IC	Classical	IC	GPV Sampler	N/A	N/A	R-LWE
Falcon-Sig	IC	TowerSolverI	IC	FFT Sampler	N/A	N/A	N/A
Falcon-IBE	IC	TowerSolverI	IC	FFT Sampler	N/A	N/A	R-LWE
Latte	IC	Classical	IC	GPV Sampler	Classical	GPV Sampler	Extended R-LWE
Skinny Latte	IC	TowerSolverI	IC	FFT Sampler	TowerSolverI	FFT Sampler	R-LWE

2.7.1 H2020 SAFEcrypto project

The Horizon 2020 (H2020) SAFEcrypto project (Secure Architectures of Future Emerging Cryptography)[5] ran from 2014 to 2018 with the aim of providing practical post-quantum cryptographic solutions. The consortium was led by Queen's University Belfast (QUB) and consisted of both academic and industrial partners. SAFEcrypto focussed on applications which are resource constrained or require real-time data transfer, with the specific case studies of IoT devices, satellite communications and municipal data analytics. The research presented in this thesis (specifically, the DSS designs in Chapter 3 and IBE scheme design in Chapter 4) contributes to Work Package 6 of the project, which focussed on the development of a suite of LBC software routines, and to a book to be published by Springer [O'Neill et al., 2020].

One of the outcomes of the project was an open-source software library [SAFEcrypto, 2018] that includes lattice-based PKE, DSS, KEM and IBE schemes. The IBE software design was an outcome of the research presented in Chapter 4 and indicates the practicality of the scheme for constrained devices. The IBE scheme was additionally implemented on hardware (results in Table 4.9 in Chapter 4) and a further outcome of the project was an IBE demonstrator which utilised the DLP-IBE scheme on sensor tags, using the Tag Serial ID numbers, to show its potential for use on embedded devices.

The libSAFEcrypto library consists of portable C code that is written to permit a compiler (gcc) to generate AVX-2 assembler code (or any other type of SIMD code) without having to revert to writing assembler code by hand. It can also be ported to other target processors (like ARM) without having to re-write the code; this portability

[5] www.safecrypto.eu

was set out as one of the major design goals in the SAFEcrypto project to provide for wide adoption. This portability comes at a performance cost, but is advantageous if the user wants to support various systems (embedded or server), quickly adapt to changing algorithms (like a NIST candidate), or is looking for wide support.

It is intended for general-purpose applications ranging from high-end 64-bit Intel Xeon servers to 32-bit ARM Cortex-M embedded systems. The designs presented here focus on 32-bit and 64-bit x86 processors in an Ubuntu/CentOS Linux environment, with options to configure client compilation on ARM v7 Cortex-A and Cortex-M target platforms. The Autotools build system is used to deliver the scheme as a library within a software distribution that can be suitably adapted to the host system at compile-time, i.e. utilising alternative algorithms for environments with constrained RAM. Additional adaptations can be configured at run-time, such as the selected underlying cryptographic functions (i.e. Cryptographically Secure Pseudo-Random Number Generator (CSPRNG) or hash functions), NTT optimisations and modular reduction techniques.

The code was validated using unit tests to ensure that the individual software units operate and behave as expected (for example, if pointer is NULL when it should not be, an integer is out of range, etc.). As the software components are built up from these units, additional functional tests were added to verify that more complex functionality works as expected and software systems do what they were designed to do (an RNG generates random numbers, a linked list can create lists that can be traversed, etc.). The cryptoschemes as a whole were then tested against test vectors supplied via the reference software).

2.8 Discussion

In this chapter, an introduction to LBC is given, alongside important related topics in order to provide context to this thesis. The global efforts towards standardisation are described and a comparison of open-source software libraries is given. Basic and advanced cryptographic primitives are defined, and lattice-based examples of each are covered. Cryptographic building blocks which apply to a range of lattice-based schemes, and which are often the performance bottlenecks and security weak-points, are described.

Although substantial efforts have been made into implementing lattice-based schemes in software, a desirable goal would be to have comparable implementations of all the schemes in order to have a fair basis upon which to judge them. Lattice-based schemes share common components so this would be a good starting point. Many lattice-based schemes entail a trapdoor sampler, which samples short vectors in the lattice. A discrete Gaussian sampler is used to provide randomness. There are concerns over the side-channel vulnerabilities of the discrete Gaussian sampler [Pessl et al., 2017, Bruinderink et al., 2016b], and the research community's current focus is towards efficient side-channel resistant trapdoor samplers that output good quality, short vectors [Klein, 2000, Micciancio and Peikert, 2012, Gentry et al., 2008]. Other aspects of LBC that can be improved in terms of efficiency and physical attack vulnerabilities are polynomial multiplication and Gram-Schmidt Orthogonalisation.

With regards to basic cryptographic primitives, the focus of the community is the NIST standardisation process [NIST, 2016]. NIST has expressed interest in continued research and analysis on all of the 2nd round candidates and has said that performance (on both hardware and software) and resistance to physical attacks will play a role

in the final selection process. Additional benchmarks and investigations into how submissions fit into applications/protocols are required [Moody, 2019]. Lattice-based schemes make up almost half of these candidates, spanning PKE, DSSs and KEMs.

Chapter 3 of this thesis focusses on two NIST candidates, Falcon [Prest et al., 2017] and Dilithium [Lyubashevsky et al., 2017] and aims to address the aforementioned gaps. An evaluation of their performance is given, and they are compared to an existing well-studied lattice-based signature scheme, BLISS-B [Ducas, 2014b]. The physical attack resistance of the schemes is also considered; the first fault attack on the Falcon trapdoor sampler is proposed and a review of existing attacks is given. Protective countermeasures are recommended and integrated into the software designs.

Advanced post-quantum cryptographic primitives are an emerging area of research, and the practicality of such schemes has not yet been proven. Prior to this research, there has only been a Proof of Concept (PoC) implementation of the DLP-IBE scheme [Ducas et al., 2014b] and no demonstration of the practicality of such advanced schemes. With the deployment of IoT, there are additional use-cases for these more advanced primitives. For example, quantum-safe IBE would be suitable for use in the TETRA communications system to replace current classical IBE approaches and in sensor tag applications. Chapter 4 presents the first practical optimisation of the DLP-IBE scheme and shows it is feasible for such applications.

The development of more complex applications, on smaller devices, means there is a requirement for even more advanced constructions. Chapter 5 introduces lattice-based HIBE schemes, which offset the workload of the master key manager and allow for key handling duties to be divided into respective localities. A novel scalable HIBE scheme is proposed. These are the first practical performance results and analyses of a quantum-safe HIBE scheme to be reported in the literature.

CHAPTER 3

The Impact of Physical Attack Countermeasures on the Performance of Lattice-based Digital Signature Schemes

3.1 Introduction

In Chapter 2, the topic of lattice-based cryptography was introduced. Lattices lend themselves to various primitives, one of which is digital signatures, important in authentication, non-repudiation and ensuring integrity of data. The first lattice-based digital signature scheme was introduced in Goldreich et al. [1996], and many different schemes and variants have been proposed, cryptanalysed and developed further [Hoffstein et al., 2003, Ducas et al., 2013a, Gentry et al., 2008, Lyubashevsky and Micciancio, 2008]. Lattice-based signatures have been shown to perform well in comparison to classical signature schemes [Howe et al., 2015]. However, more recent proposals submitted to the NIST process have not been deeply analysed and understanding around physical security is lacking. The aim of this chapter is to provide

an evaluation of the impact of physical attack countermeasures on the performance of two of the NIST candidates, Dilithium [Lyubashevsky et al., 2017, 2019, Ducas et al., 2018] and Falcon [Prest et al., 2017, 2019], together with their predecessor signature scheme, BLISS-B [Ducas et al., 2013b, Ducas, 2014c].

Lattice-based signatures have been shown in the past to be susceptible to side-channel analysis. These include timing attacks [Silverman and Whyte, 2007, Bruinderink et al., 2016b, Pessl, 2016a, Vizev, 2007], power analysis [Primas et al., 2017, Atici et al., 2008, Wang et al., 2013], cache attacks [Bruinderink et al., 2016b, Pessl et al., 2017, Espitau et al., 2017] and differential fault analysis [Espitau et al., 2016, 2018, Bindel et al., 2016]. Falcon is a lattice-based signature candidate submitted to the NIST PQC initiative, which has demonstrated good performance, however there is no research with respect to its resistance to implementation attacks.

Both BLISS-B and Dilithium signature schemes have been subject to recent cache and fault attacks (2017 and 2018, respectively), and the specification of Falcon acknowledges that there is "unclear side-channel resistance" for this scheme. This research evaluates existing attack countermeasures on these schemes and investigates the performance costs of these.

The research considers a cache attack on BLISS-B [Ducas, 2014a] and a fault attack on Dilithium [Lyubashevsky et al., 2017], examines proposed countermeasures for these attacks, and profiles the cost overhead of these countermeasures. A further contribution is the first fault attack proposed on the Falcon scheme. The attack is simulated and the runtimes and success rate of the post-processing component is presented, along with corresponding countermeasures and performance costs. An existing attack by Espitau et al. [2016] instigates a loop abort fault on the lattice vector GPV sampler (discussed in §2.5.5). However, Falcon uses a specialised FFT sampler

to sample short lattice vectors. Therefore, this research creates an analogous attack on the sampler to give equivalently faulty output signatures. Although the attacker is required to give more precise instructions than in Espitau et al., it is considered to be a genuine future threat as attacking methods become more sophisticated. This is the first research to evaluate the impact of implementing countermeasures on the performance of these schemes, in response to such attacks.

The contributions of this chapter are as follows:

- The first software designs of Dilithium, BLISS-B, and Falcon with physical attack countermeasures in place to demonstrate the effect on their performance.
- A new fault attack on the Falcon sampler using a Basis Extraction by Aborting Recursion or Zeroing (Bearz) technique.

The structure of this chapter is as follows. Section 3.2 describes the digital signature schemes included in this research as well as previous work on specific implementation attacks and proposed countermeasures for each scheme. Section 3.3 presents the first fault attack on the Falcon signature scheme and in Section 3.4 discerns countermeasures for this, as well as countermeasures reported for previous attacks on BLISS-B and Dilithium. Section 3.5 presents performance figures and evaluates the designs of the three schemes, with and without attack countermeasures. Section 3.6 concludes the chapter.

The main research of this chapter appears in the publication [McCarthy et al., 2019].

3.2 Physical attacks on LBC DSS

This research evaluates the performance of software implementations of three lattice-based digital signature schemes with and without physical attack countermeasures: Dilithium [Lyubashevsky et al., 2017, Ducas et al., 2018], BLISS-B [Ducas et al., 2013b, Ducas, 2014c], and Falcon [Prest et al., 2017]. Table 3.1 summarises the main differences between the three schemes; the rationale for choosing these schemes, short descriptions and specific implementation attacks of each scheme will now be given.

Despite being resistant to quantum computing attacks, lattice-based cryptographic schemes are susceptible to the same side-channel attacks as the primitives used today, as summarised by Hodgers et al. [2016] and Khalid et al. [2018]. Thus, considering attacks and possible countermeasures is important during standardisation and implementation. This is a young field, with the first side-channel attack on lattice-based signatures presented at CHES 2016 [Bruinderink et al., 2016b]. This research focuses on countermeasures of attacks that have been considered to date for lattice-based signatures, namely timing, cache, and fault analysis attacks. Examples of these types of attacks and countermeasures are given in Roy et al. [2014a], Bruinderink et al. [2016a], Howe et al. [2016], Espitau et al. [2016], Bindel et al. [2016] and Karmakar et al. [2018].

Timing attacks exploit data-dependent execution times in order to derive secret information, for example through secret memory access and secret-dependent branching. The Gaussian samplers used in BLISS-B and Falcon to sample short lattice vectors are known to be vulnerable to timing and power analysis attacks, however constant-time samplers can be expensive. Bruinderink et al. [2016a] proposed the first timing attack on BLISS and recovered the private key by targeting the scheme's Gaussian sampler, specifically the different response times of the CDT sampler. Roy et al. [2014a] in-

troduced shuffling of the Gaussian-distributed vector as a countermeasure, which was also extended by Howe et al. [2016]. However, Pessl [2016a] were able to attack shuffled samples through a cache attack on BLISS. Other examples of timing attacks include that by Silverman and Whyte [2007], which used differing numbers of calls to the hash function in NTRU decryption to recover the secret key. The risk of timing attacks highlights the need for constant-time implementations.

Prior to the NIST standardisation process, Fiat-Shamir with Aborts-type schemes were favoured due to their compactness and security. Amongst these schemes, BLISS-B boasted the smallest signature sizes. However, BLISS-B utilises a Gaussian sampler which has been shown to be insecure against side channel attacks. On the contrary, the Bai-Galbraith (B-G) 2014 scheme uses uniform sampling, a component which is also used in Dilithium. Dilithium was proposed by common authors to BLISS-B, therefore the most recent scheme was optimised and submitted to the NIST process as Dilithium. It incorporates the beneficial properties from Dilithium and BLISS-B in order to reduce the public key and signature sizes, whilst avoiding aspects of BLISS-B which are vulnerable to implementation attacks, as described in Section 3.2.1. Both schemes are included here as the focus of this chapter is on physical attack security and efficiency trade-off, and these two variants provide a similar foundation upon which to draw comparisons.

Falcon is considered as it is of the other “hash-and-sign" DSS type, which traditionally has been seen as less efficient, but which to date has not had the same attention with regards to its physical attack resistance. It is also a NIST candidate and so this research aims to progress the NIST evaluation process by drawing comparisons and differences between hash-and-sign and Fiat-Shamir NIST candidates, with consideration for physical attack resistance. Furthermore, Falcon and BLISS-B are both

based on NTRU-type lattices, while Dilithium is over module lattices which offer a lesser exploitable structure, at a loss in compactness. This adds another dimension to this comparison.

Table 3.1 A taxonomy of the lattice-based signature schemes.

Property	Falcon		Dilithium		BLISS-B	
NIST submission	✓		✓		✗	
Error Sampling Type	Gaussian		Uniform		Gaussian	
Ring Type	$\mathbb{Z}_q[x]/(x^N+1)$		$\mathbb{Z}_q[x]/(x^{256}+1)$		$\mathbb{Z}_q[x]/(x^N+1)$	
Sig. Scheme Type	Hash-and-sign		Fiat-Shamir		Fiat-Shamir	
Security Assumptions	NTRU Problem (variant)		Module-LWE/SIS		(NTRU) Ring-SIS	
Classical Security of Parameter Sets	Set-I	114	Set-I	58	Set-I	128
	Set-II	195	Set-II	100	Set-II	128
	Set-III	263	Set-III	138	Set-III	160
	-	-	Set-IV	174	Set-IV	192

3.2.1 BLISS-B

The BLISS signature scheme was proposed by Ducas et al. [2013b], and was further improved upon by Ducas to create BLISS-B [Ducas, 2014c]. The base distribution for sampling is a bimodal Gaussian which improves the rejection rate and therefore performance. The BLISS-B variant uses a different signature algorithm to BLISS along with modifications of geometric bounds and removal of the rejection stage during key generation. However, the BLISS-B parameters related to security are unmodified from BLISS. The BLISS-B components are provided in Algorithms 2, 3, and 4, as proposed by the authors in Ducas [2014c]. Keygen (Algorithm 2) produces a public key A and small private polynomials $(\mathbf{s_1}, \mathbf{s_2})$, Signing (Algorithm 3) uses $(\mathbf{s_1}, \mathbf{s_2})$ to mask Gaussian-sampled polynomials and Verifying (Algorithm 4) checks that the received signature falls between certain bounds and that $\mathbf{c} = H(\lfloor \mathbf{a}_1 \cdot \mathbf{z}_1 + q \cdot \mathbf{c} \rceil_d + \mathbf{z}_2' \mod p, \mu)$ holds, where $\mathbf{c}, \mathbf{z_1}, \mathbf{z_2}$ are signature components, μ is the message and $\mathbf{a_1}, q$ are public.

Pessl et al. [2017] presented a key recovery attack against BLISS, which also applies to BLISS-B. A cache attack on the signing procedure recovers some coefficients of the Gaussian noise vector $\mathbf{y}$, which allows derivation of the private key polynomial $\mathbf{s_1} = \mathbf{f}$. From this, the entire key can be recovered. Countermeasures may be employed, particularly at the sampling stage, to prevent information about the noise vector being obtained to allow the attack. Pessl et al. present the attack both in the ideal situation of a plain Cumulative Distribution Table (CDT) sampler, and in the situation where shuffling has been implemented. To achieve a success rate of 90% in recovering $\mathbf{s} \mod 2$, the CDT attack requires 325 signatures, but when shuffling is introduced, 70,000 signatures are required for an idealised attacker. Another notable attack against BLISS is by [Bruinderink et al., 2016b], targeted at the Gaussian sampling component.

It is able to retrieve the secret key using only 3500 signatures on a PoC implementation. This research addresses these attacks by proposing a software design that integrates multiple countermeasures, namely verify-after-sign, sample discard (at rates of 6.5%, 12.5% and 25%), Fisher-Yates shuffling [Fisher et al., 1938], and VectorBlindSample [Saarinen, 2017], which are further described in Section 3.4.

Algorithm 2: The BLISS-B key generation algorithm [Ducas, 2014c]

Data: N, q, α
Result: $(\mathbf{A}, \mathbf{s})$ such that $\mathbf{A}\mathbf{s} = q \mod 2q$

1 Uniformly-chosen polynomials $\mathbf{f}, \mathbf{g}$ with $d_1 = \lceil \delta_1 N \rceil$ coefficients ± 1 and $d_2 = \lceil \delta_2 N \rceil$ coefficients ± 2
2 $\mathbf{s} = (\mathbf{s_1}, \mathbf{s_2})^{\mathbf{t}} \leftarrow (\mathbf{f}, 2\mathbf{g} + 1)^t$
3 $\mathbf{a}_q = (2\mathbf{g} + 1)/f \mod q$
4 **return** $(\mathbf{A}, \mathbf{s})$ *where* $\mathbf{A} = (2\mathbf{a}_q, q - 2) \mod 2q$

Algorithm 3: The BLISS-B signing algorithm [Ducas, 2014c]

Data: Message μ, public key $\mathbf{A} = (\mathbf{a}_1, q - 2) \in \mathcal{R}_q^{1\times 2}$, private key $\mathbf{s} = (\mathbf{s_1}, \mathbf{s_2})^{\mathbf{t}} \in \mathcal{R}_{2q}^{2\times 1}$
Result: Signature $\mathbf{z}_1, \mathbf{z}_2', \mathbf{c}$ of message μ

1 $y_1, y_2 \leftarrow \mathcal{D}_{\mathbb{Z}^N,\sigma}$
2 $\mathbf{u} = \mathbf{a}_1 \cdot \mathbf{y}_1 + \mathbf{y}_2 \mod 2q$
3 $\mathbf{c} \leftarrow H(\lfloor \mathbf{u} \rceil_d \mod p, \mu)$
4 $(\mathbf{v}_1, \mathbf{v}_1) \leftarrow$ GreedySC$(\mathbf{s}, \mathbf{c})$
5 Choose a random bit b
6 $(\mathbf{z}_1, \mathbf{z}_2) \leftarrow (\mathbf{y}_1, \mathbf{y}_2) + (-1)^b \cdot \mathbf{v}_1, \mathbf{v}_2)$
7 Continue with probability $1/M \ \exp(\frac{||\mathbf{v}||^2}{2\sigma^2}) \cosh(\frac{\langle \mathbf{z}, \mathbf{v} \rangle}{\sigma^2}))$ otherwise restart
8 $\mathbf{z}_2' \leftarrow (\lfloor \mathbf{u} \rceil_d - \lfloor \mathbf{u} - \mathbf{z_2} \rceil_d) \mod p$
9 **return** $(\mathbf{z}_1, \mathbf{z}_2', \mathbf{c})$

3.2.2 Dilithium

The Dilithium signature scheme was introduced by Ducas et al. [2018]. It is also a Fiat-Shamir type signature scheme, but was intended to be more SCA resistant than

Algorithm 4: The BLISS-B verifying algorithm [Ducas, 2014c]

Data: Message μ, Public key $\mathbf{A} = (\mathbf{a}_1, q-2) \in \mathcal{R}_q^{1\times 2}$, Signature $\mathbf{z}_1, \mathbf{z}_2', \mathbf{c}$
Result: Accept or Reject

```
if ||z_1|2^d · z_2'||_2 > B_2 then
    return Accept
if ||z_1|2^d · z_2'||_inf > B_inf then
    return Accept
else
    return Reject
Accept iff c = H(⌊a_1 · z_1 + q · c⌉_d + z_2' mod p, μ)
```

BLISS and therefore samples noise from a uniform distribution rather than a Gaussian. The signature essentially constructs module-LWE samples with uniformly sampled noise. The size of the public key is reduced by dropping low-order bits. The key generation, signing and verifying components for the scheme are given in Algorithms 5, 6, 7 respectively.

Bruinderink and Pessl [2018] present a differential fault attack on Dilithium, assuming the attacker has the capability of injecting a single random fault. They consider different scenarios, all of which require the introduction of a single random fault, such as altering a polynomial coefficient during multiplication, or corrupting the seed in the expansion of the lattice basis matrix. The attack recovers part of the private key, which can be used to forge signatures. They show that 65.2% of the signing algorithm code is vulnerable to at least one of these attacks, given they are performed in the first iteration. The authors recommended the countermeasures double computation and verify-after-sign, and these have been adopted in the designs presented here for evaluation.

Algorithm 5: The Dilithium key generation algorithm [Ducas et al., 2018]

Data: $k, l, q, d, \eta, \beta, \gamma_1, \gamma_2$
Result: 2

1 $\rho, \rho' \leftarrow \{0,1\}^{256}$
2 $\mathbf{A} \in \mathcal{R}_q^{k \times l} := \text{Sam}(\rho)$
3 $(\mathbf{s_1}, \mathbf{s_2}) \in \mathcal{S}_\eta^l \times \mathcal{S}_\eta^k := \text{Sam}(\rho')$
4 $\mathbf{t} := \mathbf{A}\mathbf{s_1} + \mathbf{s_2}$
5 $\mathbf{t}_1 := \text{Power2Round}_q(\mathbf{t}, d)$
6 **return** Public key $= (\rho, \mathbf{t}_1)$, Private key $= (\rho, \mathbf{s_1}, \mathbf{s_2}, \mathbf{t})$

3.2.3 Falcon

The Falcon hash-and-sign signature scheme was proposed by Prest et al. [2017]. It is based on the DLP identity-based encryption scheme proposed by Ducas et al. [2014a], together with fast Fourier sampling techniques suggested by Ducas and Prest [2016a]. Operations are over the polynomial ring $\mathcal{R} = \mathbb{Z}_q[x] \big/ \langle x^N + 1 \rangle$, a variant of the NTRU ring. It improves on the compactness of the private key and speed of the sampling procedure, and is shown to be secure in the quantum random oracle model. The key generation, signing, sampling and verifying components for the scheme are given in Algorithms 8, 9, 10 and 11 respectively. The hash functions are instantiated as discussed in §2.5.1. Following the NIST Round 2 candidate announcements, parameter Set II has been removed from the Falcon specification [Prest et al., 2019] and therefore is also omitted from the remainder of this thesis. The removal of this parameter set was justified as it made the specification more straightforward, reducing the algorithm count from 22 to 14, and the remaining parameter sets shared a common modulus ($q = 12289$) and underlying ring $\mathbb{Z}[x]/\langle x^n + 1 \rangle$, allowing for simpler implementation and potential deployment of the signature scheme.

Algorithm 6: The Dilithium signing algorithm [Ducas et al., 2018]

Data: Private key $= (\rho, s_1, s_2, t), \mu \in \mathcal{M}$
Result: Signature $\sigma = (z, h, c)$

1 Compute $A = \text{Sam}(\rho)$ and $t_1 = \text{Power2Round}_q(t, d)$
2 $t_0 := t - t_1 \cdot 2^d$
3 $r \leftarrow \{0,1\}^{256}$
4 $y \in \mathcal{S}^l_{\gamma_1 - 1} := \text{Sam}(r)$
5 $w := Ay$
6 $w_1 := \text{HighBits}_q(w, 2\gamma_2)$
7 $c := H(\rho, t_1, w_1, \mu)$
8 $z := y + cs_1$
9 $(r_1, r_0) := \text{Decompose}_q(w - cs_2, 2\gamma_2)$
10 **if** $||z||_\infty \geq \gamma_1 - \beta$ *or* $||r_0||_\infty \geq \gamma_2 - \beta$ *or* $r_1 \neq w_1$ **then**
11 go to 3
12 $h := \text{MakeHint}_q(-ct_0, w - cs_2 + ct_0, 2\gamma_2)$
13 **if** $||ct_0||_\infty \geq \gamma_2$ *or the number of 1's in* $h > \omega$ **then**
14 go to 3
15 **return** $\sigma := (z, h, c)$

To date, there have been no physical attacks reported against Falcon. Previous research has mounted differential power analysis attacks on NTRU-based schemes [Atici et al., 2008, Lee et al., 2010, Wang et al., 2013]. However, due to the different structure of Falcon, these attacks are not directly applicable. The attack in Atici et al. [2008] targets the NTRU decryption stage, which Falcon does not possess. The attack in Lee et al. [2010] targets the convolutional product of the NTRU polynomials, in particular the power variances observed during the repeated addition step. Although Falcon is based on NTRU polynomials, the polynomial multiplication is performed in the NTT domain, in point-value form, and so does not possess this susceptible step. Recent attacks have targeted the polynomial multiplication component of BLISS [Espitau et al., 2017] using electromagnetic analysis and Dilithium [Ravi et al., 2018] via power analysis, which could potentially apply to the corresponding component of

Algorithm 7: The Dilithium verify algorithm [Ducas et al., 2018]

Data: Public key $= (\rho t_1), \mu \in \mathcal{M}, \sigma = (z, h, c)$
Result: Accept or Reject

1 $A :=$ Sam(ρ)
2 $w_1 :=$ UseHint$_q(h, Az - ct_1 \cdot 2^d, 2\gamma_2)$
3 **if** $c = H(\rho, t_1, w_1, \mu)$ *and* $||z||_\infty < \gamma_1 - \beta$ *and the number of 1's in* $h \leq \omega$ **then**
4 **return** Accept
5 **else**
6 **return** Reject

Algorithm 8: Falcon key generation (adapted from Prest et al. [2017])

Data: N, q
Result: $\mathbf{B} \in \mathbb{Z}_q^{2N \times 2N}$, $\mathbf{h} \in \mathcal{R}_q$

1 $\sigma_{\mathbf{f}} = 1.17\sqrt{\frac{q}{2N}}$
2 $\mathbf{f}, \mathbf{g}, \leftarrow \mathcal{D}_{N,\sigma_{\mathbf{f}}}$
3 Norm $\leftarrow max\left(||\mathbf{g}, -\mathbf{f}||, \left|\left|\left(\frac{q\bar{\mathbf{f}}}{\mathbf{f}*\bar{\mathbf{f}}+\mathbf{g}*\bar{\mathbf{g}}}, \frac{q\bar{\mathbf{g}}}{\mathbf{f}*\bar{\mathbf{f}}+\mathbf{g}*\bar{\mathbf{g}}}\right)\right|\right|\right)$
4 **if** *Norm* $> 1.17\sqrt{q}$ **then** go to Step 2;
5 Compute $\rho_{\mathbf{f}}, \rho_{\mathbf{g}} \in \mathcal{R}$ and $R_{\mathbf{f}}, R_{\mathbf{g}} \in \mathbb{Z}$ such that: $\rho_{\mathbf{f}} \cdot \mathbf{f} = R_{\mathbf{f}}$ and $\rho_{\mathbf{g}} \cdot \mathbf{g} = R_{\mathbf{g}}$
6 **if** $GCD(R_{\mathbf{f}}, R_{\mathbf{g}}) \neq 1$ *or* $GCD(R_{\mathbf{f}}, q) \neq 1$ **then** go to Step 2;
7 Compute $u, v \in \mathbb{Z}$ such that: $u \cdot R_{\mathbf{f}} + v \cdot R_{\mathbf{g}} = 1$
8 $F \leftarrow qv\rho_{\mathbf{g}}$ and $Q \leftarrow -qu\rho_{\mathbf{f}}$
9 $k = \left\lfloor \frac{F*\bar{\mathbf{f}}+G*\bar{\mathbf{g}}}{\mathbf{f}*\bar{\mathbf{f}}+\mathbf{g}*\bar{\mathbf{g}}} \right\rceil \in \mathcal{R}$
10 $F \leftarrow F - k * \mathbf{f}$ and $G \leftarrow G - k * \mathbf{g}$
11 $\mathbf{h} = \mathbf{g} * \mathbf{f}^{-1} mod\, q$
12 $\mathbf{B} = \begin{pmatrix} \mathbf{g} & -\mathbf{f} \\ G & -F \end{pmatrix}$
13 $\hat{\mathbf{B}} = \begin{pmatrix} FFT(\mathbf{g}) & -FFT(\mathbf{f}) \\ FFT(G) & -FFT(F) \end{pmatrix}$
14 T=ffLDL*(G)
15 $\sigma \leftarrow 1.55\sqrt{q}$
16 for each leaf of T, leaf.value $= \sigma/\sqrt{\text{leaf.value}}$
17 sk $\leftarrow (\hat{\mathbf{B}}, \text{T})$, pk $= \mathbf{h}$
18 **return** sk, pk

Falcon. Furthermore, the Falcon trapdoor sampler calls a discrete integer Gaussian sampler, which has been the subject of timing and power attacks [Bruinderink et al.,

Algorithm 9: FALCON signing (adapted from Prest et al. [2017])

Data: sk, $H : \{0,1\}^* \to \mathbb{Z}_q^N$, m, β
Result: SIG_m

```
if SK_m is in local storage then
    return Output SIG_m to message m
else
    r ← Unif({0,1}^320)
    c ← H(r||m) ;
    t ← (FFT(c), FFT(0)) · B̂^-1 z ← ffSampling(t, T)
    s = (t − z)B̂
    while ||s|| > β
    (s_1, s_2) ← invFFT(s)
    s_2 = Compress(s_2)
    SIG_m ← (r, s_2)
    return Output SIG_m associated to message m and keep in local storage
```

2016a,b, Pessl et al., 2017, Pessl, 2016a, Primas et al., 2017], as described in the next section.

However, this research focusses on a fault attack on FALCON. It is logical to draw comparisons between the DLP scheme's GPV sampler and the FALCON sampler, as the latter is a recursive form of the former (introduced in §2.5.5). As an existing fault attack on the DLP scheme's sampler has been reported in the literature [Espitau et al., 2016, 2018], it is a natural step to adapt this attack to FALCON.

The attack by Espitau et al. [2016, 2018] on the GPV-based DLP hash-and-sign signature scheme assumes the attacker is able to abort a loop early in the GPV Gaussian lattice-sampling stage in the signing algorithm. Whilst sampling vector coefficients of a signature polynomial of degree $2N$, they force the process to terminate at some iteration $m \leq N$, and obtain a faulty signature in the sub-lattice of rank m. After producing around $m + 3$ faulty signatures, they can generate the original lattice and recover a short vector with probability of 86%, thereby recovering enough of the

Algorithm 10: ffSampling Prest et al. [2017]

Data: $\mathbf{t} = (\mathbf{t_0}, \mathbf{t_1})$ in FFT format , T
Result: $\mathbf{z} = (\mathbf{z_0}, \mathbf{z_1})$ in FFT format

```
if n = 1 then
    σ' ← T.value
    z_0 ← D_{Z,t_0,σ'}
    z_1 ← D_{Z,t_1,σ'}
    return z = (z_0, z_1)
else
    (l, T_0, T_1) ← (T.value, T.leftchild, T.rightchild)
    t_1 ← splitfft(t_1)
    z_1 ← ffSampling(t_1, T_1)
    z_1 ← mergefft(z_1)
    t_0' ← t_0 + (t_1 − z_1) · l
    t_0 ← splitfft(t'_0)
    z_0 ← ffSampling(t_0, T_0)
    z_0 ← mergefft(z_0)
    return z = (z_0, z_1)
```

NTRU lattice structure to obtain the private key. Specifically, they are able to recover a sub-lattice of the lattice generated by private key polynomial F, which is introduced in §2.4.2. This allows the entire private key to be recovered. Espitau et al. recommend checking the validity of the DLP signature before sending, by running the verification algorithm. In Section 3.3, this attack is adapted for the Falcon signature scheme.

3.3 Bearz attack Falcon

This section proposes a new attack on the Falcon digital signature scheme, named Bearz (Basis Extraction by Aborting Recursion or Zeroing). The attack exerts and then exploits faults in the Falcon signature scheme in order to learn private key information, following a similar attack used on DLP [Espitau et al., 2016, 2018], described earlier

Algorithm 11: FALCON verifying (adapted from Prest et al. [2017])

Data: pk $= h \in \mathcal{R}_q$, (SIG_m, m), β
Result: Accept or Reject

```
c ← H(r||m)
s_2 = Decompress(s_2), s_1 ← c − s_2 h  mod q
if ||(s_1, s_2)|| < β and H(r||m) = c then
    return Accept
else
    return Reject
```

in this section. BEARZ utilises a different attack model, assuming that the attacker is able to skip instructions and to zero variables as assumed by Bindel et al. [Bindel et al., 2016] on other lattice-based signature schemes. The original attack is on the GPV sampler component in DLP, however there are two major differences between this sampler and the FALCON sampler: the use of *recursion* and the *FFT domain*.

To begin, a recap of the attack on DLP [Espitau et al., 2016, 2018] is given. In the GPV lattice trapdoor sampler, coefficients of the N-dimensional vectors (R, r) are sampled in a $2N$-length loop in reverse, from the last coefficient r_N to the first coefficient R_0. This is converted to a short Gaussian-distributed lattice vector by multiplying the vector (R, r) by the basis:

$$\mathbf{B} = \begin{pmatrix} \mathcal{A}(\mathbf{g}) & -\mathcal{A}(\mathbf{f}) \\ \mathcal{A}(G) & -\mathcal{A}(F) \end{pmatrix},$$

where $\mathcal{A}(\cdot)$ is the anti-circulant matrix associated with the degree-N polynomial, $\cdot$. This matrix is obtained as discussed in §2.4.2. The second half of the signature, denoted $\mathbf{s_2} = (R, r)\,(-\mathcal{A}(\mathbf{f}), -\mathcal{A}(F)) = R \cdot \mathbf{f} + r \cdot \mathbf{F}$, is a $2N$-dimensional vector.

The fault attack is a loop abort on the $2N$-loop which samples the (R, r) coefficients. It causes the loop to abort at the mth iteration, where $m \leq N$. This effectively

eradicates the vector R, so $\mathbf{s_2}$ becomes: $\mathbf{s_2} = (0, \ldots, 0, r_0, \ldots, r_{m-N})F$, or alternatively the polynomial $\mathbf{s_2} = r_0 x^{N-1} F + r_1 x^{N-2} F + \cdots + r_{m-1} x^{N-m} F$ which is in the sub-lattice of $\mathcal{L}(\mathcal{A}(F))$. With a certain number of faulty signatures, $l \approx m$, one will obtain a set spanning the lattice. From this, lattice reduction can be used to find a short vector, which should be one of the signed shifts of F, from which the private basis can be recovered. This attack can be built upon for Falcon.

3.3.1 Recursion in Falcon

The Falcon sampling algorithm, shown in Algorithm 10, can be viewed as a recursive form of the GPV sampler. The attack notation is modified to match the original Falcon specification, where (R, r) in the DLP scheme becomes $\mathbf{z} = (\mathbf{z_0}, \mathbf{z_1})$ in Falcon. The DLP attack inserts a loop abort fault at some iteration, m, of the $2N$ loop [Espitau et al., 2016, Fig. 2], however in Falcon this translates to aborting the recursive call early [Prest et al., 2017, Alg. 18]. To do this, examine the structure of the recursion. For the instance $2N = 1024$, this is shown in Figure 3.1. There are two recursive calls in the top level of the algorithm. Given the target vector $(\mathbf{t_0}, \mathbf{t_1})$, it acts first on $\mathbf{t_1}$ recursively, from right to left, and then on $\mathbf{t_0}$. Each of $\mathbf{t_1}$, and then $\mathbf{t_0}$, continuously splits into two vectors of length $N := N/2$ until $N = 2$, and then samples the coefficients from a Gaussian distribution, before building back up to give sample vector $(\mathbf{z_0}, \mathbf{z_1})$. Hence there are two recursive branches within the top level algorithm. Also, this is all performed within the FFT domain. For a vector of length N, the first $N/2$ values represent the real coefficients and the second $N/2$ represent the imaginary coefficients. So, the initial approach would be to terminate the recursive call at the required point

so that only $m - N$ spaces would be filled, but this would not work due to the nature of the FFT functions as explained in the next section.

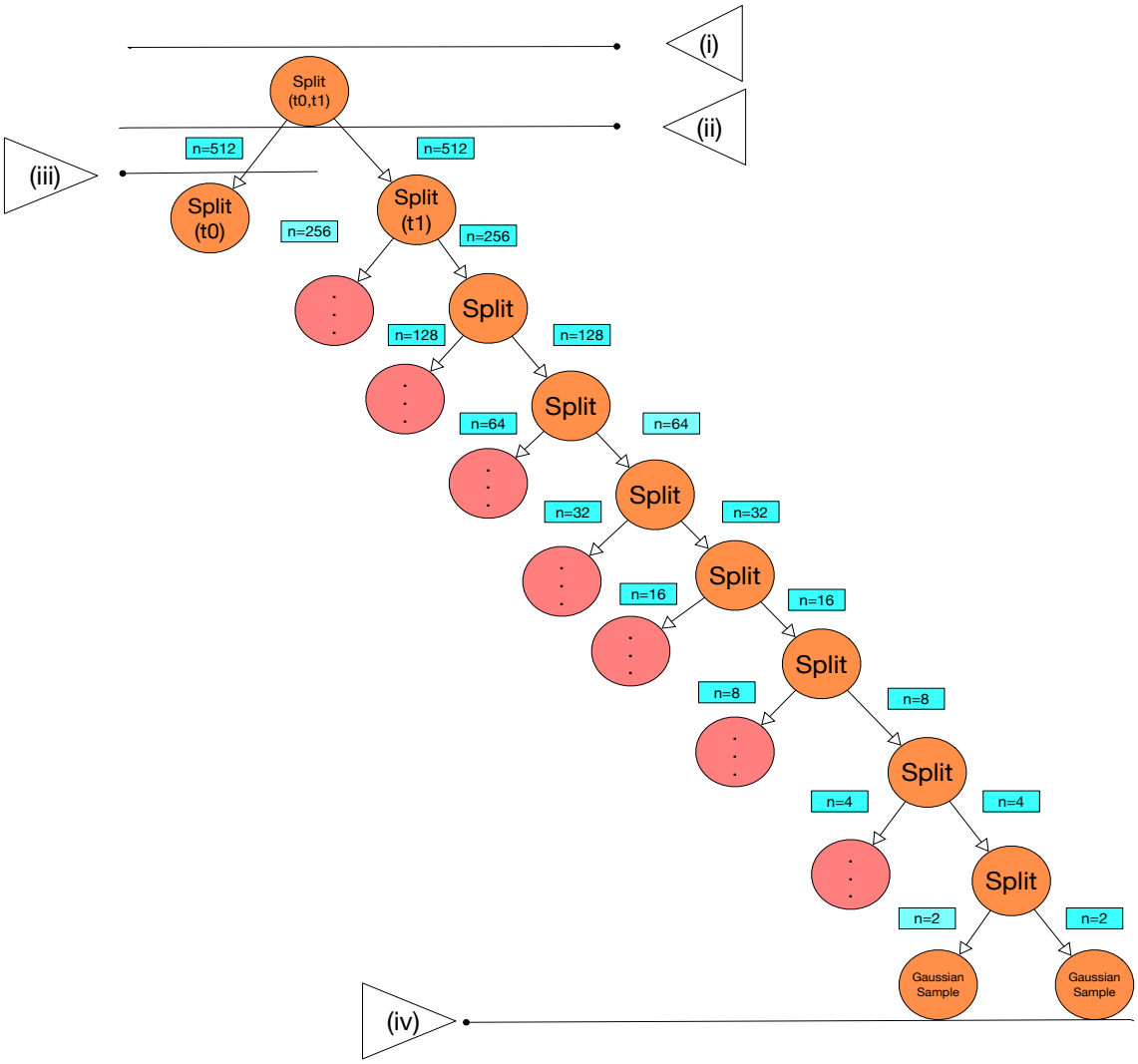


Fig. 3.1 The recursion tree of the FFT Sampler and Bearz attack points $(n = N)$

3.3.2 FFT: Merging and Splitting

The sampling algorithm calls merging and splitting functions as subroutines. The purpose of these functions is to map the polynomials from $\mathcal{R} = \mathbb{Z}_q[x] \big/ \langle \phi \rangle$, where

$\phi = x^N + 1$, into smaller polynomials over the ring $\mathcal{R} = \mathbb{Z}_q[x] \big/ \langle \phi' \rangle$, where $\phi(x) = \phi' = (x^2)$. The computations can then take place in the smaller ring, which is more computationally efficient. In the FFT sampler, this is done repeatedly until the operations are taking place over the integers. Furthermore, this allows the field norm to be easily be expressed as $N(\mathbf{f}) = \mathbf{f}_0^2 + x\mathbf{f}_1^2$, where $split(\mathbf{f}) = (\mathbf{f}_1, \mathbf{f}_1)$. These functions do not simply break up and concatenate the polynomial ring elements. The split function separates a polynomial into its even and odd coefficients and is performed in the FFT domain. Operating in the FFT domain provides a barrier to the fault attack, since a zero input does not translate to zero in the FFT domain. Thus, to apply the attack successfully, the coefficients of the samples must be tracked as they move up through the recursive tree, in order to determine the sampled coefficients to target for the attack. This pathway is the same for all invocations of the sampling algorithm, so it is stated here which target coefficients which need altered. Once the lattice vector has been obtained from the Gaussian sample, the FFT^{-1} function is applied to it, and the signing and verifying procedures are the same as in DLP. This means the same post-processing can be used from the DLP attack, however it must be ensured that the signature is not in the FFT domain by applying the inverse FFT function. Figure 3.2 shows the differences between the attack strategies.

3.3.3 Method of attack

Different methods of attack are now presented, differentiated by the point at which the abortion occurs, this being the m positions from the end of the vector. All attack methods lead to the same output vector format for z; the first $(2N - m)$ coefficients of $z = (z_0, z_1)$ are forced to zero.

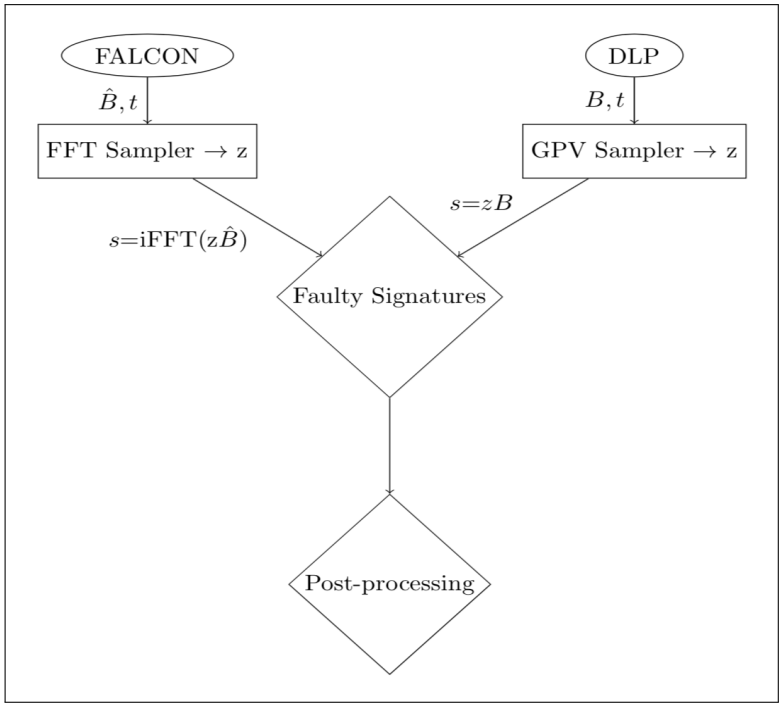


Fig. 3.2 The strategies of the Bearz and DLP attacks

(i) Abort second recursion (for $m = N$)

This attack can be performed in a straightforward manner. Simply abort at the top sampling algorithm call, after the first recursive call, after line 10 of Algorithm 10. Therefore $\mathbf{z_1}$ gets filled with sampled coefficients, yet $\mathbf{z_0}$ remains all zeros. This fault abortion is also performed for the case where $m \leq N$, in order to zero the first N coefficients.

(ii) and (iii) Zeroing or skipping attack (for $m \leq N$)

There are two options for this attack. First, at the penultimate merging step ($N = 256$ to 512), set the required coefficients (e.g., the first half if $m = N/2$) of the output coefficients to zero by skipping operations or later zeroing the corresponding

coefficients. The skipped operations are

$$f[(u << 1) + 0] = t_re \tag{3.1}$$

$$f[(u << 1) + 1] = t_re, \tag{3.2}$$

from the reference code, specifically, the falcon_poly_merge_fft function from falcon_fft.c in Appendix A [Prest et al., 2017]. Otherwise, one can set the first required number of coefficients of z_1 to zero before computing the corresponding lattice vector, that is, overwrite the sampler output $\mathbf{z_1}$.

(iv) Aborting mid-recursion ($m \leq N$)

This attack is more sophisticated, the coefficients must be tracked throughout the tree. Although this requires a prior one-off computation, it allows the fault to be applied at the one-dimensional Gaussian sampler stage, which is potentially an easier point to exploit.

Suppose the target is $m = N/2$; the left half of $\mathbf{z_1}$ is required to be zero. For this to occur, the Left Hand Side (LHS) vector in the final falcon_poly_merge_fft call has to have its first half equal to zero, and the right-hand side RHS vector has to have the first half of its coefficients equal to the second half.

Now each real coefficient (the first half) of the z_1 vector is generated by

$$\mathbf{z_1}[2u] = f_0[u] + (f_1[u] - f_1[u + N/4]) \tag{3.3}$$

$$\mathbf{z_1}t[2u + 1] = f_0[u] - (f_1[u] - f_1[u + N/4]) \,, \tag{3.4}$$

where N is the length of the higher-level vector, and $u \in \{0, \ldots, \frac{N}{4} - 1\}$.

Note, constants have been removed from the equations as they are irrelevant for the result.

For $\mathbf{z_1}[2u]$ and $\mathbf{z_1}[2u+1]$ to be zero, it can be set:

$$f_0[u] = 0 \tag{3.5}$$

$$f_1[u] = f_1[u + n/4] \tag{3.6}$$

for each $u \in \{0, \ldots, \frac{N}{4} - 1)$. So for $n = 512$, the first $\frac{N}{4} = 128$ coefficients of the LHS 256-length vector are set to equal zero, and the first 128 coefficients of the RHS 256-length vector are set to equal the second 128 coefficients.

For the LHS 256-length vector, the first half of its coefficients must equal zero. The LHS 128-vector must have its first half equal to zero, and the RHS 128-vector must have the first half of its values equal to the second half of its values. This process is repeated on the LHS 64-length vector and on down the tree. The RHS 256-length vector must have the first half coefficients equal to the second half. For this to occur, the LHS 128-length vector on this side must have the real values equal to the imaginary values, and the corresponding RHS 128-length vector must equal zero.

This conclusion is reached by considering $\mathbf{f}[2u] = \mathbf{f}[2u+N/2]$ (first real/imaginary computation per iteration) and $\mathbf{f}[2u+1] = \mathbf{f}[2u+1+\frac{N}{2}]$ (second real/imaginary) to be equal for each iteration of u.

From the falcon_poly_merge_fft function, this means

$$f_0[u] + (f_1[u] - f_1[u + n/4]) = f_0[u + n/4] + (f_1[u] + f_1[u + n/4]) \tag{3.7}$$

$$f_0[u] - (f_1[u] - f_1[u + n/4]) = f_0[u + n/4] - (f_1[u] + f_1[u + n/4])\,. \tag{3.8}$$

To satisfy the components of the equations, set $\mathbf{f_0}[u] = \mathbf{f_0}[u + N/4]$, i.e. the first half of coefficients equals the second half, and for the second component, set $\mathbf{f_1}[u] = \mathbf{f_1}[u + N/4] = 0$. Therefore the equations will simply depend on the LHS feed-in vector.

For the RHS, every branch below can be set equal to zero. Take the LHS 128-length vector. It follows the same conditions as the level above, i.e. the LHS 64-length vector must have its real and imaginary values equal, and the RHS must be zero. Any vector equalling zero must have both feed-in vectors equalling zero, so the branches below this can be zeroed. This method can be applied for any m such that $m = 2^k$, where $k \in \mathbb{Z}$.

3.3.4 Post-attack processing

The final step of the attack involves recovery of the private basis from the faulty signatures. Suppose the first $2N - m$ coefficients of vector $(\mathbf{z_0}, \mathbf{z_1})$ are equal to zero. Then, according to the signing algorithm of Falcon (Algorithm 9), the signature $\mathbf{s_2}$ is computed as:

$$\mathbf{s} = \mathbf{t} - \mathbf{z}\hat{\mathbf{B}} \tag{3.9}$$

where $\hat{\mathbf{B}}$ is the basis matrix in the FFT domain. But only the second half of s is needed to be zero, that is s_2, where

$$\mathbf{s_2} = \mathbf{t_1} - \mathbf{z}\left(\frac{-\mathcal{A}(\mathbf{f})}{-\mathcal{A}(F)}\right), \tag{3.10}$$

and since $\mathbf{t_1}$ is set to zero,

$$\mathbf{s_2} = -\mathbf{z}\left(\frac{-\mathcal{A}(\mathbf{f})}{-\mathcal{A}(F)}\right). \tag{3.11}$$

Furthermore, the first $m - N$ coefficients of $\mathbf{z}$ are zero, which further implies:

$$\mathbf{s_2} = -\mathbf{z_1}\left(-\mathcal{A}(F)\right), \tag{3.12}$$

and since some of the coefficients of $\mathbf{z_1}$ are zero, it becomes:

$$\mathbf{s_2} = (\mathbf{z_1}[m-1]x^{N-m}F + \cdots + \mathbf{z_1}[0]x^{N-1}F). \tag{3.13}$$

This leaves us with a sub-lattice of the lattice generated by F, akin to that obtained in the DLP attack. The only difference is that the vectors are in the FFT domain, which can be overcome by applying the inverse FFT function to the output. Thus, with multiple faulty signatures, the lattice spanned by F (where F is a short vector in this lattice) can be found, and the BKZ algorithm can be used to retrieve it. Then $G, \mathbf{f}$, and $\mathbf{g}$ can be deduced from the public key $\mathbf{h}$, and thus the private NTRU lattice basis can be found. Firstly, G can be recovered from F due to the relation $G = F \cdot \mathbf{h}$. Then the NTRU equation $\mathbf{f} \cdot G - \mathbf{g} \cdot F = -q$ allows $\mathbf{f}$ and $\mathbf{g}$ to be recovered using the basis completion algorithm from [Hoffstein et al., 2003].

3.3.5 Attack results

To verify this attack model, each method of attack was first tested through software simulation and ensured the output signature gave the expected number of zeros. This was successful for all attack methods and parameter sets. A total of l faulty signatures were collected for the values of m zeros given in Table 3.2, and the BKZ algorithm [FPLLL Development Team, 2016] was run to obtain the private key polynomial F. The timings in seconds for the BKZ algorithm to obtain F are given in Table 3.2. It

can be seen that the secret key is recovered within a reasonable amount of attacking time, for instance only 2.67 seconds for the high security parameter set with 1/4 of the coefficients set to zero, and 11.8 seconds for parameter set I with 3/4 of the coefficients set to zero. Bindel et al. [2016] consider a range of fault models against BLISS [Ducas et al., 2013a], Ring-Tesla [Akleylek et al., 2016] and the GLP scheme [Güneysu et al., 2012b], including randomising the modulus, zeroing a hash function and skipping addition. The assumptions and attack model used here are similar, in particular for estimating the number of faults required. It can be concluded from these results that the post-processing of the faulty signatures following the fault injection takes negligible effort.

Table 3.2 Timings (in seconds) to run the BKZ algorithm on l faulty signatures to obtain the basis polynomial F running on a single core of an Intel Core i7-6700HQ CPU at 2.60 GHz.

Parameters	m	$l = m + 1$	$l = m + 2$	$l = m + 3$
Set-I	64	0.07	0.07	0.08
	128	0.35	0.30	0.36
	256	2.17	2.80	2.36
	385	10.9	12.8	11.8
Set-III	128	0.28	0.29	0.35
	256	2.24	2.55	2.67

3.3.6 Fault Model and Feasibility of Attack

This section investigates how an attacker may obtain the faulty signatures for the proposed attacks. This is based on the SAFEcrypto [SAFEcrypto, 2018] software implementation which incorporates Prest's FFT Sampling PoC code, given in Appendix A, and demonstrates that future implementations for real-world use should incorporate

protection. Faults are commonly introduced by varying the heat, voltage or clock frequency, or subjecting the device to an electromagnetic radiation beam.

The four methods of attack used by Bearz are:

(i) **Zero** the first $2N - m$ coefficients of output $\mathbf{z} = (\mathbf{z_0}, \mathbf{z_1})$ at the final step

(ii) **Abort** second recursive call at top level on $\mathbf{t_0}+$

(iii) **Skip** or **zero** chosen steps in final merge function call to ensure required $\mathbf{z_1}$ coefficients are 0 (together with abort of second recursive call on $\mathbf{t_0}$)

(iv) **Abort** certain samples at the one-dimensional Gaussian sampling stage (these have been determined by “tracking coefficients")

This reduces to three possible types of fault injections:

a) **Skipping** function calls

b) **Zeroing** specific vector coefficients, which translates to zeroing specific bit strings

c) **Aborting** at a specific point

The limitations of performing this attack are the *preciseness* of the faults required. In the comparable attack on the DLP sampler in [Espitau et al., 2016, 2018], all that was required was a random fault injection to abort the loop within a required range, yet Bearz requires specific bit strings and function calls to be targeted due to the scrambling of the merge function.

The physical attack methods required for Bearz are early abortion, zeroing and skipping lines of code. With respect to the fault models guide [Verbauwhede et al.,

2011], the Bearz fault attack models can be classified as being on the processing part (for zeroing) and program flow (for skipping and aborting). These are high precision attacks, where the adversary has the ability to bit flip/set, and targets the cryptographic primitive.

The concept of loop abort faults was first introduced for pairings-based cryptography by Page and Vercauteren [2006], in which the authors considered a case where the loop was glitched so it was performed once too often. In the original attack on the DLP sampler [Espitau et al., 2018], the authors state that a random fault on the loop counter or skipping the jump operation will lead to the required result for their attack. Side-channel analysis can be used to detect the loop iteration pattern in order to trigger the fault injection. They state that such pattern is likely to be detected after much less than 50 iterations and thus it seems that the synchronisation here will be relatively easy. The time-window within which the fault can be applied is large due to the loose condition on the number of known coefficients in the faulted polynomials.

In comparison, for Bearz, side-channel analysis would be used to detect the window between the first call and the second recursive call, within which the algorithm can be aborted. Although there is ample time to detect the iterative pattern, the window within which the abortion would need to happen is much tighter. However, it is sufficient to tamper with the counter at any time during the execution of the second recursive call, before the corresponding half of the top level vector has been populated. With that strategy, if this second call takes sufficient time to execute, the timing of the fault injection need not be excessively precise. If targeting a public implementation, the approximate time of fault injection can be easily deduced. An example of where a similar fault has been injected is in [Gélin and Wesolowski, 2017], which targets an isogeny-based cryptosystem.

Table 3.3 Corresponding attacks and countermeasures

Countermeasure	**Attack prevention**
Double Computation	Fault attacks in signing
Verify-after-sign	Fault attacks in signing (faults in sampling may be missed)
Sample Discard	Attacks on Gaussian Sampler
Shuffling/Blinding	Timing and power attacks on Gaussian Sampler
Constant-time	Timing attacks (sampler and polynomial arithmetic)
Zero-check	Bearz

A zeroing attack can be performed as a memory-based attack, during storage of the vector in RAM, by setting the required bits to zero [Naccache et al., 2005]. In [Ordas et al., 2014], electromagnetic beam injection was performed with enhanced probes and was able to produce very local bit-set and bit-reset faults.

Finally, skipping lines of code can be performed by CPU clock glitching. In [Blömer et al., 2014] the authors build a setup which precisely generates multiple clock glitches to skip specific instructions of the code.

3.4 SCA Countermeasures

In this section countermeasures in the literature are described which protect against the previous attacks discussed in Section 3.2 and are incorporated into the software designs presented here. The purpose of this research is to evaluate their impact. Table 3.3 outlines the types of attack each countermeasure aims to prevent.

3.4.1 Double computation and verify-after-sign

A straightforward method of detecting any fault attack on the signing process (such as in Bruinderink and Pessl [2018]) is to compute the signature twice and confirm they are the same. The double computation method works for deterministic schemes such as Dilithium, and for non-deterministic schemes by fixing the randomness. A conservative estimate of this is doubling the signing time, which in the case of slow algorithms can be restraining. Running the verification process immediately after signing can also give the signer confidence that their hardware has not been subject to fault attacks [Bruinderink and Pessl, 2018]. However, where the fault attack targets the sampling, such as in [Espitau et al., 2016], this may still produce a valid signature and go undetected.

3.4.2 Sample Discard

Sample discard is the process whereby extra cache reads from random addresses are performed in an attempt to distort any statistics used in side-channel analysis. These extra reads are then thrown away. In this implementation, where it has been added as an extra preventative layer within the Gaussian sampling components, the design uses a range of discard rates at 6.25%, 12.5% and 25%.

3.4.3 Shuffling

The Fisher-Yates shuffling algorithm [Fisher et al., 1938] can be used on the Gaussian sampler output to effectively randomly permute vector coefficients in an unbiased manner. This algorithm takes in the sampled Gaussian coefficients and permutes, or shuffles them, to create a fresh Gaussian noise vector. This random shuffling

operation removes any timing information which an attacker can exploit. However, Pessl [Pessl, 2016b] suggested that this countermeasure is not sufficient on its own, by retrieving the BLISS secret key with 7000 signatures through a Gaussian sampler attack, despite shuffling in place. The VectorBlindSample algorithm introduced by Saarinen [Saarinen, 2017] extends this method and invokes the Fisher-Yates shuffling procedure, but constructs the target distribution as a sum of random samples. This algorithm is utilised to ensure a constant-time implementation to further increase the attack complexity. The algorithmic loops are not data dependent and the operations are such that regardless of whether a value is swapped or not they are always performed.

3.4.4 Constant-time

Gaussian samplers are particularly susceptible to timing attacks but their performance often suffers in constant-time implementations. These designs evaluated in this chapter use an efficient constant-time design of the CDT [Peikert, 2010a] sampling technique from libSAFEcrypto [SAFEcrypto, 2018] as this is much more simple to adapt for multiple parameter sets than other contenders. An exact number of look-up reads are performed each time, with the comparison being done using arithmetic logic using the same operations regardless of the branch taken according to the CDT algorithm. The upper bound for the number of look-ups required is $\lceil log_2 N \rceil$ and so each call of the sampler is padded out to the nearest power of two to take this same amount of time. For the required values of the standard deviation, this CDT sampler gives better performance than the Knuth-Yao [Knuth and Yao, 1976b], discrete Ziggurat [Marsaglia et al., 2000, Buchmann et al., 2013a], and Bernoulli [Ducas et al., 2013a] samplers when operated in constant-time.

The NTT and FFT functions, described in §2.5.3, are also made constant-time in the designs evaluated here. The modular reduction part of the NTT can leak timing information [Scott, 2017] and so this implementation is made constant-time by avoiding branch operations at the cost of some performance. This is done by processing all required branch variables and eschewing logic for more time consuming arithmetic operations. Performance is increased by employing lazy reduction to reduce the number of modular reductions that occur and using Single Instruction, Multiple Data (SIMD) vectorisation. Point-wise modulo multiplication computes each element independently, allowing a multiplication of two polynomial rings in the NTT domain. As the operations are sequential and unconditional, it is constant-time. The transform routine utilises the Cooley-Tukey NTT algorithm with some small optimisations designed to increase performance, enable automatic vectorisation, and ensure that no conditional branch operations are used. The inverse NTT operation is the mirrored operation of the forward NTT, but additionally requires shuffling of the output coefficients and the use of both forward and inverse roots of unity coefficient tables. Automatic vectorisation of the final shuffle of the inverse NTT was not possible as the loop conditions were considered too complex and, therefore, it is performed separately to the modular reduction in order to improve performance. The modular reduction design is a simplified range restriction that avoids the use of logical operations and instead opts to use arithmetic operations, in particular the constant time *less than* comparison function.

3.4.5 Zero-check

A simple, efficient method of detecting the Bearz attack is checking that the sampled vector does not go to zero at any point along its length at the end of the $ffSampler$ algorithm. This is performed using the following code:

```
for(i=0; i<n;i++){
k+=z0[i];
if (k!=0)
 {
   i=0;
   break;
 }
}
if (i==0) fprintf(stderr, "Z_0 is NOT all-zero\n");
if (i==n-1) fprintf(stderr, "Z_0 is all zero\n");
```

This is the form of an additive iterative loop which sums the $\mathbf{z_0}$ vector coefficients. Once the sum attains a non-zero value, which implies that $\mathbf{z_0}$ is not an all-zero vector, it exits and outputs as a success. If the whole $\mathbf{z_0}$ vector is zero, such as in the event of a Bearz attack, it will output a warning statement. This is a low overhead but effective countermeasure.

3.5 Results and evaluation

This section analyses the performance results of the optimised software designs of the signature schemes, as well as the countermeasures discussed in the previous

section. Clearly, as with any attack countermeasure, some performance degradation is expected, however what is more important is their *relative* impact on performance. Unless otherwise stated, performance figures were obtained on Intel E5-1620 CPU @ 3.7GHz, with hyperthreading and Turbo Boost disabled, running on a single CPU. The platforms of the reference values are also stated. Any results with "-" mean the results are unchanged, thus have not been affected by the countermeasure(s). The security levels for each scheme's parameter sets are given in Table 3.1.

The designs proposed here were used in the SAFEcrypto project discussed in §2.7.1 to demonstrate the practical application of signature scheme in IoT and satellite communication applications. Therefore the performance is not directly comparable to reference implementations. For the key generation figures, each scheme was run 6 times and an average runtime taken. For signing and verification, each scheme was run 3 times as there was less variation. The designs were subject to the validation methods described in §2.7.1.

3.5.1 BLISS-B with Countermeasures

The results shown in Table 3.4 show the performance of a BLISS-B optimised design, with the effect on the signature algorithm of adding the verify-after-sign, double computation, sample discard, Fisher-Yates shuffling, and VectorBlindSample countermeasures, and of adding a combination of these. Verify-after-sign slows performance by $\leq 10\%$ for all parameter sets, whereas sample discard slows performance by an average of 10%, 25%, and 47% for each discard rate. Shuffling only causes a 10% slow-down, and VectorBlindSample halves the throughput performance (ops/sec). An example of 3 countermeasures (verify-after-sign, low sample discard and Fisher-Yates

shuffling) combined is also provided, which results in a 24.5% reduction in performance. These varying security/runtime trade-off options give the user a choice depending on their desired performance needs. This research regards Set III as being sufficient for the next 30 years, when combined with either verify-after-sign or a shuffling and 25% sample discard countermeasure, on the basis of Moore's Law [Moore et al., 1965].

3.5.2 Dilithium with Countermeasures

Table 3.5 shows the performance results of the proposed software designs of Dilithium and the effect on the performance of the signature algorithm by adding verify-after-sign and double computation countermeasures, both independently and combined. As expected, the double computation countermeasure has a significant impact on performance and increases the signing time by double, and the signing procedure slows down with the verify-after-sign countermeasure, affecting performance by between $14 - 20\%$. On the basis of Moore's Law [Moore et al., 1965], the Set-III parameter set should be secure for nearly 80 years from *classical* computers. Therefore a user may prefer to use the strongest parameter set under the quantum threat, along with a verify-after-sign check to protect from fault attacks.

Table 3.4 Performance results (ops/sec) for BLISS-B with SCA countermeasures, on Intel E5-1620@ 3.7GHz unless otherwise stated. KeyGen figures are not given in the reference papers.

Implementation	Parameters	KeyGen	Sign	Verify	% decrease
Reference Sign: Intel Core @ 3.4GHz Verify: Intel Core i7 @ 3.4GHz [Ducas, 2014c, Ducas et al., 2013b]	Set-I	N/A	7813	33000	-
	Set-II	N/A	5405	33000	-
	Set-III	N/A	6849	32000	-
	Set-IV	N/A	5988	31000	-
Baseline +Constant-time NTT	Set-I	11746	3500	28769	-
	Set-II	11762	2558	30765	-
	Set-III	11074	3219	29470	-
	Set-IV	10798	2780	22927	-
+Verify-after-sign	Set-I	-	3194	-	8
	Set-II	-	2380	-	7
	Set-III	-	2939	-	9
	Set-IV	-	2501	-	10
+Sample Discard (Low) (6.25%)	Set-I	-	3133	-	10
	Set-II	-	2298	-	10
	Set-III	-	2900	-	10
	Set-IV	-	2504	-	10
+Sample Discard (Medium) (12.5%)	Set-I	-	2600	-	26
	Set-II	-	1905	-	26
	Set-III	-	2399	-	25
	Set-IV	-	2115	-	24
+Sample Discard (High) (25%)	Set-I	-	1825	-	48
	Set-II	-	1359	-	47
	Set-III	-	1689	-	48
	Set-IV	-	1516	-	45
+Fisher-Yates Shuffling	Set-I	-	3156	-	10
	Set-II	-	2275	-	11
	Set-III	-	2897	-	10
	Set-IV	-	2521	-	9
+VectorBlindSample	Set-I	-	1831	-	48
	Set-II	-	1375	-	46
	Set-III	-	1701	-	47
	Set-IV	-	1519	-	45
+Verify-after-sign +Sample Discard (Low) (6.25%) +Fisher-Yates Shuffling	Set-I	-	2625	-	25
	Set-II	-	1954	-	24
	Set-III	-	2429	-	25
	Set-IV	-	2100	-	24

Table 3.5 Performance results (ops/sec) for Dilithium with SCA countermeasures, on Intel E5-1620@ 3.7GHz unless otherwise stated.

Implementation	Parameters	KeyGen	Sign	Verify	% decrease
Reference [Lyubashevsky et al., 2017] AVX2, Intel Core i7-4770K @ 3.5GHz	Set-I	104128	338922	105584	-
	Set-II	156432	493332	150228	-
	Set-III	225432	673144	207164	-
	Set-IV	292404	711018	288398	-
Baseline +Constant-time NTT	Set-I	15624	2644	11022	-
	Set-II	10283	1415	7403	-
	Set-III	7183	979	5416	-
	Set-IV	5306	1061	3928	-
+Verify-after-sign	Set-I	-	2180	-	18
	Set-II	-	1228	-	13
	Set-III	-	842	-	14
	Set-IV	-	859	-	19
+Double Computation +Verify-after-sign	Set-I	-	1090	-	59
	Set-II	-	614	-	57
	Set-III	-	421	-	57
	Set-IV	-	430	-	59

3.5.3 Falcon with Countermeasures

Results shown in Table 3.6 show the performance of the optimised Falcon design and the effects of the countermeasures on the signature algorithm. Interestingly, Falcon's performance is not largely affected by the countermeasures. Applying verify-after-sign to protect against Bearz does not significantly slow performance ($\sim 5\%$), and the key generation only drops significantly with a high sample discard rate. The recommended combination of countermeasures (verify-after-sign, low sample discard and Fisher-Yates shuffling) together slow the key generation and signing down by less than 5% for Parameter Set-I. Sample discard has less than a 5% effect on key generation, and the effect of the 6.5% rate on Parameter Set-III is so minimal that the effect of fluctuation causes an increase from the rate without discard. Supposing security levels are required to be at the recommended 256-bit today, Set III would be sufficient for the next 30 years, using a 25% sample discard rate to mitigate Gaussian sampling attacks.

Table 3.6 Performance results (ops/sec) for FALCON with SCA countermeasures, including % decrease in affected components, on Intel E5-1620@ 3.7GHz unless otherwise stated.

Implementation	Parameters	KeyGen	Sign	Verify	% decrease
Reference [Prest et al., 2017] Intel Core i7-6567U @ 3.3GHz	Set-I	143	6081.9	37175.3	-
	Set-III	50.9	3072.5	17697.4	-
Baseline +constant-time NTT	Set-I	29.8	1484	29002	-
	Set-III	11.1	734	14366	-
+Verify-after-sign	Set-I	-	1412	-	5
	Set-III	-	699	-	5
Zero-check	Set-I	-	1006	-	32
	Set-III	-	489	-	16
+Double Computation +Verify after sign	Set-I	-	706	-	52
	Set-III	-	350	-	76
+Sample Discard (Low) (6.25%)	Set-I	26.3	-	-	12
	Set-III	12.0	-	-	+8
+Sample Discard (Medium) (12.5%)	Set-I	24.9	-	-	16
	Set-III	10.2	-	-	8
+Sample Discard (High) (25%)	Set-I	27.7	-	-	7
	Set-III	8.0	-	-	28
+Fisher-Yates Shuffling	Set-I	23.87	-	-	20
	Set-III	10.9	-	-	2
+BlindVector	Set-I	27.03	-	-	9
	Set-III	8.7	-	-	49
+Verify-after-sign +Sample Discard (Low) (6.25%) +Fisher-Yates Shuffling	Set-I	29.0	1426	-	3, 4
	Set-III	10.7	706	-	4, 4

3.5.4 Performance Evaluation

Comparing the performance of the two NIST candidates in Tables 3.5 and 3.6, Falcon performs better for comparable parameters (Set-II of Dilithium and Set-I of Falcon). This comparison holds true for the signing process both before and after the double computation and verify-after-sign measures are added (706 op/s compared to 614 op/s in the latter case). Figure 3.3 shows that a protected Falcon signing process is still competitive against Dilithium. In contrast, the Dilithium KeyGen is 300x faster, however this is usually an offline process and so not as paramount.

Figure 3.4a shows the performance of the KeyGen and Signing algorithms of Falcon and BLISS-B when the sample discard countermeasure is applied at the three different discard rates. The results are given as a percentage of the baseline operations per second, without any other countermeasures. The reader is reminded that Dilithium does not utilise a Gaussian sampler and so is omitted in this graph. Clearly, sample discard slows down the BLISS-B signing algorithm more drastically than the Falcon key generation algorithm. The sample discard affects Falcon approximately 20% less than BLISS-B. Figure 3.4b shows the effect of the verify-after-sign countermeasure on each of the schemes, again as a percentage of the baseline operations per second. It can be seen that Falcon's performance is barely affected by this countermeasure, only up to a 5% drop, due to its efficient verification component, whereas Dilithium's drops by almost 20% for the NIST security level 3 parameter set. BLISS-B has approximately a 10% decrease, bar the highest security parameter set. Overall, BLISS-B remains the most efficient scheme, but Dilithium has arguably less exploitable security assumptions, due to its use of less structured module lattices and uniform sampling, whilst Falcon leads in terms of compact public key and signature sizes.

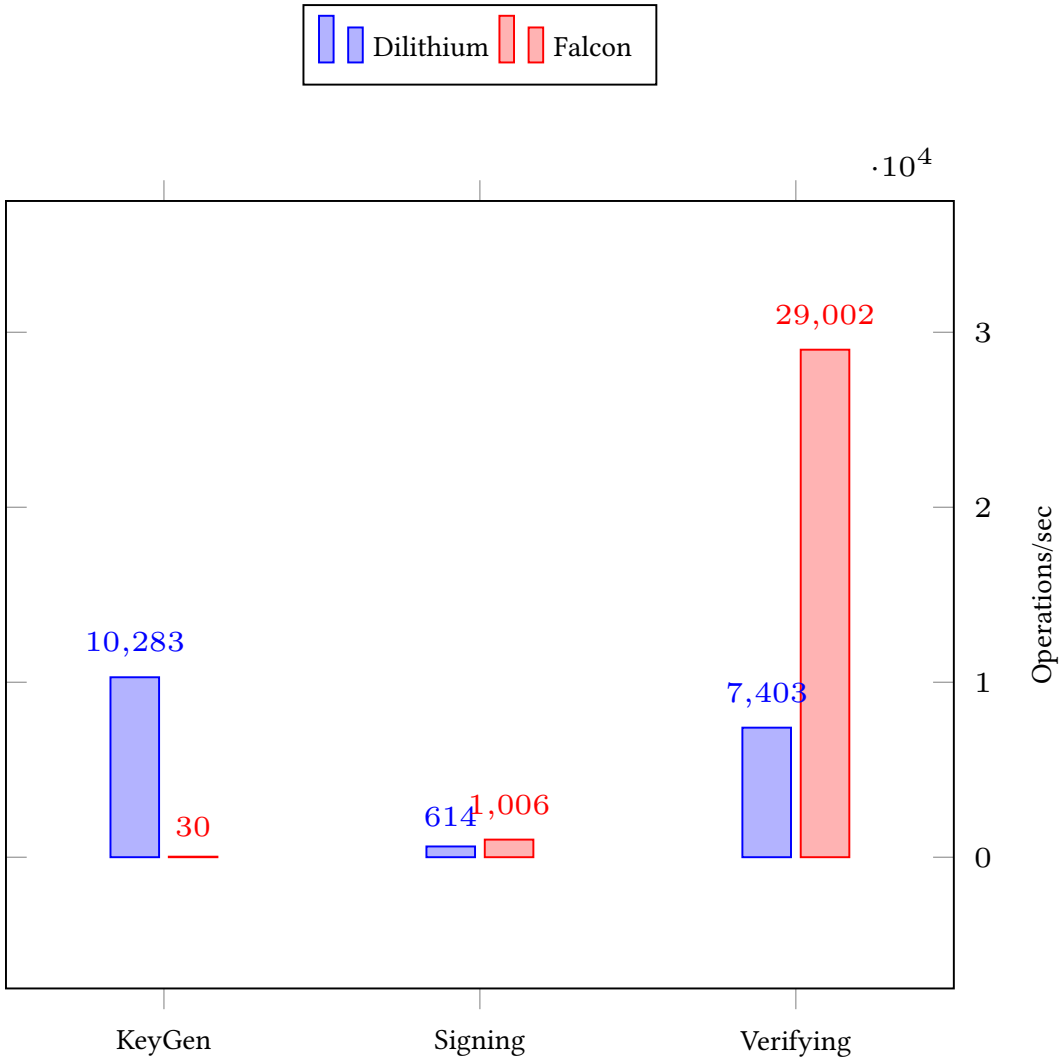
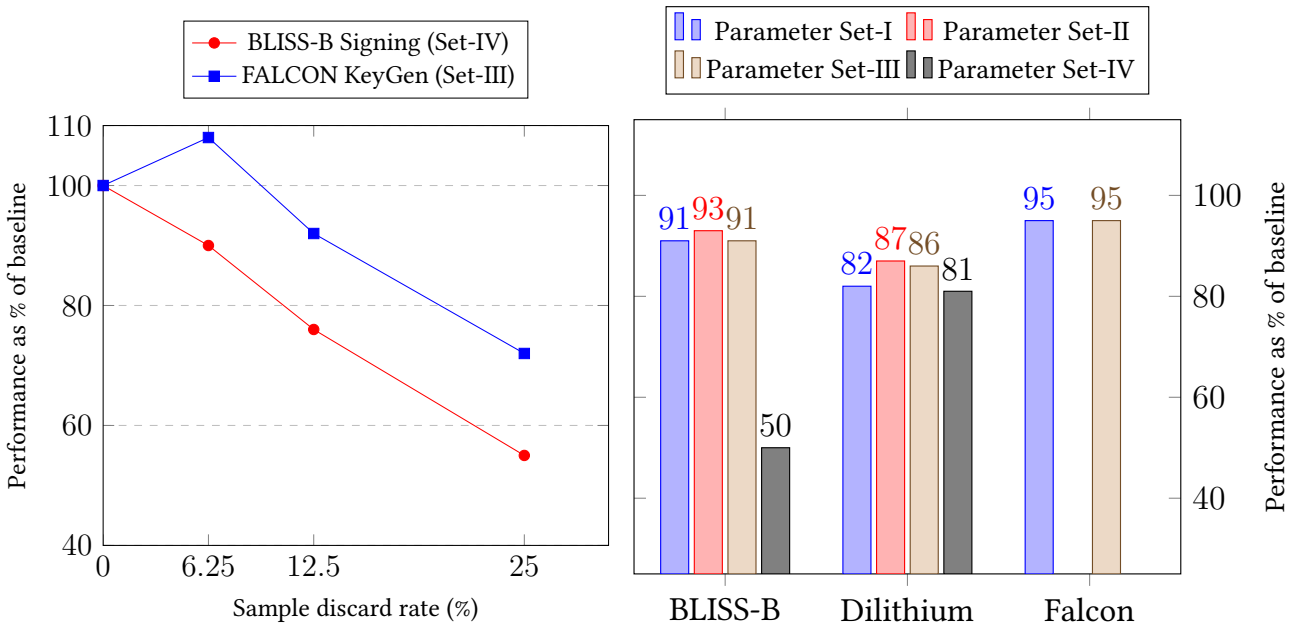


Fig. 3.3 Protected Dilithium (+Double comp. and Verify-after-sign) vs. protected Falcon (+zero-check)

3.6 Conclusions

The purpose of this research is to demonstrate the effects of countermeasures to physical attacks on the performance of three leading signature schemes. This reiterates



(a) Sample Discard countermeasure effect on BLISS-B and Falcon.

(b) Verify-after-sign effect on BLISS-B, Dilithium, and Falcon.

Fig. 3.4 Performance analysis of the proposed SCA countermeasures for the three signature schemes.

the comments of NIST with regards to considering implementation aspects whilst evaluating the candidates.

In this research, a new attack on the Falcon signature scheme has been proposed, Bearz. This has shown that Falcon is vulnerable to realistic fault attacks of its Gaussian sampler but this can be resolved by implementing appropriate countermeasures. These include the verify-after-sign countermeasure, as the faulty signature will not pass the norm bound test, and a zero-check to ensure the outputted lattice vector does not contain a string of zeros which would reveal information. Secondly, countermeasures to known physical attacks are implemented, for example double computation, a costly but reliable method, assuming the attacker only has access to the hardware during one of the signing processes. Interestingly, the performance of Falcon is not

as adversely affected as much as its NIST competitor scheme Dilithium, when SCA countermeasures are added. Also at high security levels, it does not suffer as much of a slowdown as BLISS-B. However, Falcon still remains the most inefficient of the three schemes and physical security is still new territory. Dilithium has the advantage of being immune to Gaussian sampler attacks, however protection from the aforementioned fault attacks remains a largely open problem. Its performance is slowed down by nearly 20% with countermeasures in place and has greater degradation than BLISS-B with a protected Gaussian sampler.

In overall conclusion, it is recommended to choose the optimum digital signature depending on the desired trade-off between security and performance. It is important to note that each of the schemes provide different security levels. Each of the three schemes portrayed here offer unique trade-offs and application examples. BLISS-B is the most efficient/fastest performance e.g. instantaneous communication requirements, Dilithium has the advantage of a less exploitable lattice structure and sampler e.g for long term applications such as satellites, and Falcon is the most compact (pk+sig) size e.g. suitable for constrained IoT devices. This research has shown that the effect of countermeasures has balanced the performance of the schemes, for instance Falcon is the slowest originally but does not deteriorate further with physical attack protection.

In addition to physical attack resistance being an open problem, there is limited research on advanced lattice-based cryptographic primitives. As the underlying components of Falcon have already been used in an IBE scheme [Ducas et al., 2014a], the natural next step of this research would be to investigate the range of extended schemes related to Falcon. However, upon inspection, there are no existing practical, non-protected, optimised designs. Therefore for the next chapter, this is taken as a

starting point, investigating IBE in Chapter 4 and HIBE in Chapter 5, with attacks on these extended schemes discussed as further work in Chapter 6.

CHAPTER 4

Practical Identity-Based Encryption over NTRU Lattices

4.1 Introduction

An identity-based encryption scheme enables the efficient distribution of keys in a multi-user system. Such schemes are particularly attractive in resource constrained environments where critical resources such as processing power, memory and bandwidth are severely limited. Managing keys in systems that use Public Key Cryptography (PKC) as the primary means of authentication is achieved within the established framework of a Public Key Infrastructure (PKI). However, it is widely known that a PKI requires an extensive network for operational and certificate management. This imposes significant overheads in resource constrained environments and in practice limits the use of PKC in large scale, bandwidth intensive applications requiring real-time performance.

As technology progresses to operational landscapes such as the IoT, the need for alternative PKC schemes comes to the fore. Schemes such as IBE have been introduced

to overcome the complexity issues associated with traditional PKI based approaches. IBE simplifies key generation and distribution in a multi-user system by providing a method of distributing public keys without the need for certificate management. In environments with limited resources IBE can offer the potential for PKC to be utilised when it is needed and not just when it can be accommodated.

In addition to resource efficient schemes, many applications have long-term security requirements. For such applications the threat of quantum attacks must be mitigated. Finding quantum-safe schemes analogous to classical schemes such as public key encryption and digital signature schemes is the subject of intense research. As mentioned in §2.4, one of the advantages of LBC is that it can be used to develop advanced cryptographic primitives. However, its large memory requirements and intensive computation can make it impractical for real-world devices. The aim of publication [McCarthy et al., 2017], upon which this chapter is based, is to achieve a practical lattice-based IBE scheme without compromising security. Ducas et al. [2014a] proposed the first pragmatic lattice-based IBE scheme, henceforth referred to as DLP-IBE. This research presents an optimised standalone ANSI C implementation of this scheme. There has been no prior indication of the practical performance and costs of a lattice-based IBE scheme. This work not only provides a benchmark for this type of scheme, but these efficiency improvements bring it into the realm of practicality for the post-quantum setting. The proposed designs are compared to and shown to be competitive with existing, non-quantum-secure, elliptic curve (ECC)- and quadratic residue-based IBE schemes.

Further to this, a novel IBE scheme is derived from the Falcon [Prest et al., 2017] hash-and-sign signature scheme presented in Chapter 3. This is a natural extension of

DLP-IBE as the construction is similar, but uses more efficient sampling and memory storage components.

The contributions of this chapter are:

- the first practical optimised design of the DLP-IBE scheme achieved through algorithmic and design optimisations
- revised recommended parameters for the IBE scheme to support faster multiplication methods and a wider range of security levels than previously reported
- proposal of the Falcon-IBE scheme

The chapter is organised as follows: §4.2 reviews both non-lattice and lattice-based IBE schemes and their underlying security assumptions. §4.3 introduces the DLP-IBE scheme, and describes the proposed software architecture, which includes the improved functionality of master key generation and extract algorithms as well as the encrypt and decrypt processes. The major bottlenecks and challenges are acknowledged and optimisations and methods used to address them are presented. §4.5 introduces the Falcon-IBE scheme. Conclusions are given in §4.6.

The main research of this chapter appears in the publication [McCarthy et al., 2017].

4.2 Identity-Based Encryption

Traditional PKI uses certificates to bind users to their public key. However, IBE removes the need for certificates by using a piece of meaningful information, such as an email address, MAC address or a device identifier as the user's public key. The IBE process is demonstrated in Figure 4.3 and the steps are as follows.

1. The master key generator outputs the master public/private key pair for the system
2. A trusted authority uses a master secret key to compute a user secret key that is mathematically related to the user's public key.
3. To send a message to a particular user, the sender needs the master public key and the user ID, which together form their public key, and produces the ciphertext.
4. The receiver decrypts this using their personal secret key.

The use of already established personal information such as the user ID removes the need for public key distribution. Elements such as timestamps can be incorporated into the user keys to provide a key refresh mechanism. Additionally, the use of timestamps with the user ID can allow senders to encrypt messages that can only be read in the future. In the generic IBE instantiation the central authority has complete access to keys and can therefore decrypt any message and so should be a trusted authority. Additionally, the communication channels between users and the trusted authority should be secure.

Some advantages of IBE schemes are:

- The sender does not need to go to a Certificate Authority (CA) in order to obtain a user's public key as it can be computed using the user ID and therefore obtaining keys is more efficient than traditional PKI
- Timestamps can be combined with the ID to accommodate key refresh and revocation

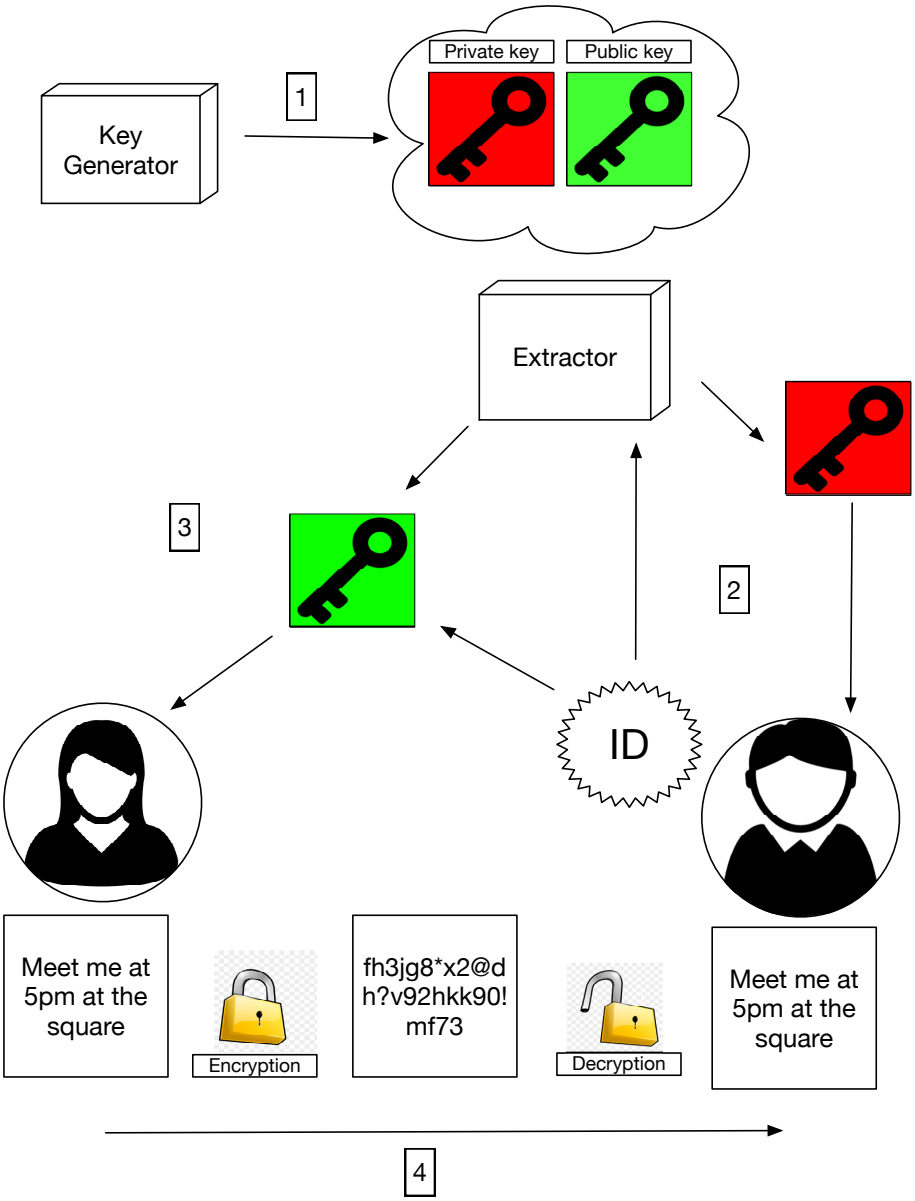


Fig. 4.1 The workings of an IBE scheme

- There is no need for pre-registration of a user onto a database
- They can be extended to further primitives such as signatures, broadcast encryption, key agreement and PKE with keyword search
- They mitigate the need for a CA and therefore the requirement for secure local storage
- Decryption capabilities can be delegated, for example to temporary devices or individual staff members within a workforce [Boneh et al.]

On the contrary, there are disadvantages:

- There is danger of key escrow as the central authority has access to all private keys and therefore any compromise would affect the whole system
- The KMS has the ability to to derive any user's private key and therefore decrypt any message
- Communication channels from the central authority are required to be secure
- Key revocation could add complexity, for instance if a key needs revoked before its expiry date
- There is a requirement to check uniqueness of user IDs

Applications of IBE:

- Voltage Security[1] and Trend-Micro[2] both have supplied commercial solutions to email and file encryption using IBE

[1]https://www.voltage.com
[2]https://www.trendmicro.com

- It is suitable for applications where key escrow is not an issue, to provide an efficient solution for closed environments such as sensor networks [Szczechowiak and Collier, 2009]

4.2.1 Current IBE schemes

The first notion of IBE was presented by Shamir [1984]. The idea was to eliminate the need for a public certificate across email systems. These schemes allowed secure communication without exchanging user keys. Shamir presented a solution for an identity-based signature scheme but it was not until 2001 that such an encryption scheme was realised [Boneh and Franklin, 2001a]. There are two main methods for constructing traditional IBE schemes; using pairings and therefore depending on the hardness of problems over bilinear mappings between groups, as in Boneh and Franklin [2001a], or quadratic residues as in Cocks [2001a]. Neither of these types of cryptography are quantum-safe. Boneh co-founded a start-up company in 2002 called Voltage Security Inc.[3] which currently provides IBE solutions to industry. These include secure email and file transfer applications. The TErrestrial Trunked RAdio (TETRA) communications network [ETS, 2005], used for UK emergency services, is currently transitioning to the PS-LTE network. The standard for the security of mission critical push to talk (MCPTT) over LTE [3GP, 2016] currently uses the ECC-based IBE scheme Mikey Sakke [Groves, 2012] based on a scheme by Sakai and Kasahara [2003]. IBE is well-suited to this application due to its instantaneous communication capabilities. However, these schemes are susceptible to quantum attacks due to Shor's algorithm, creating the need for quantum-resilient variants. The importance and

[3]https://www.voltage.com

suitability of IBE to this use-case has led to further collaborative work with NCSC in Chapter 5.

4.2.2 Lattice-based IBE schemes

The first application of lattices to IBE schemes was in 2008 by Gentry et al. [2007]. The main contribution of their work was a sampling algorithm (referred to as the GPV sampler) which demonstrated how to use a short basis as a trapdoor for generating short lattice vectors. This sampler was used to construct a lattice-based IBE scheme that resembled Cocks' traditional scheme (due to the use of a trapdoor), and can be considered as the dual of Regev's LWE scheme [Regev, 2009]. However, the security proof was in the ROM and the user secret keys were large - sizes of $O(n^2)$ bits, translating to 27000 bits at 192-bit security level, and a 30000 bits ciphertext. Agrawal et al. [2010b] proposed a LWE-based IBE scheme with a trapdoor structure, with performance comparable to the GPV scheme. It uses a sampling algorithm to obtain a basis with low GS-norm, as described in §2.5.4, for the master secret key and forms a lattice family with two associated trapdoors to generate short vectors; one for all lattices in the family and the other for all but one. It improves on previous schemes [Agrawal and Boyen, 2009, Cash et al., 2010] which process the user identities bit by bit by instead considering them as a whole. The public key is $O(nm)$, where the lattice basis is of size n x m. In 2016, Apon et al. [Apon et al., 2016] proposed the most efficient *standard* LWE IBE scheme to date with a public key size of $O(2nm\log(q))$. This includes the design of a new encoding scheme for identities, incorporating a collision-resistant hash function. The first ring-LWE based IBE scheme was the DLP-IBE scheme by Ducas et al. [2014a]. The use of the ring variant increases efficiency by reducing

the public key to a polynomial/vector with $O(n)$ complexity and ciphertext $O(2n)$. However, it makes additional assumptions to standard LWE which may introduce security vulnerabilities. In particular, it uses the GPV sampling algorithm on a certain distribution of NTRU lattices to increase its efficiency. Other ring-LWE schemes have since been proposed, for example Katsumata and Yamada [2016] introduced a scheme based on Yamada's 2016 standard-LWE scheme [Yamada, 2016], and exploit the ring properties and assume the ring-LWE problem hardness for fixed polynomial approximation factors. The public parameters in this scheme are of size $O(nl^{1/d}\log(n))$, ciphertext $O(n\log(n))$ and private key $O(n\log(n))$. Since McCarthy et al. [2017], the only other published lattice-based IBE scheme showing practicality has been by Bert et al. [2018] which is based on the ring-LWE/SIS problems and combines the construction of Agrawal et al. [2010a] with the sampler of Micciancio and Peikert [2012]. However, the storage requirements for this scheme are large (77kb for the user private key) and so the DLP-IBE scheme is still considered the most efficient scheme to date due to its smaller key sizes.

4.3 The DLP-IBE Scheme

The DLP-IBE scheme was introduced in Ducas et al. [2014a] as the first efficient lattice-based IBE scheme, whereby the underlying computational hardness is based on the NTRU and the ring-LWE assumption. Operations are over the polynomial ring $\mathcal{R}_q = \mathbb{Z}_q[x] \big/ \langle x^N + 1 \rangle$, a variant of the NTRU ring. The first practical effort towards evaluating the performance of the DLP-IBE scheme was a PoC implementation by the original authors [Prest, 2016]. Their implementation is written in C++ and depends on

the NTL library[4]; it also relaxes some constraints such as security thresholds for the lattice basis size. Recently Güneysu and Oder [2017] demonstrated the efficiency of the encrypt and decrypt components of this scheme on a range of low-cost microcontrollers and reconfigurable hardware. The research in this chapter investigates and optimised design of the scheme in the C programming language. In particular the focus is on the computationally intensive tasks of the key generation and extraction algorithms. The design includes the enforcement of the aforementioned security thresholds and in particular provides the first indication of practical performance of the overall scheme.

4.3.1 Description of the DLP-IBE Scheme

An IBE scheme consists of 4 main components. Table 4.1 shows the inputs and outputs of each algorithm in the DLP-IBE scheme. The DLP-IBE scheme also uses an additional algorithm for Gaussian sampling over the lattice.

Table 4.1 Algorithm summary

Algorithm	**Inputs**	**Outputs**
Master KeyGen	N, q	$\mathbf{B} \in \mathbb{Z}_q^{2N \times 2N}$, $\mathbf{h} \in \mathcal{R}_q$
Extract	$\mathbf{B} \in \mathbb{Z}_q^{2N \times 2N}$, $H : \{0,1\}^* \rightarrow \mathbb{Z}_q^N$, id	$\mathbf{SK}_{\text{id}} \in \mathcal{R}_q$
Encryption	$\mathbf{h} \in \mathcal{R}_q$, id, $m \in \{0,1\}^m$, $H : \{0,1\}^* \rightarrow \mathbb{Z}_q^N$, $H' : \{0,1\}^N \rightarrow \{0,1\}^m$.	$(\mathbf{u}, \mathbf{v}, \mathbf{c}) \in \mathcal{R}_q^2$
Decryption	$\mathbf{SK}_{id}$, $(\mathbf{u}, \mathbf{v}, \mathbf{c}) \in \mathcal{R}_q^2$	$m \in \{0,1\}^N$

[4]https://www.shoup.net/ntl/

The Key Generation algorithm

This algorithm generates the master secret key and the master public key. This happens once per environment setup. In DLP-IBE, the private key is an NTRU lattice basis and the public key is an identifier of that lattice, in the form of a polynomial. The scheme works over the polynomial ring modulo (x^N+1) of integers modulo q, denoted $\mathcal{R}_q = \mathbb{Z}_q[x]/\langle x^N+1\rangle$.

In this scheme, the inputs of the key generation algorithm (Algorithm 12) are the degree N of polynomials $\mathbf{f}, \mathbf{g}$ and $\mathbf{h}$ and the modulus q and the outputs are the master secret key $\mathbf{B} \in \mathbb{Z}_q^{2N\times 2N}$ and master public key $\mathbf{h} \in \mathcal{R}_q$. The inituition behind polynomials F and G is described in §2.4.2.

Algorithm 12: Key Generation [Ducas et al., 2014a]

Data: N, q
Result: $\mathbf{B} \in \mathbb{Z}_q^{2N\times 2N}$, $\mathbf{h} \in \mathcal{R}_q$

1 $\sigma_{\mathbf{f}} = 1.17\sqrt{\frac{q}{2N}}$
2 $\mathbf{f}, \mathbf{g}, \leftarrow \mathcal{D}_{N,\sigma_{\mathbf{f}}}$
3 Norm $\leftarrow max\left(||\mathbf{g}, -\mathbf{f}||, \left|\left|\left(\frac{q\bar{\mathbf{f}}}{\mathbf{f}*\bar{\mathbf{f}}+\mathbf{g}*\bar{\mathbf{g}}}, \frac{q\bar{\mathbf{g}}}{\mathbf{f}*\bar{\mathbf{f}}+\mathbf{g}*\bar{\mathbf{g}}}\right)\right|\right|\right)$
4 **if** *Norm* $> 1.17\sqrt{q}$ **then** go to Step 2;
5 Compute $\rho_{\mathbf{f}}, \rho_{\mathbf{g}} \in \mathcal{R}$ and $R_{\mathbf{f}}, R_{\mathbf{g}} \in \mathbb{Z}$ such that:
$-\rho_{\mathbf{f}} \cdot \mathbf{f} = R_{\mathbf{f}}$ and $-\rho_{\mathbf{g}} \cdot \mathbf{g} = R_{\mathbf{g}}$
6 **if** $GCD(R_{\mathbf{f}}, R_{\mathbf{g}}) \neq 1$ *or* $GCD(R_{\mathbf{f}}, q) \neq 1$ **then** go to Step 2;
7 Compute $u, v \in \mathbb{Z}$ such that: $u \cdot R_{\mathbf{f}} + v \cdot R_{\mathbf{g}} = 1$
8 $F \leftarrow qv\rho_{\mathbf{g}}$ and $G \leftarrow -qu\rho_{\mathbf{f}}$
9 $k = \left\lfloor \frac{F*\bar{\mathbf{f}}+G*\bar{\mathbf{g}}}{\mathbf{f}*\bar{\mathbf{f}}+\mathbf{g}*\bar{\mathbf{g}}} \right\rceil \in \mathcal{R}$
10 $F \leftarrow F - \lfloor k \rceil * \mathbf{f}$ and $G \leftarrow G - \lfloor k \rceil * \mathbf{g}$
11 **return** $\mathbf{h} = \mathbf{g} * \mathbf{f}^{-1} \mod q$
12 and $\mathbf{B} = \begin{pmatrix} \mathcal{A}(\mathbf{g}) & -\mathcal{A}(\mathbf{f}) \\ \mathcal{A}(G) & -\mathcal{A}(F) \end{pmatrix}$

In Step 1, the standard deviation of the Gaussian distribution from which $\mathbf{f}, \mathbf{g}$ are generated is set to $\sigma_{\mathbf{f}} = 1.17\sqrt{\frac{q}{2N}}$ and is chosen so that the expected value of the

norm of $\mathbf{b}_1$ (and therefore $||\tilde{B}||$) is $E[||\mathbf{b}_1||] = 1.17\sqrt{q}$. In Step 3, $||\tilde{B}_{\mathbf{f},\mathbf{g}}||$, the GS-norm (see §2.5.4) of $B_{\mathbf{f},\mathbf{g}}$, is computed, where $B_{\mathbf{f},\mathbf{g}}$ is a basis of the NTRU lattice associated to $\mathbf{f}$ and $\mathbf{g}$ (and so $\mathbf{h} = \mathbf{g} * \mathbf{f}^{-1} \mod q$). If $||\tilde{B}_{\mathbf{f},\mathbf{g}}|| > 1.17\sqrt{q}$, the algorithm returns to Step 2 as the GS-norm needs to be small enough so the basis can form a short trapdoor for sampling elements in the user key extraction component. In Step 5 the Extended Euclidean algorithm is used to compute $\rho_{\mathbf{f}}, \rho_{\mathbf{g}} \in \mathcal{R}$ and $R_{\mathbf{f}}, R_{\mathbf{g}} \in \mathbb{Z}$ such that $\rho_{\mathbf{f}} \cdot \mathbf{f} = R_{\mathbf{f}}$ and $\rho_{\mathbf{g}} \cdot \mathbf{g} = R_{\mathbf{g}}$. If $GCD(R_{\mathbf{f}}, R_{\mathbf{g}}) \neq 1$ or $GCD(R_{\mathbf{f}}, q) \neq 1$, the algorithm returns to Step 2. Next, the algorithm computes $u, v \in \mathbb{Z}$ such that: $u \cdot R_{\mathbf{f}} + v \cdot R_{\mathbf{g}} = 1$. These integers are obtained from the Extended Euclidean algorithm (this extended version keeps track of the coefficients). In step 8, $F = qv\rho_{\mathbf{g}}$ and $Q = -qu\rho_{\mathbf{f}}$ are computed so that the condition $\mathbf{f} * G - \mathbf{g} * F = q$, needed to find a short basis, is satisfied. Next $k = \left\lfloor \frac{F*\bar{\mathbf{f}}+G*\bar{\mathbf{g}}}{\mathbf{f}*\bar{\mathbf{f}}+\mathbf{g}*\bar{\mathbf{g}}} \right\rceil$ is computed and F and G are reduced: $F = F - k*\mathbf{f}$ and $G = G - k * \mathbf{g}$. The final steps generate and output the keys. Polynomial $\mathbf{h} = \mathbf{g} * \mathbf{f}^{-1} \mod q$ is the master public key and defines a lattice $\Lambda_{\mathbf{h},q}$. Matrix $\mathbf{B} = \begin{pmatrix} \mathcal{A}(\mathbf{g}) & -\mathcal{A}(\mathbf{f}) \\ \mathcal{A}(G) & -\mathcal{A}(F) \end{pmatrix}$ is the master secret key and is a short basis for $\Lambda_{\mathbf{h},q}$. $\mathcal{A}$ is an anti-circulant matrix as defined previously in §2.4.2.

Gram-Schmidt Norm

The obvious way to compute the Gram-Schmidt Norm, introduced in §2.5.4, would be to compute the norms of each of the vectors and take the maximum. However, in the case of the NTRU lattices, it was proved in §3.2 of [Ducas et al., 2014a] that there are only two candidate vectors with the largest norm, namely $\tilde{\mathbf{b}}_1$ and $\tilde{\mathbf{b}}_{N+1}$ (with the vectors ordered as in the definition of $\mathbf{B}$). Further to this, it can be proved that

$$||\tilde{\mathbf{b}}_{N+1}|| = \left|\left|\left(\frac{q\bar{\mathbf{f}}}{\mathbf{f} * \bar{\mathbf{f}} + \mathbf{g} *}, \frac{q\bar{\mathbf{g}}}{\mathbf{f} * \bar{\mathbf{f}} + \mathbf{g} * \bar{\mathbf{g}}}\right)\right|\right| \tag{4.1}$$

It is always the case that $\tilde{\mathbf{b}}_1 = \mathbf{b}_1$, therefore $||\tilde{\mathbf{b}}_1|| = ||\mathbf{b}_1||$. In this NTRU lattice basis, $\mathbf{b}_1 =$ the top row of $(\mathcal{A}(\mathbf{g}), -\mathcal{A}(\mathbf{f})) = (\mathbf{g}, -\mathbf{f})$, therefore the GS-norm can be computed solely from $\mathbf{f}$ and $\mathbf{g}$ as

$$||\tilde{B}_{\mathbf{f},\mathbf{g}}|| = \max_{i \in I} \left\{ ||\mathbf{b}_1||, \left|\left|\left(\frac{q\bar{\mathbf{f}}}{\mathbf{f} * \bar{\mathbf{f}} + \mathbf{g} * \bar{\mathbf{g}}}, \frac{q\bar{\mathbf{g}}}{\mathbf{f} * \bar{\mathbf{f}} + \mathbf{g} * \bar{\mathbf{g}}}\right)\right|\right| \right\} \tag{4.2}$$

The Extended Euclidean Algorithm

The Extended Euclidean Algorithm [Saunderson, 1740, Naranjo et al., 2010] computes the GCD of two numbers (or polynomials) $\mathbf{x}, \mathbf{y}$, and the corresponding Bezout coefficients u, v, such that $u\mathbf{x} + v\mathbf{y} = GCD(\mathbf{x}, \mathbf{y})$. Note the “extended" version refers to the algorithm in which the Bezout coefficients are stored after computation. The key generation algorithm requires the GCD computation of both integers and polynomials and this is a bottleneck due to the requirement of polynomial division. The Extended Euclidean algorithm for polynomials is along the same principles as the regular version for integers. The differences are that the variables are polynomials, the input polynomials are divided by their leading coefficients in order to become monic for use in the algorithm, and (the more intensive) polynomial multiplication and division is used.

Step 5 of Algorithm 12 states that given $\mathbf{f}, \mathbf{g} \in \mathcal{R}$, find $\rho_{\mathbf{f}}, \rho_{\mathbf{g}} \in \mathcal{R}$ and $R_{\mathbf{f}}, R_{\mathbf{g}} \in \mathbb{Z}$ which satisfy Equations 4.3 and 4.4.

$$-\rho_{\mathbf{f}} \cdot \mathbf{f} = R_{\mathbf{f}} \mod (x^N + 1) \tag{4.3}$$

$$-\rho_{\mathbf{g}} \cdot \mathbf{g} = R_{\mathbf{g}} \mod (x^N + 1) \tag{4.4}$$

The Extended Euclidean Algorithm for polynomials is used twice here. The first time, it is used to find the GCD of $\mathbf{f}$ and $x^N + 1$. The second time, it is used to find the GCD of $\mathbf{g}$ and $x^N + 1$. During the computation, the algorithm holds the coefficients while calculating the GCD of two integers or polynomials for use in the following step.

Extract

The extract algorithm generates the user secret key for a given user ID. This algorithm is run once per user. In this scheme, the extract algorithm requires the master secret key $\mathbf{B} \in \mathbb{Z}_q^{2N \times 2N}$, a public hash function $H : \{0,1\}^* \to \mathbb{Z}_q^N$ (as introduced in §2.5.1) and user identity id. The user identity can be any type of data. The extract algorithm outputs the user secret key $\mathbf{SK}_{\text{id}} \in \mathcal{R}_q$.

Algorithm 13: Extract [Ducas et al., 2014a]

Data: $\mathbf{B} \in \mathbb{Z}_q^{2N \times 2N}$, $H : \{0,1\}^* \to \mathbb{Z}_q^N$, id
Result: $\mathbf{SK}_{id}$

```
if SK_id is in local storage then
    return Output SK_id to user id
else
    t ← H(id) ;
    (s_1, s_2) ← (t, 0) − GPVSampler(B, σ, (t, 0));
    SK_id ← s_2;
    return Output SK_id to user id and keep in local storage
```

The first steps check if $\mathbf{SK}_{\text{id}}$ is in local storage; if so then the secret key has already been extracted for this user so the extractor must use this one. Extracting multiple secret keys for the same user would compromise the security of the system by leaking information. If an existing key is not found, the extraction process begins. Extraction begins by hashing the user id to arrive at an integer vector $\mathbf{t}$ of length N. The vector $\mathbf{t}$ is then concatenated with a zero vector of length N to obtain a $2N$-length vector which will become the centre of the gaussian sample over the lattice. This step uses the GPV sampler (§2.5.5) to sample a vector from the lattice defined by $\mathbf{B}$. This vector is then subtracted from the $(\mathbf{t}, \mathbf{0})$ vector to obtain $(\mathbf{s_1}, \mathbf{s_2})$ such that $\mathbf{s_1} + \mathbf{s_2} * \mathbf{h} = \mathbf{t}$. This equality is due to the use of the GPV algorithm with the NTRU lattice. The user secret key $\mathbf{SK}_{\text{id}}$ is set to $\mathbf{s_2}$ and this is output and kept in local storage.

GPV Sampler

The Gaussian Sampling algorithm presented in the original DLP paper is a variant of the GPV algorithm [Gentry et al., 2007] discussed in §2.5.5. Originally, it was deemed impractical but this distribution of NTRU lattices, along with reducing the standard deviation by a factor of $\sqrt{2}$ due to consideration of Kullbeck-Leibler Divergence (see §4 of Ducas et al. [2014a]), improves the runtime. The GPV sampler returns a short lattice vector without revealing the trapdoor. The GPV samples from a Gaussian distribution with standard deviation that is essentially the length of the GS-norm. Aside from this, there are no other characteristics of the basis used which could leak information. A better quality (shorter) basis therefore means a narrower Gaussian distribution and the samples are closer to the centre, $\mathbf{c}$. The algorithm is a randomised variant of Babai's Nearest Plane [Babai, 1986b] for solving (or approaching the solution of) the CVP. Babai's algorithm inductively finds a lattice vector $\mathbf{v}$ close to some vector

$\mathbf{w}$. To do this, it solves the problem in a lower dimension; specifically, the sub-lattice (or plane) spanned by the first $\mathbf{b}_{2N-1}$ basis vectors. This is computed for dimension $2N$ and iterated until dimension 1, when the next “plane" is a vector. The output vectors are summed to “reverse-project" back onto the original lattice. The difference between Babai’s algorithm and the GPV sampler is movement to the next plane. Whilst Babai’s moves to the nearest plane in each iteration, the GPV sampler chooses the next plane with probability determined by distance to the centre point. The 1-dimensional Gaussian Sampler is used for this random plane selection process. This is illustrated in Figure 4.2 for a target vector $\mathbf{t}$, its nearest plane, and a Gaussian distribution over the sampled plane.

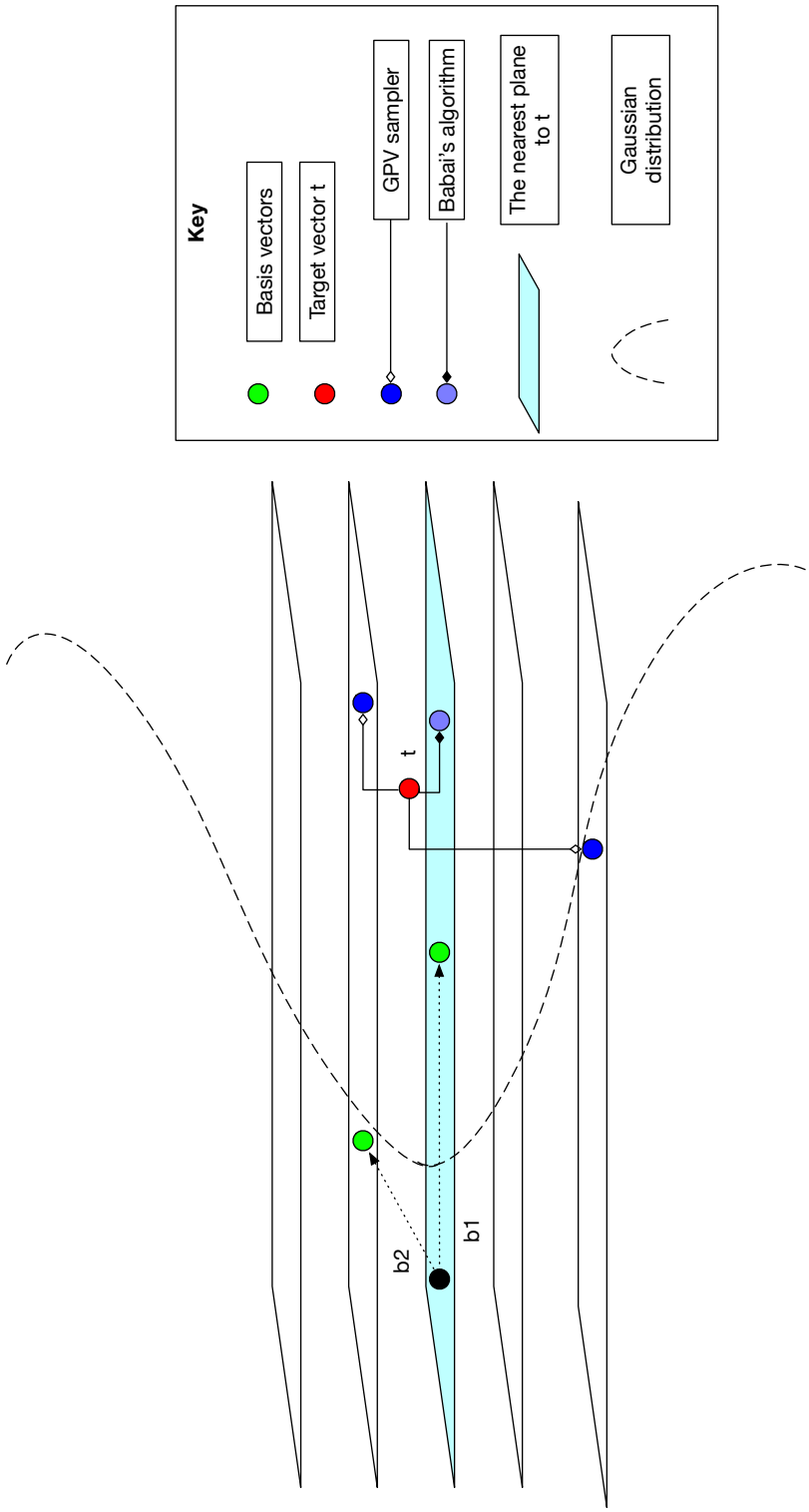


Fig. 4.2 Babai's algorithm vs. the GPV sampler

The GPV Sampler requires the basis of a $2N$-dimensional lattice (the master secret key) $\mathbf{B} \in \mathbb{Z}_q^{2N \times 2N}$ with standard deviation $\sigma > 0$ and sample centre $\mathbf{c} \in \mathbb{Z}^N$. These input parameters mean the probability of a vector $\mathbf{v}$ being sampled is proportional to $exp(-\pi||\mathbf{v} - \mathbf{c}||^2/2\sigma^2)$. The GPV Sampler algorithm outputs a sampled vector $\mathbf{v}$ in the Gaussian distribution $\mathcal{D}_{\lambda,\sigma,\mathbf{c}}$ over the lattice.

Algorithm 14: GPV Sampler [Ducas et al., 2014a]

Data: $\mathbf{B},\ \sigma > 0,\ \mathbf{c} \in \mathbb{Z}^n$
Result: $\mathbf{v} \in \mathcal{D}_{\lambda,\sigma,\mathbf{c}}$

1 $\mathbf{v}_{2N} \leftarrow 0$
2 $\mathbf{c}_{2N} \leftarrow c$
3 **for** $i \leftarrow 2N, ..., 1$ **do**
4 $\quad c'_i \leftarrow \langle \mathbf{c}_i, \tilde{\mathbf{b}_i} \rangle / ||\tilde{\mathbf{b}_i}||^2$
5 $\quad \sigma'_i \leftarrow \sigma / ||\tilde{\mathbf{b}_i}||$
6 $\quad z_i \leftarrow Gaussian(\sigma'_i, c'_i)$
7 $\quad \mathbf{c}_{i-1} \leftarrow \mathbf{c}_i - z_i \mathbf{b}_i$ and $\mathbf{v}_{i-1} \leftarrow \mathbf{v}_i + z_i \mathbf{b}_i$
8 **return** $\mathbf{v}_0$

The algorithm is presented as Algorithm 14 and is as follows; it iterates through vectors $\mathbf{v}_{2N} \rightarrow \mathbf{v}_0$ and $\mathbf{c}_{2N} \rightarrow \mathbf{c}_0$. The vector $\mathbf{v}_0$ is the sample vector output at the end. To begin, $\mathbf{v}_{2N}$ is set to the zero vector and $\mathbf{c}_{2N}$ is set to the centre vector $\mathbf{c}$. The algorithm then iterates through i from $i = 2N \rightarrow 1$. The projection coefficient of $\mathbf{c}_i$ on the lower-dimensional plane is computed as $c'_i = \langle \mathbf{c}_i, \tilde{\mathbf{b}_i} \rangle / ||\tilde{\mathbf{b}_i}||^2$, where $\tilde{\mathbf{b}_i}$ is the Gram-Schmidt orthogonalisation of the basis vector $\mathbf{b}_i$. In Step 5, the standard deviation is scaled down to $\sigma'_i = \sigma / ||\tilde{\mathbf{b}_i}||$. Step 6 calls the Gaussian sampler over the uniform distribution of integers for each i to obtain $z_i = Gaussian(\sigma'_i, c'_i)$. The next steps involve the projection of the centre and sample vector onto the next plane. The vector $\mathbf{v}_{i-1}$ is the closest vector to the centre in that (randomly sampled) plane:

$\mathbf{c}_{i-1} = \mathbf{c}_i - z_i\mathbf{b}_i$ and $\mathbf{v}_{i-1} = \mathbf{v}_i + z_i\mathbf{b}_i$. Finally, at the finish of all the loops, the vector $\mathbf{v}_0$ is output.

Encryption and Decryption

The DLP-IBE uses the generic R-LWE encryption scheme proposed by Lyubashevsky et al. [2013]. The encryption process uniformly samples small error polynomials to encapsulate a uniformly-sampled key, the hash of which is used to one-time-pad the message (see §2.5.1). To decrypt, the key is recovered by rounding, and this allows the message to be output.

Algorithm 15: Encrypt [Ducas et al., 2014a]

Data: $\mathbf{h}, id, m, H, H'$.
Result: $(\mathbf{u}, \mathbf{v}, \mathbf{c})$

1 $\mathbf{r}, \mathbf{e_1}, \mathbf{e_2} \leftarrow \{-1, 0, 1\}^N$, $\mathbf{k} \leftarrow \{0, 1\}^N$;
2 $\mathbf{t} \leftarrow H(id)$;
3 $\mathbf{u} \leftarrow \mathbf{r} * \mathbf{h} + \mathbf{e_1} \in \mathcal{R}_q$;
4 $\mathbf{v} \leftarrow \mathbf{r} * \mathbf{t} + \mathbf{e_2} + \lfloor q/2 \rfloor \cdot \mathbf{k}$;
5 Drop the least significant bits of $\mathbf{v}$: $\mathbf{v} \leftarrow 2^l \lfloor \mathbf{v}/2^l \rfloor$;
6 **return** $(\mathbf{u}, \mathbf{v}, m \oplus H'(\mathbf{k}))$

The encryption and decryption algorithms of the DLP-IBE scheme are given as Algorithms 15 and 16 respectively. These are based on the original R-LWE cryptosystem [Lyubashevsky et al., 2013] and are consequently well studied.

Algorithm 16: Decrypt [Ducas et al., 2014a]

Data: $\mathbf{SK}_{id}$, $(\mathbf{u}, \mathbf{v}, \mathbf{c})$
Result: $m \in \{0, 1\}^m$

1 $\mathbf{w} \leftarrow \mathbf{v} - \mathbf{u} * \mathbf{s_2}$;
2 $\mathbf{k} \leftarrow \lfloor \frac{\mathbf{w}}{q/2} \rfloor$;
3 **return** $m \leftarrow \mathbf{c} \oplus H'(\mathbf{k})$

4.3.2 Optimised Software Design of the DLP-IBE Scheme

The DLP-IBE scheme is implemented in portable ANSI C as part of libSAFEcrypto [SAFEcrypto, 2018], as discussed in §2.7.1. The proposed DLP-IBE architecture considers a range of security levels to suit deployment needs, as in practice it is only possible to deploy particular levels of security on constrained devices, depending upon their capabilities, memory resources and the information being protected. For example, a battery-powered temperature sensor offering a 192-bit security strength would certainly be secure, but the reduced battery life and increased cost are unlikely to appeal to consumers. More powerful devices within a typical IoT architecture may be able to support the full functionality of DLP-IBE, but peripheral devices may only support the much more efficient encrypt and/or decrypt functionality. In a standard scenario the peripheral devices could be provided with a user secret key for decryption purposes by one of three means: (a) embedded in firmware, (b) provided during device installation or (c) issued periodically by the DLP-IBE trusted authority. In such a scenario it is possible for DLP-IBE to provide public-key encryption for constrained devices.

Key Generation Design

Key generation is the most intensive component of the DLP-IBE scheme due to the arithmetic involving multiple-precision polynomials. The main operations are generating the basis and multiple-precision arithmetic, using NTT for faster polynomial multiplication and entropy coding for compressing the private key. A CDT Sampler is used in Step 2 to generate polynomials $\mathbf{f}, \mathbf{g}$ from a discrete Gaussian distribution

$\mathcal{D}_{N,\sigma_{\mathbf{f}}}$ over $\mathcal{R}_q$, as this achieved the best performance in constant time, in comparison to the other samplers listed in §2.8.

Throughout the entire design, floating-point Barrett reduction is used with a precomputed inverse of q, i.e. multiply by the inverse of q, truncate towards zero and then multiply by q, subtracting the result from the input to obtain the remainder, thereby replacing divisions by multiplications. In the previous design, the modular reduction function was provided by the NTL library, which uses preconditioning to select the most suitable reduction method.

The GS-norm computation is the first main bottleneck in the key generation algorithm. The b_1 norm calculation is simply placing two vectors in tandem and computing the dot product. However, the $\tilde{\mathbf{b}}_{N+1}$ norm computation is more intensive as it involves polynomial multiplication over a ring, with polynomial coefficients being of approximately 2000-4000 bits in length. Therefore, the NTT is used to transform the polynomials into the polynomial ring of integers modulo p, meaning multiplication can be done coefficient-wise by reducing it to a negative wrapped convolution, rather than the more complex classical school-book method as discussed in §2.5.3.

As the PoC was for reference purposes, it does not fully implement the f and g selection, which means that it generates keys that do not meet the security criteria and will not perform many retries for key selection if the bound thresholds are not met. The design proposed here enforces this security threshold.

Obtaining $\mathbf{B}$ and $\mathbf{B}_{GS}$ is relatively straightforward and fast. Depending upon the size of the modulus q it is possible to store B using 16-bit or 32-bit types, rather than a double, to both reduce memory storage and improve speed. It will take a shorter time to allocate, initialise and copy the data in float format. Overall this leads to reduced memory bandwidth and fewer cache misses. To reduce memory usage, this design

uses 32-bit floats to store both $\mathbf{B}$ and $\mathbf{B}_{GS}$ and thus for $N = 512$, 4MB is required for each matrix.

To implement the Extended Euclidean algorithm, the PoC reference implementation [Prest, 2016] uses the XGCD function from NTL [Shroup, 1996], which selects the most suitable strategy considering polynomial properties such as coefficient bit-length and degree. In this design, Brown's Modular GCD [Brown, 1971] is used for computing GCDs of multi-precision polynomials, which consists of two subroutines. This algorithm was chosen for its efficiency and removes the need for preconditioning. It does not require a verification division of the polynomials by the resultant GCD, which leads to good performance.

First, it maps the polynomials into the bivariate polynomial ring of integers modulo q, $\mathbb{Z}_q[x, y]$, and then uses the Chinese Remainder Theorem (CRT) to compute the coefficients. A further subroutine is required to map these polynomials from $\mathbb{Z}_q[x, y] \approx \mathbb{Z}_q[x][y]$ into $\mathbb{Z}_q[x]$, perform the GCD computation and recover the y terms using CRT. The GCD computations within the subroutines are computed using Euclid's algorithm or a variant called Half-GCD, which recursively runs half-way through the Euclidean algorithm and uses the intermediate polynomials to reduce the original ones. Equation 4.5 illustrates the mathematical problem in the familiar format of $u\mathbf{x} + v\mathbf{y} = GCD(\mathbf{x}, \mathbf{y})$, where u and v are the Bezout coefficients.

$$-\rho_{\mathbf{f}} \cdot \mathbf{f} + \triangle \cdot (x^N + 1) = R_{\mathbf{f}} \tag{4.5}$$

The $\triangle$ represents one of the Bezout coefficients computed during the Extended Euclidean Algorithm but it becomes obsolete as it is reduced $mod(x^N + 1)$ on each side to obtain Equation 4.6.

$$-\rho_{\mathbf{f}} \cdot \mathbf{f} = R_{\mathbf{f}} \mod (x^N + 1) \tag{4.6}$$

The useful outputs here are the other Bezout coefficient $\rho_{\mathbf{f}}$ and the greatest common divisor $R_{\mathbf{f}}$. The second use of the algorithm is similar but for $\mathbf{g}$ instead of $\mathbf{f}$ to obtain $\rho_{\mathbf{g}}$ and $R_{\mathbf{g}}$. The GNU Multiple Precision Arithmetic Library (GMP) is used to provide the multiple-precision arithmetic within the Extended GCD, multiplication and division components.

Extract Design

The software components of the extract operation described in Algorithm 13 are expanding the basis into its matrix form, the use of the hash function and CSPRNG, the samplers over integers and the lattice, its required GSO and entropy coding. The process as a whole is computationally simple but requires large memory resources. First the private keys $(\mathbf{f}, \mathbf{g}, \mathbf{F}, \mathbf{G})$ are used to form the polynomial basis $\mathbf{B}$ and its GSO $\mathbf{B}_{GS}$ (both $2N * 2N$ square matrices). Then the user's unique ID is converted into its public key form (a ring polynomial modulo q) using a random oracle. For this design, the user identity is considered to be a MAC address, which can be expressed as a 48-bit char array. In practice, the proposed software design for a random oracle relies upon a hash function and a CSPRNG; these are given in Appendix A as Algorithms 34 and 35. The design by Saarinen [2015] is used as the SHA-3 design for the hash function.

The final and most intensive stage is obtaining the user secret key using Gaussian sampling over a lattice.

In this design, software acceleration of the extract process is achieved using auto-vectorisation and more efficient use of types within the dot product and other loops in the GPV sampler over the lattice. Once the user secret key has been obtained it can be further compressed for storage or transmission. For this, Huffman coding [Huffman, 1952] is used, which is a form of lossless data compression that assigns less bits to a character that occurs more frequently.

GPV Sampler Design

The GPV sampler requires a 1-dimensional Gaussian sampler as a subroutine. A range of algorithms have been presented in the literature for such purposes, for example Bernoulli [Ducas et al., 2013b], CDT [Peikert, 2010b], Ziggurat [Buchmann et al., 2013b] and Knuth-Yao [Knuth and Yao, 1976a]. This research deploys the CDT method as it achieves the best performance in constant time.

The CDT method gives an efficient form of generating integers according to a Gaussian distribution by reducing the problem to a binary search on precomputed values of the Cumulative Distribution Function (CDF). An efficient CDT sampler has been developed in Brannigan et al. [2017] and Saarinen [2016]. This CDT method requires 16 kB to store the CDF with 64-bit precision and offers constant-time sampling. A disadvantage of the GPV algorithm is the requirement to sample over varying standard deviations, requiring the re-initialisation of the Gaussian sampler $2N$ times for a ring length of N. The initialisation time of the Gaussian sampler has been identified as a performance bottleneck, and of the range of samplers available, the CDT method was optimal in this respect. The sampling scheme has been further modified

to improve performance by reducing the number of re-initialisations from $2N$ to 2. This is achieved by noting that the standard deviation varies insignificantly for the first and latter N samples. This is because it is scaled according to the basis vector modulus, but as this is already capped at $1.17\sqrt{q}$ during basis generation this step has negligible effect. The performance of the Gaussian sampling algorithm can also be improved by pre-computing the inverse of the norm of the columns of $\mathbf{B}_{GS}$ for a given IBE master key, permitting division to be replaced with faster multiplication.

For a small cost to RAM requirements it is beneficial to store the inverse of $||\tilde{\mathbf{b}}_i||$ (see step 4 and 5 of Algorithm 14). This allows divisions to be replaced by more efficient multiplication operations. The Gaussian sampler requires repeated division by this norm and its squared value, the performance of which is much improved by utilising this method. The Gaussian sampler is also initialised for each sample that must be produced when generating a user secret key; this is quite intensive and can potentially be omitted if the standard deviation does not vary between iterations of the algorithm.

In terms of side-channel attacks, the Gaussian sampling algorithm is required only for the server-side operations of key generation and extract. In those applications where the server is vulnerable to physical access by an attacker the constant-time operation of the CDT limits timing analysis.

Gram-Schmidt Orthogonalisation

A GSO basis of the lattice is required by the GPV sampler. This process is described in §2.5.4. In order to accelerate this algorithm, this design currently uses Prest's implementation of the algorithm [Prest, 2016], which is based on improvements proposed in Lyubashevsky and Prest [2015]. This uses the isometric structure of the NTRU basis to optimise the algorithm to reach a complexity of $O(2N^2)$ as opposed to

Security level (bit)	Implementation	Encrypt (cycles)	Decrypt (cycles)
80	Software (this work, set 2)	425985	177864
	Cortex M4 [Güneysu and Oder, 2017]	13958	9530
192	Software (this work, set 5)	1111729	347041
	Cortex M4 [Güneysu and Oder, 2017]	28586	19535

Table 4.2 Comparison with hardware design of DLP-IBE

$O(2N^3)$ of the classical algorithm, where the lattice basis matrix is of dimension $2N$ x $2N$. The intuition behind this is, given an NTRU basis

$$\{\mathbf{b}, r(\mathbf{b}), r^2(\mathbf{b}), ..., r^{2N-1}(\mathbf{b})\}$$

and $\tilde{\mathbf{b}_k}$ the GSO of $\mathbf{b}_k$, then $r(\tilde{\mathbf{b}_k})$ is the GSO of $\mathbf{b}_{k+1}$. In fact, this is orthogonal to $\mathbf{b}_2, ..., \mathbf{b}_k$, but not $\mathbf{b}_1$. Therefore the final step to achieve $\tilde{\mathbf{b}}_{k+1}$ is to reduce $r(\tilde{\mathbf{b}_k})$ with respect to $\mathbf{b}_1 - Proj(\mathbf{b}_1, Span(\mathbf{b}_2, ..., \mathbf{b}_k))$.

Encryption/Decryption Design

The key generation and extract components are server-side functions in IBE, whereas encryption and decryption are seen as client-side functions and could therefore be implemented in either software or hardware. Therefore the hardware design of Güneysu and Oder [2017] could be incorporated with this software design of key generation and extract to create an even faster scheme. The encrypt and decrypt results of these two designs are compared in Table 4.2, and it can be seen that this would have a significant improvement on overall performance.

The software procedures needed here are the use of two hash functions $H(id)$ and $H'(k)$. The NTT and entropy coding are utilised again. Encrypt requires two NTTs and two inverse NTTs (steps 3 and 4 in Algorithm 15) in order to efficiently perform a number of ring modular multiplications (for smaller moduli, sparse multiplication can be used), while decrypt requires only a single NTT and inverse NTT in step1, Algorithm 16. As the master public key and user secret key are repeatedly used for this purpose their NTT representation is precomputed to reduce complexity at the expense of additional storage. Additionally, the encrypt operation requires two hash functions ($H(id)$ and $H'(\mathbf{k})$) while Decrypt requires one ($H'(k)$). How this operation is constructed is not specified in the original work and is implementation dependent. The methods in this design for mapping an arbitrary length user ID into a ring polynomial and where a random bit string is used to generate a one-time pad both use the hash function SHA-3 and a NIST AES CTR-DRBG (Counter mode - Deterministic Random Bit Generator using AES). Encrypt benefits greatly from sparse multiplication when calculating $\mathbf{r} * \mathbf{h}$ and $\mathbf{r} * H(id)$ when q is less than 26 bits, but requires less efficient NTT multiplication with larger moduli. Decryption is reliable with NTT with less frequent reduction when q is less than 26 bits, otherwise the more costly version with frequent reduction steps needs to be used, as in encrypt. Additionally, the proposed architecture of encrypt and decrypt has been modified to support any length m of the message instead of specifically N bits, at the cost of performance, but it is envisaged that the increased flexibility is more suitable for full scale testing of the practicality of the scheme.

Parameters

One of the main problems surrounding lattice-based cryptography is choosing secure yet efficient parameters. This is due to constantly evolving attacks and the use of bounds rather than concrete estimates in their analysis. The original authors suggest parameters in Ducas et al. [2014a] and these are given in Table 4.3. The security levels of these parameters are estimated in Section 5.2 of Ducas et al. [2014a] by considering both key recovery and ciphertext attacks. The encryption component of the scheme is the most vulnerable, so the security level estimations depend on the strongest attack known to recover the small errors $\mathbf{e_1}$ and $\mathbf{e_2}$, which is estimated to reach a root Hermite factor (see §2.6) of $\gamma = 1.0075$ for $N = 512$ and $\gamma = 1.0044$ for $N = 1024$.

This design increases the number of available parameter sets from two to six. This gives the user a wider range to choose from depending on security requirements and bandwidth and memory resources. For instance, a larger modulus lowers the security but may be preferred as it allows for a larger message space. Additionally, due to the use of the NTT in this design, the value of the modulus q has been changed. To apply the NTT, an $2N^{th}$ root of unity has to be found. The condition $q = 1 \mod 2N$ also needs to be satisfied in order to compute the negative wrapped convolution (see §2.5.3). Therefore, one of the contributions of this design is the proposal of suitable parameter sets, given in Table 4.4. Included is a parameter l from the encryption algorithm, which corresponds to the truncation of the ciphertext vector $\mathbf{v}$. The security level is estimated from the root Hermite factor γ introduced in Gama and Nguyen [2008], which measures the hardness of the underlying lattice problem (see §2.6). For $\gamma \approx 1.007$, 80-bit security is estimated and for $\gamma < 1.004$, 192-bit security is estimated. The bit size information for the selected parameter sets is set out in Table 4.5. Compression uses Huffman

coding [Huffman, 1952] throughout. The master public key cannot be compressed as the nature of the NTRU assumption requires it to appear random.

Table 4.3 IBE Scheme original parameters

Security parameter	**80-bit**	**192-bit**
Polynomial degree N	512	1024
Modulus q	2^{23} or 2^{24}	2^{27}

Table 4.4 Proposed IBE Scheme parameters

Parameter Set	**Root Hermite Factor**	**Security Level**	q	q **in bits**	N	l	$2N^{th}$ **root of unity**
0	1.0075	80	5767169	23	512	19	971
1	1.0079	80	10223617	24	512	20	3981
2	1.0085	$<$80	51750913	26	512	23	115658
3	1.0038	192	5767169	23	1024	18	19484
4	1.0039	192	10223617	24	1024	20	6877
5	1.0043	$<$192	51750913	26	1024	22	36945

Table 4.5 DLP-IBE key and encrypted message bit sizes. All values are in bits.

Parameter Set	**Message length**	**Master Public Key**	**Master Private Key**		**User Secret Key**		**Message**
			Uncoded	**Compressed**	**Uncoded**	**Compressed**	
0	512	11776	27648	23650	9216	7576	14336
1	512	12288	27648	23779	9216	7807	14848
2	512	13312	29696	25908	9728	8528	15360
3	1024	23552	51200	43834	17408	15522	29696
4	1024	24576	51200	44161	17408	15702	29696
5	1024	26624	59392	51458	19456	17614	31744

4.4 Performance results and discussion

The performance results have been obtained using GNU GCC 5.4.0. An Intel Core i7 6700 with both hyper-threading and TurboBoost disabled has been used, wherein the four CPUs are placed in performance mode at 4 GHz. GNU GMP 6.1.2 has been used to provide multiple precision arithmetic.

The performance results obtained are shown in Table 4.6. The proposed DLP-IBE design precomputes $\mathbf{B}$, $\mathbf{B}_{GS}$, the inverse of $||\mathbf{B}_{GS}||$ and the NTT representations of the public master key and the user secret key. The random oracles are substituted by the SHA-3 hash function and the AES CTR-DRBG random number generator. A CDT Gaussian Sampler is used to randomly sample over the lattices. It should be noted that this design performs all necessary tasks and does not require any offline computation. As the key generation is only run once per scheme setup, the time in seconds for one run-through is given. The remaining components are called multiple times (extract once per user, encrypt/decrypt once per message) and so the number of times each can be run per second is given. Extract I refers to the original extract function, while Extract II utilises compression techniques.

Table 4.6 DLP-IBE performance in terms of KeyGen processing time in seconds and Extract, Encrypt and Verify operations per second on an Intel Core i7 6700 @ 4 GHz with SHA-3, CDT Gaussian Sampling and AES CTR-DRBG

Parameter Set	q	N	**KeyGen**	**Extract I**	**Extract II**	**Encrypt**	**Decrypt**
0	5767169	512	2.666	544	521	9726	33070
1	10223617	512	2.74	541	526	9753	33359
2	51750913	512	4.034	537	527	9390	22489
3	5767169	1024	16.860	138	135	4179	17493
4	10223617	1024	23.004	137	133	3854	17568
5	51750913	1024	25.126	137	134	3598	11526

Table 4.7 Prest's NTL implementation results comparison, operations per sec [Prest, 2016]

Security Level	**Extract**		**Encrypt**		**Decrypt**	
	This work	**Prest**	**This work**	**Prest**	**This work**	**Prest**
80-bit	537	186	9390	1758	22489	2580
192-bit	137	49	3598	856	11526	1260

For reference, comparable figures from Prest's NTL-based implementation [Prest, 2016] are given in Table 4.7 (scaled up to account for differences in CPU). Prest's results were given in ms on an Intel Core i5-3210M laptop with 2.5GHz CPU and 6GB RAM so they have been converted to "per second" to represent how they would look like on a 4GHz platform: For example, 8.6ms = 0.0086s, which means 116.28 per sec at 2.5GHz, which is equivalent to 116.38/2.5 * 4 =186.04 per sec at 4GHz. One can compare Prest's 80-bit and 192-bit security results to parameter sets 2 and 5 respectively of this work, without compression. It can be seen that the proposed software architecture outperforms the original PoC in all respects, for example for $N = 512, q = 2^{24}$ (or 80-bit security), the proposed extract function can operate at three times the speed of Prest's design, and the encrypt/decrypt over x5 and x9 times respectively. This accounts to a x7 speedup of the full primitive runtime. Note that Prest did not include key generation results. However, when running both implementations on the same platform (Intel Core i7 6700 @ 4 GHz), optimal performance of the key generation algorithm (i.e. the first f and g polynomials that are randomly selected are within bounds) is now ≈0.3 seconds slower in the design presented here than Prest's PoC implementation [Prest, 2016], specifically 2.7 seconds vs 2.4 seconds. This is because the design proposed here enforces the security threshold for sampling small polynomials, unlike the previous PoC, as discussed in §4.3.2.

Table 4.8 Performance comparison of conventional schemes (results from [Cheng et al., 2011], scaled to 4GHz) and the DLP-IBE scheme (this work).

Scheme	**Extract (cycles)**	**Encrypt (MB/s)**	**Decrypt (MB/s)**
Cocks	4644800	0.000396	3.533
Boneh-Franklin	9378664	2.7	3.15
DLP-IBE (This work)	7448790	13.146	44.978

In comparison to classical IBE schemes, the DLP lattice-based IBE scheme also has respectable performance. In 2011, performance testing of the Boneh-Franklin IBE scheme [Cheng et al., 2011] on a Pentium Dual T2330 at 1.60GHz reported that Extraction could be run at 170.6 ops/s, Encrypt at 1.08 op/s and Decrypt at 1.26 op/s. Table 4.8 presents the performance results from [Cheng et al., 2011] scaled to a 4GHz platform. DLP-IBE outperforms the classical IBE schemes in most aspects, for instance x1.2 the speed for extract, x5 for encrypt and x15 for decrypt of the Boneh-Franklin components. The Cocks extract component outperforms DLP-IBE by 40%. However, the slow encrypt/decrypt times of Cocks, together with its large ciphertext overhead of 1000 times the plaintext, renders it impractical for real applications. Therefore, this research shows that replacing current schemes with post-quantum schemes will improve security without impacting efficiency and in fact outperforms conventional IBE schemes.

Yet, some applications may require even finer grained access control. The next chapter focuses, for the first time, on creating a scalable hierarchical-IBE scheme.

4.5 The Falcon-IBE Scheme

The Falcon signature scheme Prest et al. [2017] naturally lends itself to an IBE which can be viewed as a more compact and faster version of the DLP-IBE. This becomes the basis of the novel HIBE scheme introduced in Chapter 5 and so it is now described.

4.5.1 Description of the Falcon-IBE Scheme

The master key generation, given in Algorithm 17 is similar to that of the Falcon signature scheme. The master key pair is a public/private NTRU lattice basis pair, yet this algorithm utilises the faster “tower of rings" approach to solve the NTRU equation.

The extract procedure, given in Algorithm 18, can be derived from the Falcon signing process. In the signature scheme, a short lattice vector is sampled to create the signature. In Falcon-IBE, this short vector becomes the secret key associated with a specified user ID.

The encrypt and decrypt components are directly taken from the DLP-IBE scheme, as described in Algorithms 15 and 16.

4.6 Conclusions

The chapter has proposed the first optimised software design of a quantum-safe IBE scheme. This has been achieved through both algorithmic and software optimisations. Additionally, revised parameter sets have been proposed to allow for faster multiplication methods to be utilised and a wider range of security targets. Not only has this design improved upon the existing PoC implementation of Prest [2016] by 7 times,

Algorithm 17: FALCON key generation (adapted from Prest et al. [2017])

Data: N, q
Result: $\mathbf{B} \in \mathbb{Z}_q^{2N \times 2N}$, $h \in \mathcal{R}_q$

1 $\sigma_f = 1.17\sqrt{\frac{q}{2N}}$
2 $f, g, \leftarrow \mathcal{D}_{N,\sigma_f}$
3 Norm $\leftarrow max\left(||g, -f||, \left|\left|\left(\frac{q\bar{f}}{f*\bar{f}+g*\bar{g}}, \frac{q\bar{g}}{f*\bar{f}+g*\bar{g}}\right)\right|\right|\right)$
4 **if** *Norm* $> 1.17\sqrt{q}$ **then** go to Step 2;
5 Compute $\rho_f, \rho_g \in \mathcal{R}$ and $R_f, R_g \in \mathbb{Z}$ such that: $\rho_f \cdot f = R_f$ and $\rho_g \cdot g = R_g$
6 **if** $GCD(R_f, R_g) \neq 1$ *or* $GCD(R_f, q) \neq 1$ **then** go to Step 2;
7 Compute $u, v \in \mathbb{Z}$ such that: $u \cdot R_f + v \cdot R_g = 1$
8 $F \leftarrow qv\rho_g$ and $Q \leftarrow -qu\rho_f$
9 $k = \left\lfloor \frac{F*\bar{f}+G*\bar{g}}{f*\bar{f}+g*\bar{g}} \right\rceil \in \mathcal{R}$
10 $F \leftarrow F - k * f$ and $G \leftarrow G - k * g$
11 $h = g * f^{-1} mod\, q$
12 $\mathbf{B} = \begin{pmatrix} g & -f \\ G & -F \end{pmatrix}$
13 $\hat{\mathbf{B}} = \begin{pmatrix} FFT(g) & -FFT(f) \\ FFT(G) & -FFT(F) \end{pmatrix}$
14 T=ffLDL*(G)
15 $\sigma \leftarrow 1.55\sqrt{q}$
16 for each leaf of T, leaf.value $= \sigma/\sqrt{\text{leaf.value}}$
17 sk $\leftarrow (\hat{\mathbf{B}}, \text{T})$, pk $= h$
18 **return** sk, pk

but it has been shown to outperform existing pairings-based schemes used today by 7 times also. This indicates that lattice-based IBE has potential for future deployment as a quantum-safe alternative.

To further emphasise the potential of this scheme for deployment, concurrent hardware results of the IBE encrypt/decrypt functionality as an output of the SAFE-crypto project [O'Neill et al., 2016] are shown in Table 4.9. The software results for key generation and extraction, combined with encrypt and decrypt on hardware, would accommodate for an optimal IBE deployment.

Algorithm 18: FALCON Extract (adapted from Prest et al. [2017])

Data: sk, $H : \{0,1\}^* \to \mathbb{Z}_q^N$, m, β
Result: SIG_m

if SK_m *is in local storage* **then**
  return Output SIG_m to message m
else
  $r \leftarrow$ Unif($\{0,1\}^{320}$)
  $c \leftarrow H(r||m)$;
  $t \leftarrow (FFT(c), FFT(0)) \cdot \hat{\mathbf{B}}^{-1}$ $z \leftarrow$ ffSampling(t, T)
  $s = (t - z)\hat{\mathbf{B}}$
  while $||s|| > \beta$
  $(s_1, s_2) \leftarrow invFFT(s)$
  $s_2 =$ Compress(s_2)
  $SIG_m \leftarrow (r, s_2)$ **return** Output SIG_m associated to message m and keep in local storage

Table 4.9 Hardware performance results of DLP-IBE scheme (cycles) [O'Neill et al., 2016]

	N=**512**, q=**16813057**			N=**1024**, q=**134348801**		
	Cortex M0	**Cortex M4**	**Spartan FGPA**	**Cortex M0**	**Cortex M4**	**Spartan FPGA**
IBE Encryption	3297380	972744	13958	6202910	1719444	28586
IBE Decryption	1155000	318539	9530	2171000	557015	19535

CHAPTER 5

Skinny Latte: A Novel Scalable Hierarchical Identity-Based Encryption Scheme

5.1 Introduction

One of the advantages of LBC is the ability to build advanced cryptographic primitives such as IBE and ABE. HIBE schemes extend the concept of using a personal identity as a public key to a multi-levelled scenario, such as one would find within a functioning company. HIBE has further applications such as forward-secure encryption and public key broadcast encryption. However, it is still unknown territory within the post-quantum field. Additionally, with the growth of IoT, which brings with it complex interconnected systems of constrained devices, there is a greater requirement for lightweight, advanced primitives unlike ever before. The long-term security considerations indicate that these should be made quantum-secure today. The aim of this chapter is to develop a scalable, quantum-safe HIBE scheme.

Thus far in this thesis, lattice-based signature schemes have been considered and the scheme Falcon has been extended into an IBE scheme. However, the DLP-IBE

scheme [Ducas et al., 2014a] considered in Chapter 4 was in 2017 combined with the Bonsai tree HIBE scheme introduced in Cash et al. [2010] to create Latte by Campbell and Groves [2017]. This research was carried out by the NCSC, with a view to utilising the scheme in UK public-safety communications. They are currently working with ETSI in a move towards standardising the scheme. However, LATTE is not practical beyond 2/3 levels. This is due to user public key sizes reaching 6kB (kilobytes) and ciphertext sizes reaching 15kB for three levels at 128-bit security.

The main contribution of this chapter is a more practical variant of LATTE. To mitigate the problems of Latte, techniques from the fixed dimension HIBE over standard lattices by Agrawal et al. [2010c] are applied to create Skinny Latte which instead fixes the dimension during delegation, curtailing the growth of public key and ciphertext size. The fast, compact FFT sampler and the NTRU polynomial generation components from Prest et al. [2017] are utilised to increase the efficiency of the scheme. A theoretical analysis of key and ciphertext sizes and security is given. This chapter proposes Skinny Latte as a more practical quantum-safe solution.

Whilst assessing the novel Skinny Latte variant, it was discovered that there were no existing benchmarking results of a quantum-safe HIBE scheme. Therefore software design and benchmarking of the HIBE scheme Latte by Campbell and Groves [2017] has been carried out as part of this research. As it shares many components with the DLP-IBE scheme, many of the optimisations from Chapter 4 apply.

The contributions of this chapter are:

- Skinny Latte : the first NTRU-based HIBE scheme with fixed dimension
- Security analysis of Skinny Latte
- First performance benchmarking of a quantum-safe HIBE scheme, Latte

Section 5.2 gives a background to the lattice-based concepts used in (H)IBE schemes. Section 5.3 describes the existing LATTE HIBE scheme and characterises its strengths and limitations, alongside performance results. Section 5.4 introduces SKINNY LATTE and gives a theoretical comparison and security analysis. The chapter finishes with conclusions in Section 5.5.

5.2 Hierarchical Identity-based Encryption

Hierarchical identity-based encryption (HIBE) schemes were introduced by Horwitz and Lynn [2002] and can be considered a generalisation of an IBE scheme to multiple levels. An HIBE scheme consists of five components: master key generation, key delegation, user key extraction, encryption and decryption. Figure 5.1 shows the relationship between the components of a 2-level HIBE scheme. The steps are as follows:

1. The master key generator establishes the master public and private key s
2. Through a delegation function, the master key generator creates a public/private key pair for the sub key manager. This gives it the ability to delegate further key pairs, and extract user private keys along the hierarchy
3. The sub key manager delegates a further public/private key to the next level of the hierarchy
4. The extractor uses their public/private key pair to extract and share user public/private keys, as in the single-level IBE scheme
5. Encrypt/decrypt works as in IBE

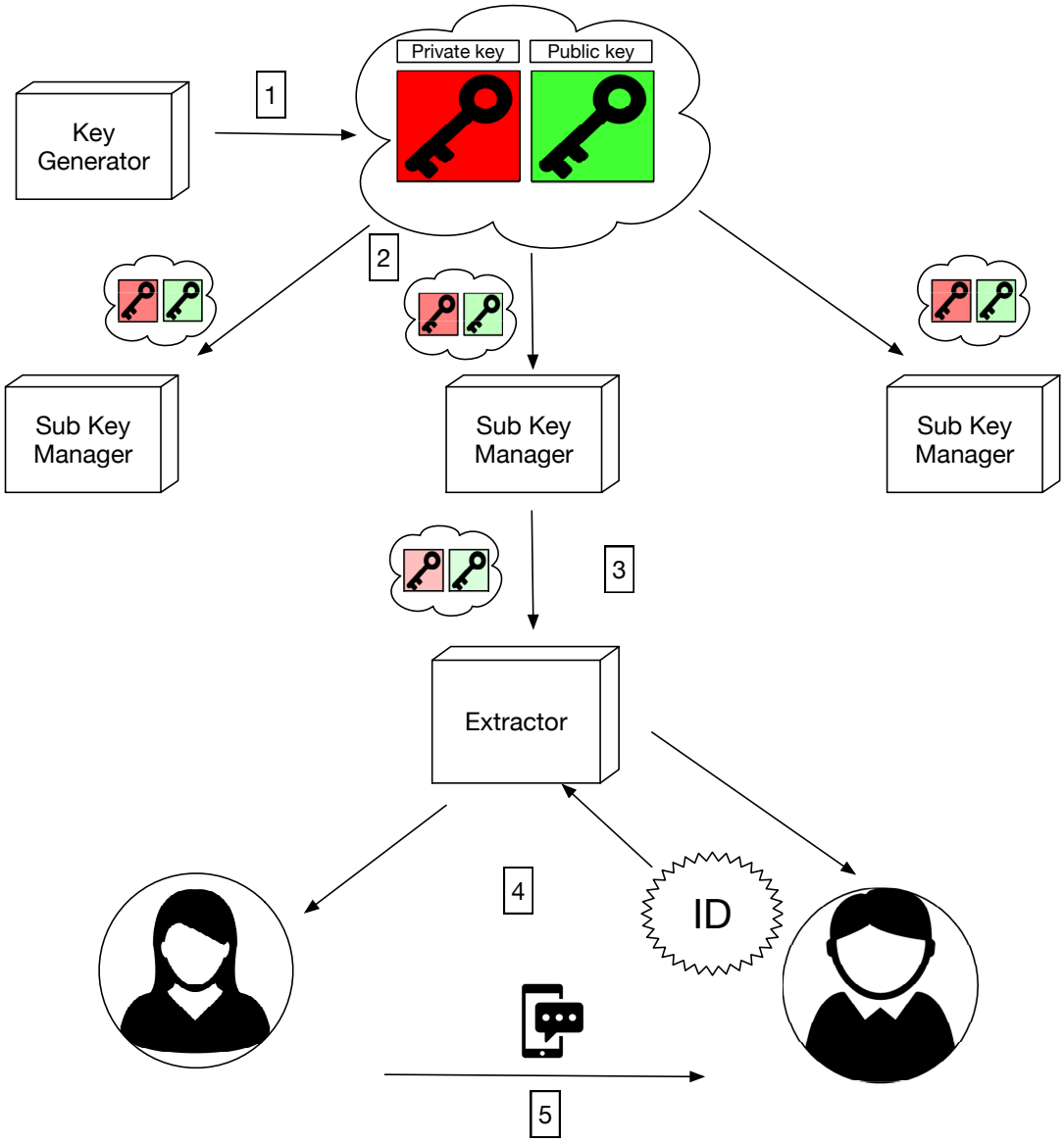


Fig. 5.1 A 2-level HIBE scheme

Advantages and additional capabilities of HIBE over IBE include:

- HIBE limits damage performed and access gained by compromise of a KMS, as an attacker would only gain control over those users controlled by the sub-KMS, rather than all the users in the system
- HIBE creates finer-grained control over key management within an organisation as responsibility of distributing keys can be allocated to groups further down the hierarchy. Similarly, for public-safety communications, the headquarters could retain control of the central KMS, whilst allowing local provisioning of users by sub-KMSs situated in regional emergency stations
- HIBE reduces the workload of the master key generator
- HIBE can be extended into more advanced primitives such as generating short-lived keys for portable computing devices [Horwitz and Lynn, 2002] and turning the NNL broadcast encryption system [Naor et al., 2001] into a public-key broadcast system.

This chapter begins with an introduction to non-quantum-safe HIBE schemes, which are based on elliptic curves and pairings. Gentry and Silverberg proposed the first secure HIBE scheme in the random oracle model [Gentry and Silverberg, 2002], which was an extension of the Boneh-Franklin IBE scheme [Boneh and Franklin, 2003], a Weil-pairing based scheme, the security of which relies on the bilinear Diffie-Hellman problem. This was shown to be secure against adaptive identity and chosen ciphertext attacks, by use of the Fujisaki-Okamoto transformation [Fujisaki and Okamoto, 1999], although the security degrades exponentially with the number of levels.

Boneh-Boyen built on this to create a scheme which was secure without random oracles [Boneh and Boyen, 2004]. However, as both the ciphertext and private keys grew linearly with the number of levels of the hierarchy, in Boneh et al. [2005] a scheme was proposed which fixed the ciphertext size to three group elements, and curtailed private key growth to within level l group elements. [1] In 2019, an isogeny-based version of the Decisional Bilinear Diffie-Hellman-based scheme was proposed [Koshiba and Takashima, 2019]. Despite isogenies possessing quantum-safe properties, this variant only serves to strengthen the existing classical security, by proving it secure under the assumption of *either* the classical version or the isogeny-based version of the problem and therefore is not necessarily quantum-safe. To the best of the author's knowledge, the only quantum-safe HIBE schemes so far proposed are based on lattices. These are elaborated on in the next section as they are essential to understanding the new HIBE scheme proposed in this chapter.

5.2.1 Existing lattice-based (H)IBE schemes

DLP IBE scheme

In 2014, Ducas, Lyubashevsky and Prest proposed the first efficient lattice-based identity-based encryption scheme [Ducas et al., 2014a], as detailed in Chapter 4. They based their construction on the IBE scheme by Gentry, Peikert and Vaikuntanathan [Gentry et al., 2007], using a variant of NTRU lattices. The underlying security problems are the NTRU problem for key generation and R-LWE for the encryption. The ciphertexts therefore have more practical sizes than previous constructions, for example 30Kbits for 192-bit classical security. The use of structured lattices also

[1] In (H)IBE schemes, user identities at level l are here denoted ID_l.

allowed for implementation optimisations such as the Number Theoretic Transform (NTT), as demonstrated by McCarthy et al. [2017], whose software performance of the DLP-IBE outperformed that of the elliptic-curve based Boneh-Franklin IBE scheme. The DLP-IBE scheme is covered in detail in Chapter 4.

Bonsai Trees HIBE

Cash et al. [2010, 2012] proposed the use of Bonsai trees to create a hierarchical structure for IBE. They model the hierarchical network of users as a tree, whereby arborists, or sub-key-managers, have control over the sub-trees, and have the authority to delegate user private keys. Delegation requires the knowledge of a trapdoor basis of the lattice at that level. During the process whereby keys are delegated down the tree, the lattice is extended, and therefore its dimension and hence the key and ciphertext sizes increase. The public key is $O(d^3kn^2)$ and ciphertext $O(d^3kn)$ at depth d, security parameter n and hash output length k. The root authority has control of the whole tree by knowing the short trapdoor basis for the master root lattice. The security of this HIBE scheme is based on LWE over standard lattices.

Fixed dimension HIBE

In 2010, Agrawal, Boneh and Boyen [Agrawal et al., 2010c] presented a technique for keeping the lattice dimension constant upon delegation. This means that private keys and ciphertexts also remain a constant length as the hierarchical depth increases, improving efficiency and scalability. Their technique is described as follows. Suppose there exists a uniform public matrix $\mathbf{A} = \{\mathbf{a}_1, ..., \mathbf{a}_n\}$ at level $l-1$ with a short basis $T_{\mathbf{A}}$ and the KMS wishes to delegate public/private keys to a user at level l. It can compute a low-norm matrix $\mathbf{R}$, depending on the user identity, and set the new user public basis

to be $\mathbf{AR}^{-1}$. They define a linearly independent set $\mathbf{S}_l = \{\mathbf{R^{-1}a_1}, ..., \mathbf{R^{-1}a_n}\}$ in this matrix, and use algorithms TrapGen from [Alwen and Peikert, 2009] and RandBasis from [Cash et al., 2010] to generate a short, randomised basis for $\Lambda(\mathbf{AR}^{-1})$. The public key and ciphertext sizes remain constant at each level, and are $O(k^2 n d_{max}^2)$ and $O(k^3 n^2 d_{max}^3)$ respectively, where k is the number of bits in identity with security parameter n, maximum depth d_{max} and lattice dimension $O(k d_{max} n)$ determined at setup.

5.3 LATTE

LATTE was proposed in 2017 [Campbell and Groves, 2017] and can be considered as a combination of the DLP-IBE scheme [Ducas et al., 2014a] and Bonsai Trees HIBE scheme [Cash et al., 2010] to create a hierarchical lattice-based IBE scheme. Table 5.1 summarises the LATTE algorithms.

The key generation, given in Algorithm 19, generates an NTRU-type basis (as in §2.4.2). This is performed by sampling the short basis polynomials $\mathbf{f}, \mathbf{g}$ from a Gaussian distribution. Operations are over the polynomial ring $\mathcal{R} = \mathbb{Z}_q[x] \big/ \langle x^N + 1 \rangle$, a variant of the NTRU ring. The Gram-Schmidt norm of the associated basis is computed to ensure it is small enough to allow for short private keys to be delegated to the next level. If not, the polynomials are resampled. The rest of the basis, polynomials F, G, are computed as in the DLP scheme [Ducas et al., 2014a], and the public key polynomial $\mathbf{h} = \mathbf{g} \cdot \mathbf{f}^{-1}$ is computed. An additional public polynomial $\mathbf{b}$ is sampled from a uniformly random distribution.

The delegation process, given in Algorithm 20, creates a public/secret key pair for the next level in the tree, allowing it to become a sub-KMS. Suppose the KMS

Table 5.1 LATTE Algorithm summary

Algorithm	Inputs	Outputs
Master KeyGen	$N, q \in \mathbb{Z}, \sigma_0 \in \mathbb{R}$	$\mathbf{S_0}, \mathbf{B_0} \in \mathbb{Z}_q^{2N\times 2N}, \mathbf{b} \in \mathcal{R}_q$
Delegation	$\sigma_l \in \mathbb{R}, \mathbf{S}_{l-1} \in \mathbb{Z}_q^{2N\times 2N}, id_l$ $H : \{0,1\}^* \to \mathbb{Z}_q^N$	$\mathbf{S}_l \in \mathbb{Z}_q^{(l+2)N\times(l+2)N}$
Extract	$\sigma_l \in \mathbb{R}, \mathbf{S}_{l-1} \in \mathbb{Z}_q^{2N\times 2N}, \mathrm{id}_l$ $H : \{0,1\}^* \to \mathbb{Z}_q^N$	$\mathbf{t_0}, ..., \mathbf{t_{l+1}} \in \mathcal{R}$
Encryption	$\sigma_e \in \mathbb{R}, b \in \mathcal{R}_q$ $KDF, \mathrm{id}_l, \mu \in \{0,1\}^{256}$ $H : \{0,1\}^* \to \mathbb{Z}_q^N$, $H' : \{0,1\}^N \to \{0,1\}^m$	$Z, C_0, ..., C_l, C_{l+1} \in \mathcal{R}_q$
Decryption	$Z, C_0, ..., C_l, C_{l+1} \in \mathcal{R}_q$	$\mu' \in \{0,1\}^{256}$

Algorithm 19: The LATTE key generation algorithm [ETS, 2019]

Data: N, q, σ_0
Result: $\mathbf{S_0}, \mathbf{B} \in \mathbb{Z}_q^{2N\times 2N}$, $\mathbf{h} \in \mathcal{R}_q$,

1 $\mathbf{f}, \mathbf{g}, \leftarrow \mathcal{D}_{N,\sigma_0}$
2 Norm $\leftarrow max\left(||\mathbf{g}, -\mathbf{f}||, \left|\left|\left(\frac{q\bar{\mathbf{f}}}{\mathbf{f}*\bar{\mathbf{f}}+\mathbf{g}*\bar{\mathbf{g}}}, \frac{q\bar{\mathbf{g}}}{\mathbf{f}*\bar{\mathbf{f}}+\mathbf{g}*\bar{\mathbf{g}}}\right)\right|\right|\right)$
3 **if** *Norm* $> \sigma_0 \cdot \sqrt{2N}$ **then** go to Step 1;
4 Compute $\rho_\mathbf{f}, \rho_\mathbf{g} \in \mathcal{R}$ and $r_\mathbf{f}, r_\mathbf{g} \in \mathbb{Z}$ such that: $\rho_\mathbf{f} \cdot \mathbf{f} = r_\mathbf{f}$ and $\rho_\mathbf{g} \cdot \mathbf{g} = r_\mathbf{g}$
5 **if** $GCD(r_\mathbf{f}, r_\mathbf{g}) > 1$ *or* $r_\mathbf{f} = 0 \mod q$ **then** go to Step 1;
6 Compute $v_\mathbf{f}, v_\mathbf{g} \in \mathbb{Z}$ such that: $v_\mathbf{f} \cdot r_\mathbf{f} + v_\mathbf{g} \cdot r_\mathbf{g} = 1$
7 $F \leftarrow q v_\mathbf{g} \rho_\mathbf{g}$ and $G \leftarrow -q v_\mathbf{f} \rho_\mathbf{f}$
8 $k = \left\lfloor \frac{F*\bar{\mathbf{f}}+G*\bar{\mathbf{g}}}{\mathbf{f}*\bar{\mathbf{f}}+\mathbf{g}*\bar{\mathbf{g}}} \right\rceil \in \mathcal{R}$
9 $F \leftarrow F - k * \mathbf{f}$ and $G \leftarrow G - k * \mathbf{g}$
10 $\mathbf{b} \leftarrow \mathcal{U}$
11 **return** $\mathbf{h} = \mathbf{g} * \mathbf{f}^{-1} mod\, q$
12 and $\mathbf{S_0} = \begin{pmatrix} \mathcal{A}(\mathbf{g}) & -\mathcal{A}(\mathbf{f}) \\ \mathcal{A}(G) & -\mathcal{A}(F) \end{pmatrix}$
13 and $\mathbf{b}$

wishes to delegate a key from level $l-1$ to level l. Then it can extend the public basis of the user at level l by appending $A_l = H(ID_1|...|ID_l)$, where H is a hash function (considered as per §2.5.1), to the first column. The dimension of the new

matrix becomes $(l+2)N \times (l+2)N$. The corresponding secret basis is generated by sampling short vectors (using the GPV Sampler by Gentry et al. [2007], introduced in §2.5.5) from the level $(l-1)$ lattice using its secret basis. These are reduced similarly to in the key generation component to ensure the basis is of the required quality for further delegation.

In the LATTE key extraction algorithm, the user private key is a short solution $(\mathbf{t_0}, \mathbf{t_1}, ..., \mathbf{t_{l+1}})$ to

$$A \cdot \mathbf{t}_0 + A_1 \cdot \mathbf{t}_1 + ... + A_l \cdot \mathbf{t}_l + \mathbf{t}_{l+1} = \mathbf{b}$$

(Algorithm 21). Again, this is found using the GPV Sampler over the short basis from the previous level. An extended version of Ring-LWE encrypt/decrypt is used for ciphering messages as given in Algorithms 22 and 23. The message is encrypted using a seed, and then this seed is encrypted and sent. The ciphertext consists of the encrypted message Z, ephemeral public keys $C_0, ..., C_l$ and the encrypted seed C_{l+1}. This is a variant of the Fujisaki-Okamoto transform [Fujisaki and Okamoto, 1999] to protect against invalid ciphertexts. The decryption process uses the user private key to decrypt the seed and compute the message. The parameter sets are given in Table 5.2 and further details can be found in Campbell and Groves [2017]. The toy parameter is not recommended for real world use and so its security is not analysed.

Algorithm 20: The LATTE key delegation algorithm (delegating from level $l-1$ to level l) [ETS, 2019]

Data: $N, q, \sigma_l, \mathbf{S}_{l-1} H : \{0,1\}^* \to \mathbb{Z}_q^N, id_l$

Result: $\mathbf{S}_l \in \mathbb{Z}_q^{(l+2)N \times (l+2)N}$

1 $A_i = H(ID_1|...|ID_i)$ for $i = 1, ...l$

2 **for** $i \in \{0, ..., l\}$ **do**

3 $\quad \mathbf{s}_{i,l+1} \leftarrow \mathcal{D}_{N,\sigma_l}$

4 $\quad (\mathbf{s}_{i,0}, \mathbf{s}_{i,1}, ..., \mathbf{s}_{i,l}) \leftarrow GPVSampler(\mathbf{S}_{l-1}, (-A_i \cdot \mathbf{s}_{i,l+1}, 0, ..., 0), \sigma_l)$

5 $\quad$ **if** $||(\mathbf{s}_{i,0}, \mathbf{s}_{i,1}, ..., \mathbf{s}_{i,l}, \mathbf{s}_{i,l+1})|| > \sqrt{(l+2)N} \cdot \sigma_l$ **then**

6 $\quad\quad$ resample

7 **for** $j \in \{0, ..., l+1\}$ **do**

8 $\quad M_j = (-1)^{j+l+1} det \begin{pmatrix} \mathbf{s}_{0,0} & ... & \mathbf{s}_{0,j-1} & \mathbf{s}_{0,j+1} & ... & \mathbf{s}_{0,l+1} \\ \mathbf{s}_{1,0} & ... & \mathbf{s}_{1,j-1} & \mathbf{s}_{1,j+1} & ... & \mathbf{s}_{1,l+1} \\ \vdots & & \vdots & \vdots & & \vdots \\ \mathbf{s}_{l,0} & ... & \mathbf{s}_{l,j-1} & \mathbf{s}_{l,j+1} & ... & \mathbf{s}_{l,l+1} \end{pmatrix}$

9 $\quad$ Compute $r_j \in \mathbb{Z}, \mathbf{u}_j \in \mathcal{R}$ s.t. $M_j \cdot \mathbf{u}_j = r_j$

10 **if** *possible* **then**

11 $\quad$ Compute (via a series of Extended Euclidean Algorithms) $v_0, ..., v_{l+1} \in \mathbb{Z}$ s.t $v_0 r_0 + ... + v_{l+1} r_{l+1} = 1$

12 **else**

13 $\quad$ go to Step 2

14 **for** $j \in \{0, ..., l+1\}$ **do**

15 $\quad$ Define $\mathbf{s}_{l+1,j} = q v_j \cdot \mathbf{u}_j \in \mathcal{R}$

16 **if** $GCD(v_k, r_k) > 1$ (*for* $k = 1, ..., l+1$) *or* $r_j = 0 \mod q$ **then**

17 $\quad$ go to Step 2

18 Set $\mathbf{C} = \begin{pmatrix} c_{0,0} & c_{0,1} & ... & c_{0,l} \\ c_{1,0} & c_{1,1} & ... & c_{1,l} \\ \vdots & & \ddots & \vdots \\ c_{l,0} & c_{l,1} & ... & c_{l,l} \end{pmatrix}$, where $c_{i,j} = \mathbf{s}_{j,0} \cdot \mathbf{s}_{\bar{i},o} + ... + \mathbf{s}_{j,l+1} \cdot \mathbf{s}_{\bar{i},l+1}$, $d_i = \mathbf{s}_{l+1,0} \cdot \mathbf{s}_{\bar{i},0} + ... + \mathbf{s}_{l+1,l+1} \cdot \mathbf{s}_{\bar{i},l+1}$ and $\mathbf{C}_i(d)$ is the matrix $\mathbf{C}$ with its ith column replaced by d_i: $\begin{pmatrix} c_{0,0} & c_{0,1} & ... & c_{0,i-1} & d_0 & c_{0,i+1} & ... & c_{0,l} \\ c_{1,0} & c_{1,1} & ... & c_{1,i-1} & d_1 & c_{1,i+1} & ... & c_{1,l} \\ \vdots & & \ddots & \vdots & & & & \\ c_{l,0} & c_{l,1} & ... & c_{l,i-1} & d_l & c_{l,i+1} & ... & c_{l,l} \end{pmatrix}$.

19 Then, $k = (k_i)_{0 \le i \le n}$ is the solution to $\mathbf{C}k = d$. By Cramer's rule, $k_i = \frac{det(\mathbf{C}_i(d))}{det(\mathbf{C})}$.

20 **for** $i \in \{0, ..., l\}$ **do**

21 $\quad (\mathbf{s}_{l+1,0}, ..., \mathbf{s}_{l+1,l+1}) = (\mathbf{s}_{l+1,0}, ..., \mathbf{s}_{l+1,l+1}) - \lfloor k_i \rceil \cdot (\mathbf{s}_{i,0}, ..., \mathbf{s}_{i,l+1})$

22 **return** $\mathbf{S}_l = \begin{pmatrix} \mathbf{s}_{0,0} & \mathbf{s}_{0,1} & ... & \mathbf{s}_{0,l+1} \\ \mathbf{s}_{1,0} & \mathbf{s}_{1,1} & ... & \mathbf{s}_{1,l+1} \\ \vdots & & \ddots & \vdots \\ \mathbf{s}_{l+1,0} & \mathbf{s}_{l+1,1} & ... & \mathbf{s}_{l+1,l+1} \end{pmatrix}$

Algorithm 21: The LATTE user key extraction algorithm (extracting at level l) [ETS, 2019]

Data: $N, q, \sigma_l, \mathbf{S}_l, H : \{0,1\}^* \to \mathbb{Z}_q^N, id_l$
Result: $\mathbf{t_0}, ..., \mathbf{t_{l+1}} \in \mathcal{R}$
$A_i = H(ID_1|...|ID_i)$ for $i = 1, ...l$
$\mathbf{t}_l \leftarrow \mathcal{D}_{N,\sigma_l}$
$(\mathbf{t_0}, \mathbf{t_1}, ..., \mathbf{t_{l-1}}, \mathbf{t_{l+1}}) \leftarrow GPVSampler(\mathbf{S}_l, \sigma_l)$
return $\mathbf{t_0}, ..., \mathbf{t_{l+1}} \in \mathcal{R}$

Algorithm 22: The LATTE encryption algorithm [ETS, 2019]

Data: $N, q, \sigma_e, \mathbf{b}, KDF, id_l, \mu \in \{0,1\}^{256}$
Result: $(Z, C_0, ..., C_l, C_{l+1})$
$seed \leftarrow \{0,1\}^{256}$
$Z = \mu \oplus KDF(seed)$
$\mathbf{e}, \mathbf{e_0}, ..., \mathbf{e_{l+1}} \leftarrow \mathcal{D}_{N,\sigma_e}$
for *i=0,...,l* **do**
  $C_i = A_i \cdot \mathbf{e} + \mathbf{e}_i$ where $A_0 = \mathbf{h}$ and $A_i = H(ID_1|...|ID_i)$ for $i = 1, ...l$
$\mathbf{m} \leftarrow Encode(seed)$
$C_{l+1} = \mathbf{b} \cdot \mathbf{e} + \mathbf{e}_{l+1} + \mathbf{m}$
Drop the least significant bits of $C_{l+1} : C_{l+1} \leftarrow q\lfloor C_{l+1}/q \rceil \mod q$
return $(Z, C_0, ..., C_l, C_{l+1})$

Algorithm 23: The LATTE decryption algorithm [ETS, 2019]

Data: $N, q, (Z, C_0, ..., C_l, C_{l+1})$
Result: μ'
Decompress $C_{l+1} : C_{l+1} \leftarrow \lfloor qC_{l+1}/q \rceil \mod q$
Compute $V = C_{l+1} - C_o \cdot t_0 - \ldots - C_l \cdot t_l$, lift to a polynomial in $\mathcal{R}$
$seed' \leftarrow Decode(V)$
$\mathbf{e}, \mathbf{e'_0}, ..., \mathbf{e'_{l+1}} \leftarrow \mathcal{D}_{N,\sigma_e}$
for *i=0,...,l* **do**
  $C'_i = A_i \cdot \mathbf{e}' + \mathbf{e}'_i$ where $A_0 = \mathbf{h}$ and $A_i = H(ID_1|...|ID_i)$ for $i = 1, ...l$
$\mathbf{m}' \leftarrow Encode(seed')$
$C'_{l+1} = \mathbf{b} \cdot \mathbf{e}' + \mathbf{e}'_{l+1} + \mathbf{m}'$
Drop the least significant bits of $C'_{l+1} : C'_{l+1} \leftarrow q\lfloor C_{l+1}/q \rceil \mod q$
Check $(C'_0, ..., C'_l, C'_{l+1})$ agree with $(C_0, ..., C_l, C_{l+1})$
return $\mu' = Z \oplus KDF(seed')$

Table 5.2 LATTE Parameters

Set	Security	N	q	σ_0
LATTE-0	-	256	$2^{32} - 2^{20} + 1$	3388.3
LATTE-1	128	1024	$2^{24} - 2^{14} + 1$	105.9
LATTE-2	256	2048	$2^{25} - 2^{12} + 1$	105.9
LATTE-3	80	1024	$2^{36} \smile 2^{20} + 1$	6 777.4
LATTE-4	160	2048	$2^{38} \smile 2^{26} + 1$	9 583.5

Table 5.3 LATTE Public Key and Ciphertext Sizes (bytes)

Parameter Set	Classical Security Level (bits)	Master Public Key	Ciphertext		
			$l = 1$	$l = 2$	$l = 3$
LATTE-1	128	6144	9248	12320	15392
LATTE-2	256	12800	19232	25632	32032
LATTE-3	80	9216	13856	18464	23072
LATTE-4	160	19456	29216	38944	48672

5.3.1 Theoretical Analysis

Key and Ciphertext Sizes

The key and ciphertext sizes for LATTE are given in Table 5.3, for up to a 3-level deep hierarchy. Key sizes are large in comparison to elliptic curve based schemes.

Security

The security of each component of LATTE depends on an associated lattice problem. For key generation, this is the NTRU problem; finding short vectors $\mathbf{f}, \mathbf{g}$ from $\mathbf{h} = \mathbf{g} \cdot \mathbf{f}^{-1}$, which is assumed to be computationally infeasible. However, we can use known methods in cryptanalysis to reduce the basis and make the problem easier to break. For key delegation, it amounts to solving the SVP in Λ_{l-1} and for user key extraction, solving CVP in Λ_l. Message recovery depends on solving an extended version of

Table 5.4 Latte Estimated cost of delegated key recovery

N	l	β	**Classical Security**	**Quantum Security**
1024	1	1024	316	288
	2	391	131	120
2048	1	2239	670	610
	2	908	282	257
	3	513	166	152
4096	2	2015	605	550
	3	1179	361	329
	4	793	248	227

Table 5.5 Latte Estimated cost of user key recovery

N	l	β	**Classical Security**	**Quantum Security**
1024	1	787	246	225
	2	329	113	104
2048	1	1777	535	487
	2	785	246	224
	3	458	150	138
4096	2	1771	534	486
	3	1069	329	300
	4	731	230	210

R-LWE, which reduces to an instance of the primal-CVP or dual-SVP. In the primal-CVP attack, the ephemeral private keys are recovered via a close vector problem. In the dual-SVP attack an attempt is made to distinguish the ciphertext elements from uniformly random polynomials in $\mathcal{R}_q$.

Table 5.6 LATTE performance results (op/s)

		$l = 0$			$l = 1$			
Set	**Keygen (s)**	**Ext**	**Enc**	**Dec**	**Del**	**Ext**	**Enc**	**Dec**
LATTE-0	203.1	262.9	1108.2	2900.9	0.183	50.167	936.5	2085.1
LATTE-1	117.2	41.8	297.9	707.3	0.011	3.43	262.39	498.18
LATTE-2	1013.38	10.88	163.8	374.2	0.002	0.887	141.6	256.3

	$l = 2$				$l = 3$			
Set	**Del**	**Ext**	**Enc**	**Dec**	**Del**	**Ext**	**Enc**	**Dec**
LATTE-0	0.342	25.5	833.3	1676.1	0.2	14.4	740.9	1380.5
LATTE-1	0.018	1.67	233.15	382.54	0.010	0.940	210.26	311.12
LATTE-2	0.002	0.433	118.4	189.2	-	-	-	-

5.3.2 Performance

The performance results have been obtained using GNU GCC 4.8.4. An Intel Xeon E5-1620 CPU with both hyper-threading and TurboBoost disabled has been used, wherein the eight CPUs are placed in on-demand mode at 3.70 GHz. GNU GMP 6.1.2 has been used to provide multiple precision arithmetic. Results are given in Table 5.6.

As expected, the key generation process is the most time consuming component of the scheme, and this increases as security and therefore lattice dimension increase, for example 177s for Set 1, and 1040s for Set 2. The trend down the hierarchical levels is that the delegate, extract, encrypt and decrypt all become more time consuming as hierarchical level increases. For extract in LATTE-1, this corresponds to a 97% decrease in op/s from level 0 to level 3. There is an average performance slowdown in the delegate/extract function of Y% from LATTE-1 to LATTE-2. Performance at LATTE-0, hierarchical level Z is comparable to LATTE-2 at level Y. Constrained devices used in IoT may not be able to accommodate for the extended hierarchical levels.

The substantial decrease in performance from LATTE-1 to LATTE-2 is due to the parameters exceeding the modular_prime look-up table size which is built into *libsafecrypto*. This means that during the polynomial GCD computation in both key generation and delegation, the modular primes have to be computed rather than simply retrieved from memory. For this reason, results were not obtained for LATTE-2, level 3.

Pairings-based HIBE scheme timing results from [Luzhnica, 2011] are given in 5.7. Parameter s indicates the security, which is comparable to the bit-length of RSA modulus by providing the same security. Although not directly comparable, these results give a good indication of the feasibility of LATTE at levels 0 or 1. However, at

Table 5.7 Timings in ms for the Gentry-Silverberg HIBE scheme (2-level) on Intel Core i5 4GB Mac OS X 10.6.7 using Java v1.6 [Luzhnica, 2011]

s	**length of m**	**Enc**	**Dec**
1024	160	91.5	156
2048	224	463.5	754
3072	256	1352.5	2278

levels 2 and 3, the performance may not be suitable for constrained devices with low processing power.

5.4 Skinny Latte

In this section, a novel variant of Latte with fixed user public key and ciphertext size is proposed. This uses the techniques of Agrawal et al. [2010c] (henceforth referred to as the ABB-scheme) with the NTRU lattice structure, as well as other algorithmic optimisations, namely more compact sampling and NTRU polynomial generation methods.

5.4.1 Skinny Latte description

The Skinny Latte key generation follows the same concept as Latte, but with a more efficient NTRU-Solve algorithm as introduced in Prest et al. [2017], and is given in Algorithm 24. On input of the system parameters N, q and σ_0, an NTRU-type lattice is generated and its public and secret polynomials are output. It also samples a polynomial $\mathbf{b}$ from a uniformly random distribution over $\mathcal{R}_q$.

The Skinny Latte delegation algorithm applies the fixed-dimension technique to the structured lattices of Latte. Recall from §5.2 that the fixed-dimension scheme in Agrawal et al. [2010c] takes the basis matrix at level $l-1$ and multiplies it by the output

of the hash function of the identity belonging to the user at level l. This generates the public user lattice basis of level l, and the private key is a short basis for this lattice. However, in order to maintain the NTRU-type structure of the keys in Skinny Latte, the user key is required to be generated as in Latte, by sampling short vectors in the lattice which satisfy the NTRU equations. Whereas the method in Agrawal et al. [2010c] considers a *uniform matrix* as the master public key, this is replaced with $\begin{pmatrix} \mathcal{A}(\mathbf{h}) & \mathbf{I}_N \\ q\mathbf{I}_N & \mathbf{0}_N \end{pmatrix}$, where $\mathbf{h}$ is a *uniform ring polynomial*, to maintain the cyclic structure of the lattices. The output of the hash function $\mathbf{R}_l = H(ID_1)H(ID_2)...H(ID_l)$ will also be a cyclic-structured matrix with low norm, as opposed to a uniform matrix as in the ABB-scheme. This means the new public key of the user at level l is of the form: $\mathbf{B}_l = \begin{pmatrix} \mathcal{A}(\mathbf{h}) & \mathbf{I}_N \\ q\mathbf{I}_N & \mathbf{0}_N \end{pmatrix} \cdot \mathbf{R}_l^{-1}$ with associated short basis $\mathbf{S}'_l := \begin{pmatrix} \mathcal{A}(\mathbf{g}) & -\mathcal{A}(\mathbf{f}) \\ \mathcal{A}(G) & -\mathcal{A}(F) \end{pmatrix} \begin{pmatrix} \mathbf{R}_{0,0} & \mathbf{R}_{0,1} \\ \mathbf{R}_{1,0} & \mathbf{R}_{1,1} \end{pmatrix}$. However, the user at level l cannot learn $\mathbf{S}'_l$ or else the security of the levels above and other level-l users would be compromised. Therefore a slightly lower-quality secret basis $\mathbf{S_l} = \begin{pmatrix} \mathcal{A}(\mathbf{s}_{0,0}) & \mathcal{A}(\mathbf{s}_{0,1}) \\ \mathcal{A}(\mathbf{s}_{1,0}) & \mathcal{A}(\mathbf{s}_{1,1}) \end{pmatrix}$ must be generated, i.e. one with a larger GS-norm. To do this, a short vector $(\mathbf{s}_{0,0}, \mathbf{s}_{0,1})$ in the lattice Λ_l at level l is generated using the lattice sampler and the known short basis $\mathbf{S}'_l$. From there, steps 4-9 of the master key generation are reproduced for $(\mathbf{s}_{0,0}, \mathbf{s}_{0,1})$ to create the secret NTRU-type basis $\mathbf{S}_l$. The intuition behind this follows that of §2.4.2. The first level delegation is shown in Figure 5.2 and the components in Table 5.8.

The user key extraction, Algorithm 29, requires finding a short solution to

$$\begin{bmatrix} \mathcal{A}(\mathbf{h}) & \mathbf{I}_N \\ q\mathbf{I}_N & \mathbf{O}_N \end{bmatrix} \cdot \mathbf{R}_l^{-1} \cdot \begin{bmatrix} \mathbf{t}_0 \\ \mathbf{t}_1 \end{bmatrix} = \mathbf{b}.$$

Table 5.8 Summary of Skinny Latte components

Name of component	Notation
Public master key $(l = 0)$	$\mathbf{B}_0 = \begin{bmatrix} \mathcal{A}(\mathbf{h}) & \mathbf{I}_N \\ q\mathbf{I}_N & \mathbf{O}_N \end{bmatrix}$
Private master key $(l = 0)$	$\mathbf{S}_0 = \begin{bmatrix} \mathbf{g} & -\mathbf{f} \\ G & -F \end{bmatrix}$ such that $\mathbf{h} = \mathbf{g} \cdot \mathbf{f}^{-1}$
Public basis $(l = 1)$	$\mathbf{B}_1 = \begin{bmatrix} \mathcal{A}(\mathbf{h}) & \mathbf{I}_N \\ q\mathbf{I}_N & \mathbf{O}_N \end{bmatrix} \cdot \mathbf{R}_1^{-1}$, $\mathbf{R}_1 = H(ID_1)$
Private basis before randomisation $(l = 1)$	$\mathbf{S}'_1 = \begin{bmatrix} \mathbf{g} & -\mathbf{f} \\ G & -F \end{bmatrix} \cdot \mathbf{R}_1^{-1}$
Private delegated user basis $(l = 1)$	$\mathbf{S}_1 = \begin{bmatrix} \mathbf{s}_{0,0} & \mathbf{s}_{0,1} \\ \mathbf{s}_{1,0} & \mathbf{s}_{1,1} \end{bmatrix}$
Public basis $(l = 2)$	$\mathbf{B}_2 = \begin{bmatrix} \mathcal{A}(\mathbf{h}) & \mathbf{I}_N \\ q\mathbf{I}_N & \mathbf{O}_N \end{bmatrix} \cdot \mathbf{R}_2^{-1}$, $\mathbf{R}_2 = H(ID_1)H(ID_2)$
Private basis before randomisation $(l = 2)$	$\mathbf{S}'_2 = \begin{bmatrix} \mathbf{g} & -\mathbf{f} \\ G & -F \end{bmatrix} \cdot \mathbf{R}_2^{-1}$
Private delegated user basis $(l = 2)$	$\mathbf{S}_2 = \begin{bmatrix} \mathbf{s}_{0,0}^{(2)} & \mathbf{s}_{0,1}^{(2)} \\ \mathbf{s}_{1,0}^{(2)} & \mathbf{s}_{1,1}^{(2)} \end{bmatrix}$

To solve this, the FFT Sampler is called to find a short lattice vector in Λ_l using the preliminary short basis computed by the parent, $\mathbf{S}'_l$ and centre vector $\mathbf{c} = (\mathbf{b}, 0, ..., 0)$ and target standard deviation σ_l to obtain a lattice vector $(\mathbf{t}'_0, \mathbf{t}'_1)$. The KMS sets $(\mathbf{t}_0, \mathbf{t}_1) = (\mathbf{t}'_0, \mathbf{t}'_1)$ and returns this as the user private key.

The user key encryption and decryption are similar to Latte (traditional R-LWE based) although the lattice dimension is fixed for all levels. These are given in Algorithms 30 and 31 respectively. Encrypt computes the ciphertext $(\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2)$ as

$$\begin{bmatrix} \mathbf{c}_0 \\ \mathbf{c}_1 \end{bmatrix} = \begin{bmatrix} \mathcal{A}(\mathbf{h}) & \mathbf{I}_N \\ q\mathbf{I}_N & \mathbf{O}_N \end{bmatrix} \cdot \mathbf{R}_l^{-1} \cdot \begin{bmatrix} \mathbf{e} \\ \mathbf{e}^* \end{bmatrix} + \begin{bmatrix} \mathbf{e}_0 \\ \mathbf{e}_1 \end{bmatrix}$$

$$\mathbf{c}_2 = b \cdot \mathbf{e} + \mathbf{e}_2 + m$$

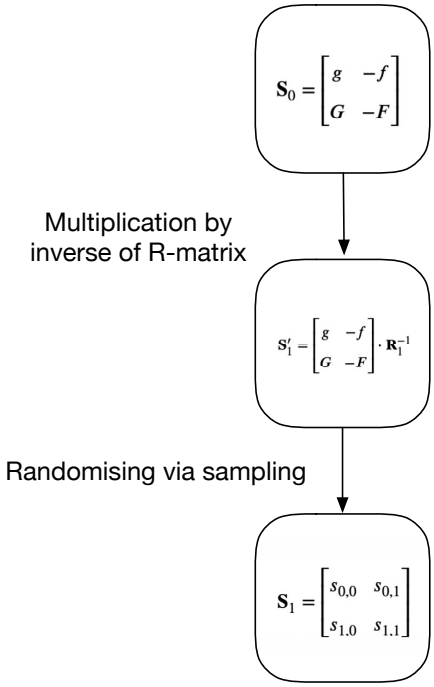


Fig. 5.2 Basis changes during Skinny Latte delegation

Decryption makes use of the user private key, $(\mathbf{t}_0, \mathbf{t}_1)$. Upon receiving the ciphertext $(Z, \mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2)$, the receiver can use their private key to compute:

$$\begin{aligned} V &= \mathbf{c}_2 - \mathbf{c}_0 \cdot \mathbf{t}_0 - \mathbf{c}_1 \cdot \mathbf{t}_1 \\ &= (q-1)/2 \cdot \mathbf{m} + \mathbf{e}_2 + (\mathbf{b} - \mathbf{h} \cdot \mathbf{t}_0 - A_1 \cdot \mathbf{t}_1) - \mathbf{e}_0 - \mathbf{e}_2 \\ &= (q-1)/2 \cdot \mathbf{m} + \mathbf{e}_2 - \mathbf{e}_0 \cdot \mathbf{t}_0 - \mathbf{e}_1 \cdot \mathbf{t}_1 \end{aligned}$$

This can then be lifted to a polynomial in $\mathcal{R}$ and be decoded, as in Latte.

Skinny Latte parameters are derived by the practical security notions given in Section 5.4.3.

5.4.2 Latte to Skinny Latte

The main constraint of Latte is its inability to scale to multiple hierarchical levels. Skinny Latte aims to address this challenge through the following features.

NTRU polynomial generation

The signature scheme Falcon [Prest et al., 2017] uses a novel algorithm NTRU-Solve in its key generation process to compute the NTRU polynomials F, G from sampled $\mathbf{f}, \mathbf{g}$, as described in §2.4.2. This exploits the tower of rings structure in order to deal with larger coefficients more efficiently. Skinny Latte utilises this algorithm both in the key generation and key delegation components. The original algorithm is given as Algorithm 25. The input consists of two sampled ring polynomials $\mathbf{f}, \mathbf{g}$ and outputs F, G such that the NTRU equation is satisfied. The field norm is applied repeatedly to map the polynomials into a subring, where computations can be performed more efficiently. This is due to the equivalence of computations over polynomials $\mathbf{f}(x), \mathbf{g}(x) \in \mathcal{R} = \mathbb{Z}_q[x] \big/ \langle x^{N/2}+1 \rangle$ and over polynomials $\mathbf{f}(x^2), \mathbf{g}(x^2) \in \mathcal{R} = \mathbb{Z}_q[x] \big/ \langle x^N + 1 \rangle$. The *Reduce* algorithm is called to reduce the polynomials before they are returned and this is stated as Algorithm 26.

Fixed dimension technique of Agrawal et al. [2010c]

As described in Section 5.2, Agrawal et al. proposed a method to keep the lattice dimension fixed during key delegation. Rather than using the user ID to *extend* the master lattice, they used it to create a lattice of the *same dimension*, with a slightly lower quality user private basis, i.e. one with a larger Gram-Schmidt norm. The lattices in Latte incorporate a NTRU-like structure, so this needs to be maintained

throughout the delegation. Hence a *structured* matrix $\mathbf{R}_l$ is generated based on the hash of the user ID at level l, and the inverse of this is multiplied by the level 0 public key in order to produce that of the level l lattice. The hash function $Hash$ is of the form:

$$Hash(ID) = \begin{pmatrix} H(ID|00) & H(ID|01) \\ H(ID|10) & H(ID|11) \end{pmatrix}$$

where H is as defined in §2.5.1. The corresponding $\mathbf{f}', \mathbf{g}', F'$ and G' are sampled and computed using an analogous process to the NTRU generation process. This adjustment implies smaller user public keys and therefore smaller ciphertext sizes than Latte.

Fast Fourier Sampling

The GPV Sampler (§2.5.5) is replaced in both key delegation and user key extraction with the FFT-domain sampler used in Falcon [Prest et al., 2017]. Again, this exploits the tower of rings structure and uses a recursive call to sample polynomials in the two-dimensional ring and then reconstructs it to create the lattice sample in the required ring. Operation in FFT representation further increases efficiency. The original algorithm is given as Algorithm 32. The tree T is generated from the private basis $\mathbf{S}'_l$ using the same process as given for Falcon, given in Algorithm 27.

Key and Ciphertext Sizes

For Skinny Latte, the master public key sizes are the same as Latte but the user public key and ciphertext stays constant across all levels (Table 5.9), allowing for scalability to a deeper hierarchy.

Algorithm 24: The Skinny Latte key generation algorithm

Data: N, q, σ_0
Result: $\mathbf{S_0} \in \mathcal{R}_q^{2\times 2}$, $\{\mathbf{h}, \mathbf{b}\} \in \mathcal{R}_q^2$,

1 $\mathbf{f}, \mathbf{g} \leftarrow \mathcal{D}_{N,\sigma_0}$
2 $Norm \leftarrow max\Big(||\mathbf{g}, -\mathbf{f}||, \Big|\Big|\Big(\frac{q\bar{\mathbf{f}}}{\mathbf{f}\cdot\bar{\mathbf{f}}+\mathbf{g}\cdot\bar{\mathbf{g}}}, \frac{q\bar{\mathbf{g}}}{\mathbf{f}\cdot\bar{\mathbf{f}}+g\cdot\bar{\mathbf{g}}}\Big)\Big|\Big|\Big)$
3 **if** $Norm > \sigma_0 \cdot \sqrt{2N}$ **then** go to Step 1;
4 $F, G \leftarrow NTRUSolve(\mathbf{f}, \mathbf{g})$
5 $\mathbf{S_0} = \begin{pmatrix} \mathcal{A}(\mathbf{g}) & -\mathcal{A}(\mathbf{f}) \\ \mathcal{A}(G) & -\mathcal{A}(F) \end{pmatrix}$
6 $b \leftarrow \mathcal{U}(\mathcal{R}_q)$
7 **return** $(\mathbf{h} = \mathbf{g} \cdot \mathbf{f}^{-1} mod\, q,)$
8 and $\mathbf{S_0}$

Algorithm 25: NTRUSolve ([Prest et al., 2017])

Data: $\mathbf{f}, \mathbf{g}$
Result: F, G such that $\mathbf{f}G - \mathbf{g}F = q \mod x^N + 1$

1 **if** $N = 1$ **then**
2 Compute $u, v \in \mathbb{Z}$ such that $u\mathbf{f} - v\mathbf{g} = GCD(\mathbf{f}, \mathbf{g})$ **if** $GCD(\mathbf{f}, \mathbf{g}) \neq 1$ **then**
3 abort;
4 $(F, G) \leftarrow (vq, uq)$
5 **return** (F, G)
6 **else**
7 $\mathbf{f}' \leftarrow N(\mathbf{f})$
8 $\mathbf{g}' \leftarrow N(\mathbf{g})$
9 $(F', G') \leftarrow NTRUSolve_{N/2,q}(\mathbf{f}', \mathbf{g}')$
10 $F \leftarrow F'(x^2)\mathbf{f}'(x^2)/\mathbf{f}(x)$
11 $G \leftarrow G'(x^2)\mathbf{g}'(x^2)/\mathbf{g}(x)$
12 $Reduce(\mathbf{f}, \mathbf{g}, F, G)$
13 **return** (F, G)

5.4.3 Security Analysis and Parameter Derivation

With regards to provable security, Skinny Latte can be shown to be secure in the ROM (as introduced in §2.2.2) and this is achieved in this section. Additionally, the

Algorithm 26: Reduce (from FALCON)

Data: $\mathbf{f}, \mathbf{g}, F, G$
Result: F, G reduced w.r.t.$\mathbf{f}, \mathbf{g}$
1 **while** $k \neq 0$ **do**
2 $\quad k = \left\lfloor \frac{F \cdot \bar{\mathbf{f}} + G \cdot \bar{\mathbf{g}}}{f \cdot \bar{\mathbf{f}} + \mathbf{g} \cdot \bar{\mathbf{g}}} \right\rceil \in \mathcal{R}$
3 $\quad F \leftarrow F - k \cdot \mathbf{f}$ and $G \leftarrow G - k \cdot \mathbf{g}$
4 **return** (F, G)

Algorithm 27: The Tree computation algorithm $Tree()$

Data: $\mathbf{S}'_l, \sigma_l$
Result: $\hat{\mathbf{S}}'_{\mathbf{l}} \in \mathcal{R}_q^{2\times 2}$, tree T_l,
1 $\hat{\mathbf{S}}'_{\mathbf{l}} = FFT(\mathbf{S}'_l)$
2 $\mathbf{G_l} \leftarrow \hat{\mathbf{S}}'_{\mathbf{l}} \times \mathbf{S}'_{\mathbf{l}}$
3 $T = ffLDL^*(\mathbf{G_l})$
4 for each leaf of T_l, leaf.value $= \sigma_l / \sqrt{\text{leaf.value}}$
5 **return** $(\hat{\mathbf{S}}'_{\mathbf{l}}, T_l)$

Algorithm 28: The Skinny Latte key delegation algorithm

Data: $N, q, \sigma_l, \mathbf{S}_0, H : \{0,1\}^* \to \mathbb{Z}_q^N, ID_l$
Result: $\mathbf{B}_l, \mathbf{S}_l \in \mathcal{R}_q^{2\times 2}$
1 Compute $R_l = Hash(ID_1)Hash(ID_2)...Hash(ID_l)$ and set $\mathbf{B}_l = B_0 \cdot R_l^{-1}$
2 $\mathbf{S}'_l = \mathbf{S}_0 \cdot R_l^{-1}$
3 $(\hat{\mathbf{S}}'_{\mathbf{l}}, T_l) \leftarrow Tree(\mathbf{S}'_l, \sigma_l)$
4 $(\mathbf{s}_{0,0}, \mathbf{s}_{0,1}) \leftarrow FFTSampler((0,0), T_l)$
5 *Norm of* $\mathbf{S}_l = ||\mathbf{S}_{\mathbf{s}_{0,0},\mathbf{s}_{0,1}}||_{GS}$
6 **if** *Norm of* $\mathbf{S}_l > GSB_l$ **then** go to Step 4;
7 $(\mathbf{s}_{1,0}, \mathbf{s}_{1,1}) \leftarrow NTRUSolve(\mathbf{s}_{0,0}, \mathbf{s}_{0,1})$
8 $\mathbf{S_l} = \begin{pmatrix} \mathbf{s}_{0,0} & \mathbf{s}_{0,1} \\ \mathbf{s}_{1,0} & \mathbf{s}_{1,1} \end{pmatrix}$
9 **return** $\mathbf{B}_l$ and $\mathbf{S_l}$

basis quality degradation through delegation is investigated in order to give practical security estimates and hence derive the parameters in Table 5.15.

Algorithm 29: The Skinny Latte user key extraction algorithm

Data: $N, q, \sigma_l, \mathbf{S}_0, H : \{0,1\}^* \to \mathbb{Z}_q^N, ID_l$
Result: $\mathbf{SL}_l = \{\mathbf{t}_0, \mathbf{t}_1\} \in \mathcal{R}$

1 Compute $R_l = Hash(ID_1)Hash(ID_2)...Hash(ID_l)$ and $\mathbf{S}'_l = \mathbf{S}_0 \cdot R_l^{-1}$
2 $\mathbf{t} = \mathbf{b}$
3 $(\hat{\mathbf{S}}'_{\mathbf{l}}, T_l) \leftarrow Tree(\mathbf{S}'_l, \sigma_l)$
4 $(\mathbf{t_0}, \mathbf{t_1}) \leftarrow (\mathbf{t}, 0) - FFTSampler((\mathbf{t}, 0), T_l)$
5 **return** $\mathbf{SK}_l = \{\mathbf{t_0}, \mathbf{t_1}\} \in \mathcal{R}_q^2$

Algorithm 30: The Skinny Latte encryption algorithm

Data: $N, q, \alpha_l, B, KDF, ID_l, \mu \in \{0,1\}^{256}$
Result: $(Z, \mathbf{c_0}, \mathbf{c_1}, \mathbf{c_2})$,where $Z \in \{0,1\}^{256}, \mathbf{c}_i \in \mathcal{R}_q$

1 Compute $R_l = Hash(ID_1)Hash(ID_2)...Hash(ID_l)$
2 $seed \leftarrow \{0,1\}^{256}$
3 $Z = \mu \oplus KDF(seed)$
4 $\mathbf{e}, \mathbf{e}^*, \mathbf{e_0}, \mathbf{e_1}, \mathbf{e_2} \leftarrow \mathcal{D}_{N,\alpha_l}$
5 $\begin{bmatrix} \mathbf{c}_0 \\ \mathbf{c}_1 \end{bmatrix} = \begin{bmatrix} \mathcal{A}(\mathbf{h}) & \mathbf{I}_N \\ q\mathbf{I}_N & \mathbf{O}_N \end{bmatrix} \cdot \mathbf{R}_l^{-1} \cdot \begin{bmatrix} \mathbf{e} \\ \mathbf{e}^* \end{bmatrix} + \begin{bmatrix} \mathbf{e}_0 \\ \mathbf{e}_1 \end{bmatrix}$
6 $\mathbf{m} \leftarrow Encode(seed)$
7 $\mathbf{c}_2 = \mathbf{b} \cdot \mathbf{e} + \mathbf{e}_2 + \mathbf{m}$
8 Drop the least significant bits of $\mathbf{c}_2 : \mathbf{c}_2 \leftarrow q\lfloor \mathbf{c}_2/q \rceil \mod q$
9 **return** $(Z, \mathbf{c_0}, \mathbf{c_1}, \mathbf{c_2})$

Practical Security

The first step to assessing the practical security of Skinny Latte is to clarify the degradation in basis quality which occurs in delegation. The quality of a basis can be measured by its GS-norm. The Skinny Latte basis delegation process follows that of the ABB-scheme but with NTRU-type lattice bases, given as Algorithm 28. Concisely, it follows these steps:

Algorithm 31: The Skinny Latte decryption algorithm

Data: $N, q, (Z, \mathbf{c_0}, \mathbf{c_1}, \mathbf{c_2}), ID_l, \mathbf{SK}_l = \{\mathbf{t}_0, \mathbf{t}_1\}$
Result: μ'

Decompress $\mathbf{c}_2 : \mathbf{c}_2 \leftarrow \lfloor q\mathbf{c}_2/q \rceil \mod q$
Compute $V = \mathbf{c}_2 - \mathbf{c}_0 \cdot \mathbf{t}_0 - \mathbf{c}_1 \cdot \mathbf{t}_1$, lift to a polynomial in $\mathcal{R}$
$seed' \leftarrow Decode(V)$
$\mathbf{e}', \mathbf{e}^{*\prime}\mathbf{e}'_0, \mathbf{e}'_1 \leftarrow \mathcal{D}_{N,\alpha_l}$
$\begin{bmatrix} \mathbf{c}'_0 \\ \mathbf{c}'_1 \end{bmatrix} = \begin{bmatrix} \mathcal{A}(\mathbf{h}) & \mathbf{I}_N \\ q\mathbf{I}_N & \mathbf{O}_N \end{bmatrix} \cdot \mathbf{R}_l^{-1} \cdot \begin{bmatrix} \mathbf{e}' \\ \mathbf{e}^{*\prime} \end{bmatrix} + \begin{bmatrix} \mathbf{e}'_0 \\ \mathbf{e}'_1 \end{bmatrix}$
$\mathbf{m} \leftarrow Encode(seed)$
$\mathbf{c}'_2 = \mathbf{b} \cdot \mathbf{e}' + \mathbf{e}'_2 + \mathbf{m}$
Drop the least significant bits of $\mathbf{c}'_1 : \mathbf{c}'_1 \leftarrow q\lfloor \mathbf{c}_1/q \rceil \mod q$
Check $(\mathbf{c}'_0, \mathbf{c}'_1, \mathbf{c}'_2)$ agree with $(\mathbf{c_0}, \mathbf{c_1}, \mathbf{c_2})$
return $\mu' = Z \oplus KDF(seed')$

Algorithm 32: The FFT Sampler is the recursive form of the GPV Sampler, and operates in the FFT domain for maximum efficiency (from FALCON)

Data: $\mathbf{t} = (\mathbf{t_0}, \mathbf{t_1})$ in FFT format , tree T
Result: $\mathbf{z} = (\mathbf{z_0}, \mathbf{z_1})$ in FFT format

if $n = 1$ **then**
  $\sigma' \leftarrow$ T.value
  $z_0 \leftarrow \mathcal{D}_{\mathbb{Z},t_0,\sigma'}$
  $z_1 \leftarrow \mathcal{D}_{\mathbb{Z},t_1,\sigma'}$
  return $\mathbf{z} = (z_0, z_1)$
else
  $(l, T_0, T_1) \leftarrow$ (T.value, T.leftchild, T.rightchild)
  $\mathbf{t_1} \leftarrow$ splitfft$(\mathbf{t}_1)$
  $\mathbf{z_1} \leftarrow$ ffSampling$(\mathbf{t_1}, T_1)$
  $\mathbf{z}_1 \leftarrow$ mergefft$(\mathbf{z_1})$
  $\mathbf{t}'_0 \leftarrow \mathbf{t}_0 + (\mathbf{t_1} - \mathbf{z_1}) \cdot l$
  $\mathbf{t_0} \leftarrow$ splitfft$(\mathbf{t}'_0)$
  $\mathbf{z_0} \leftarrow$ ffSampling$(\mathbf{t_0}, T_0)$
  $\mathbf{z}_0 \leftarrow$ mergefft$(\mathbf{z_0})$
  return $\mathbf{z} = (\mathbf{z_0}, \mathbf{z_1})$

At level 0:

$$\text{Public key} = \begin{bmatrix} \mathcal{A}(\mathbf{h}) & \mathbf{I}_N \\ q\mathbf{I}_N & \mathbf{O}_N \end{bmatrix}$$

Table 5.9 Skinny Latte Public Key and Ciphertext Sizes bytes

Parameter Set	Master Public Key	Ciphertext
SK-LATTE-1	6144	9248
SK-LATTE-2	12800	19232
SK-LATTE-3	9216	13856
SK-LATTE-4	19456	29216

which can be implicitly be defined by h, and

$$\text{Private key} = \begin{bmatrix} g & -f \\ G & -F \end{bmatrix}$$

which can be implicitly be defined by $\mathbf{f}$ and $\mathbf{g}$.

To delegate to a user at level l, the KMS computes

$$\mathbf{R}_l = Hash(ID_1)Hash(ID_2)...Hash(ID_l)$$

where the function $Hash$ is of the form:

$$Hash(ID) = \begin{pmatrix} H(ID|00) & H(ID|01) \\ H(ID|10) & H(ID|11) \end{pmatrix}$$

$\mathbf{R}_l$ is denoted as

$$\mathbf{R}_l = \begin{bmatrix} R_{00} & R_{01} \\ R_{10} & R_{11} \end{bmatrix}$$

and therefore

$$\mathbf{R}_l^{-1} = (1/det) \begin{bmatrix} R_{11} & -R_{01} \\ -R_{10} & R_{00} \end{bmatrix}$$

This can be scaled by det and the public key at level l set as

$$\mathbf{B}_l = \begin{bmatrix} \mathcal{A}(\mathbf{h}) & \mathbf{I}_N \\ q\mathbf{I}_N & \mathbf{O}_N \end{bmatrix} \cdot \begin{bmatrix} R_{11} & -R_{01} \\ -R_{10} & R_{00} \end{bmatrix}$$

and the private key (before randomisation),

$$\mathbf{S}'_l = \begin{bmatrix} \mathbf{g} & -\mathbf{f} \\ G & -F \end{bmatrix} \cdot \begin{bmatrix} R_{11} & -R_{01} \\ -R_{10} & R_{00} \end{bmatrix}$$

Then, a short vector is sampled from this lattice using the private basis as an input to the FFT Sampler. NTRU-Solve is performed to compute a randomised NTRU-type basis as the level-l user private key, $\mathbf{S}_l$.

$$\mathbf{S}_l = \begin{bmatrix} \mathbf{s}_{0,0} & -\mathbf{s}_{0,1} \\ \mathbf{s}_{1,0} & -\mathbf{s}_{1,1} \end{bmatrix}$$

Next is proposed an analogous version of Theorem 3 from Section 4 of Agrawal et al. [2010c], for Skinny Latte lattices, which bounds the degradation in the quality of the basis during delegation.

Theorem 1 shows that Skinny Latte basis delegation produces a random basis of Λ_l whose Gram-Schmidt norm is bounded as a function of $||\mathbf{S}_{l-1}||_{GS}$.

Theorem 1 (analogous to Theorem 3 of Agrawal et al. [2010c]). *Suppose a sub-KMS with identity ID_{l-1} is delegating from level $(l-1)$ to level l. Polynomial $\mathbf{h}$ corresponding to the public basis $\mathbf{B}_0$, polynomials $\{\mathbf{f}, \mathbf{g}\}$ corresponding to private basis $\mathbf{S}_0$ and matrix $\mathbf{R}_{l-1}$ is known to the sub-KMS. It takes as input $Hash(ID_l)$, computes matrix $\mathbf{R}_l$ and*

outputs the polynomials $\{\mathbf{s}_{0,0}, \mathbf{s}_{0,1}\}$ *corresponding to private basis* $\mathbf{S}_l$.

Suppose $\mathbf{R}$ *is sampled from* $\mathcal{D}_{2N\times 2N,\sigma_{\mathbf{R}}}$ *and* $\{\mathbf{s}_{0,0}, \mathbf{s}_{0,1}\}$ *are outputs of the FFT Sampler with the GS-norm Bound (GSB) in Key Delegation satisfying*

$$GSB_l = ||\mathbf{S}_{l-1}||_{GS} \cdot \sigma_{\mathbf{R}} \cdot (2N)^{3/2} \cdot \eta_\epsilon^3(\mathbb{Z}^{2N}).$$

Then the random variable yielding the private basis polynomials $\{\mathbf{s}_{0,0}, \mathbf{s}_{0,1}\}$ *is statistically close to the random variable* $FFTSampler(\mathbf{S}, \mathbf{t}, \sigma)$*, where* $\mathbf{S}$ *is an arbitrary basis of* Λ_l *corresponding to the public basis* $\mathbf{B}_l$ *satisfying* $||\mathbf{S}||_{GS} < \frac{GSB_l}{\eta_\epsilon(\mathbb{Z}^{2N})}$*. When considering* $\mathbf{R}_l$ *as a product of* $l \geq 1$ *matrices sampled from* $\mathcal{D}_{2N\times 2N}$*, then the* GSB_l *degrades to*

$$GSB_l \geq ||\mathbf{S}_0||_{GS} \cdot \left(\sigma_{\mathbf{R}} \cdot \sqrt{2N} \cdot \eta_\epsilon^2(\mathbb{Z}^{2N})\right)^l \cdot \eta_\epsilon(\mathbb{Z}^{2N}).$$

From [Ducas et al., 2014a], if

$$\sigma_l = ||\mathbf{S}'_l||_{GS} \cdot \eta_\epsilon(\mathbb{Z}^{2N}),$$

it can be deduced that

$$||\mathbf{S}_l||_{GS} \leq ||\mathbf{S}'_l||_{GS} \cdot \eta_\epsilon(\mathbb{Z}^{2N})$$

and hence that

$$||\mathbf{S}_l||_{GS}/||\mathbf{S}_{l-1}||_{GS} \leq \left(\sigma_{\mathbf{R}} \cdot \sqrt{2N} \cdot \eta_\epsilon^2(\mathbb{Z}^{2N})\right) \cdot \eta_\epsilon(\mathbb{Z}^{2N}).$$

When considering $\mathbf{R}$ as a product of l matrices, then the smallest possible GSB_l in Theorem 1 is

$$GSB_l = ||\mathbf{S}_0||_{GS} \cdot \left(\sigma_{\mathbf{R}} \cdot \sqrt{2N} \cdot \eta_\epsilon^2(\mathbb{Z}^{2N})\right)^l \cdot \eta_\epsilon(\mathbb{Z}^{2N})$$

This quantity is the minimum degradation in basis quality as one delegates down l levels of the hierarchy.

The following lemmas are used in the proof of Theorem 1.

Lemma 1 (Concrete version of Lemma 38 from Agrawal et al. [2010c])**.** *Let* $\mathbf{u} \in \mathbb{R}^{2N}$. *Let* $\mathbf{r} \in \mathbb{Z}^{2N}$ *be a random variable sampled from* $\mathcal{D}_{\mathbb{Z}^{2N},\sigma}$. *Then the inner product* $\langle \mathbf{r}^T \cdot \mathbf{u} \rangle$ *satisfies* $|\langle \mathbf{r}^T \mathbf{u} \rangle| \leq ||\mathbf{u}|| \cdot \sigma \cdot \eta_\epsilon^2(\mathbb{Z}^{2N})$ *with overwhelming probability.*

Proof. To sample $\mathbf{r} = (\mathbf{r_1}, ..., \mathbf{r}_{2N})$ from $\mathcal{D}_{\mathbb{Z}^{2N},\sigma}$, each $\mathbf{r}_i$ can be considered as an independent sample from $\mathcal{D}_{\mathbb{Z},\sigma}$. By the properties of the smoothing parameter of the lattice $\mathbb{Z}^{2N}$, and the Lemma detailed in [Agrawal et al., 2010c], it can be stated that

$$Pr\left[|\mathbf{r}_i| > \sigma \cdot \eta_\epsilon(\mathbb{Z}^{2N})\right] = 2e^{-\pi\eta_\epsilon^2(\mathbb{Z}^{2N})}.$$

Now, consider the inner product $\langle \mathbf{r}^T \cdot \mathbf{u} \rangle = \sum_{i=1}^{2N} \mathbf{r}_i \mathbf{u}_i$. This can be modelled as independent, bounded samples, with expectation 0 and bounded by $|r_i \cdot u_i| \leq |u_i| \cdot \sigma \cdot \eta_\epsilon(\mathbb{Z}^{2N})$ and therefore by the Hoeffding bound [Hoeffding, 1963] it can be claimed

$$Pr\left[|r_i| > \eta_\epsilon(\mathbb{Z}^{2N}) \cdot ||\mathbf{u}|| \cdot \sigma \cdot \eta_\epsilon(\mathbb{Z}^{2N})\right] = 2e^{-\eta_\epsilon^2(\mathbb{Z}^{2N})/2} + \epsilon'(2N)$$

where $\epsilon'(2N)$ is negligible. And so the Lemma follows. □

Lemma 2. *Let* $\mathbf{R} \in \mathbb{Z}^{2N \times 2N}$ *be distributed as* $(\mathcal{D}_{\mathbb{Z}^{2N},\sigma_{\mathbf{R}}})^{2N}$ *and let* $\mathbf{u} \in \mathbb{R}^{2N}$*. Then, with overwhelming probability,* $||\mathbf{Ru}|| \leq ||\mathbf{u}|| \cdot \sigma_{\mathbf{R}} \cdot \sqrt{2N} \cdot \eta_{\epsilon}^{2}(\mathbb{Z}^{2N})$.

Proof. The rows of $\mathbf{R}$ have a norm of $\leq \sigma_{\mathbf{R}}\sqrt{2N}$ with overwhelming probability, due to Lemma 4.4 of Micciancio and Regev [2004]. By the Cauchy-Schwartz inequality, each coordinate of $\mathbf{Ru}$ is at most $||\mathbf{u}|| \cdot \sigma_{\mathbf{R}}\sqrt{2N}$ in absolute value, from which it follows that $||\mathbf{Ru}|| \leq ||\mathbf{u}|| \cdot \sigma_{\mathbf{R}} \cdot 2N$.

Furthermore, considering the randomness of the rows of $\mathbf{R}$ and by Lemma 1, it can be stated that all coordinates of $\mathbf{Ru}$ are $\leq ||\mathbf{u}|| \cdot \sigma_{\mathbf{R}} \cdot \eta_{\epsilon}^{2}(\mathbb{Z})$ in absolute value. Hence, $||\mathbf{Ru}|| \leq ||\mathbf{u}|| \cdot \sigma_{\mathbf{R}} \cdot \sqrt{2N} \cdot \eta_{\epsilon}^{2}(\mathbb{Z}^{2N})$, as required. □

Lemma 3. *Let* $\mathbf{R}$ *be a matrix in* $\mathbb{R}^{2N \times 2N}$ *and* $\mathbf{S} = \{\mathbf{s_1}, \mathbf{s_2}, ..., \mathbf{s}_{2N}\}$ *be a linearly independent set. Let* $\mathbf{S_R} = \{\mathbf{Rs_1}, \mathbf{Rs_2}, ..., \mathbf{Rs}_{2N}\}$*. Then,* $||\mathbf{S_R}||_{GS} \leq \max_{1 \leq i \leq 2N} ||\mathbf{R}\tilde{\mathbf{s}}_i||$.

Proof. [Agrawal et al., 2010c]

It is shown that for all $i = 1, ..., 2N$, the norm of the ith element of the Gram-Schmidt orthogonalisation of $\mathbf{S_R}$ is less than $||\mathbf{R}\tilde{\mathbf{s}}_i||$. This will prove the lemma.

For $i \in \{1, ..., 2N\}$, let $\mathbf{V} := Span(\mathbf{Rs}_1, ..., \mathbf{Rs}_{i-1})$. Set $\mathbf{v} := \mathbf{s}_i - \tilde{\mathbf{s}}_i$. Then, $\mathbf{v} \in Span(\mathbf{s}_1, ..., \mathbf{s}_{i-1})$ and therefore $\mathbf{Rv} \in \mathbf{V}$. Let u be the projection of $\mathbf{R}\tilde{\mathbf{s}}_i$ onto $\mathbf{V}$, and let $\mathbf{z} := \mathbf{R}\tilde{\mathbf{s}}_i - \mathbf{u}$. Then $\mathbf{z}$ is orthogonal to $\mathbf{V}$ and

$$\mathbf{Rs}_i = \mathbf{Rv} + \mathbf{R}\tilde{\mathbf{s}}_i = \mathbf{Rv} + \mathbf{u} + \mathbf{z} = (\mathbf{Rv} + \mathbf{u}) + \mathbf{z}.$$

By construction, $\mathbf{Rv} + \mathbf{u} \in \mathbf{V}$ and hence, since $\mathbf{z}$ is orthogonal to $\mathbf{V}$, this must be the ith GS-vector of $\mathbf{S_R}$. Since $\mathbf{z}$ is the projection of $\mathbf{R}\tilde{s}_i$ on $\mathbf{V}^{\perp}$, it can be obtained that $||\mathbf{z}|| \leq ||\mathbf{R}\tilde{\mathbf{s}}_i||$. Hence, for all $i = 1, ..., 2N$, the norm of the ith element of the Gram-Schmidt orthogonalisation of $\mathbf{S_R}$ is no greater than the norm of the image under

$\mathbf{R}$ of the ith element of the Gram-Schmidt orthogonalisation of $\mathbf{S}$ itself. In particular, if $1 \leq i_0 \leq 2N$ corresponds to an element of the Gram-Schmidt orthogonalisation of $\mathbf{S_R}$ of maximal norm, so that its norm is, in fact, equal to $||\mathbf{S_R}||_{GS}$, then we must have $||\mathbf{S_R}||_{GS} \leq ||\mathbf{R}\tilde{\mathbf{s}}_{i_0}|| \leq \max_{1 \leq i \leq 2N} ||\mathbf{R}\tilde{\mathbf{s}_i}||$. □

This allows Theorem 1 to be proved.

Proof. Let the private basis from level $(l-1)$ be denoted $\mathbf{S}_{l-1} = \{\mathbf{s_1}, \mathbf{s_2}, ..., \mathbf{s}_{2N}\}$ and define $GSB'_l := GSB_l/\eta_\epsilon(\mathbb{Z}^{2N})$. With overwhelming probability, the GS-norm of the basis associated to the vectors generated in Delegation (Algorithm 28) satisfies:

$$\begin{aligned} ||\mathbf{S}'_l||_{GS} &\leq \max_{1 \leq i \leq 2n} ||\mathbf{R}\tilde{\mathbf{s}_i}|| \text{ from Lemma 3} \\ &\leq \max_{1 \leq i \leq 2n} ||\tilde{\mathbf{s}_i}|| \cdot \sigma_{\mathbf{R}} \cdot \sqrt{2N} \cdot \eta_\epsilon^2(\mathbb{Z}^{2N}) \text{ from Lemma 2} \\ &= ||\mathbf{S}_{l-1}||_{GS} \cdot \sigma_{\mathbf{R}} \cdot \sqrt{2N} \cdot \eta_\epsilon^2(\mathbb{Z}^{2N}) = GSB'_l \end{aligned}$$

Since $GSB_l = GSB'_l \cdot \eta_\epsilon(\mathbb{Z}^{2N})$ by assumption, as $||\mathbf{S}_l|| \leq GSB_l$, the given bound on the output delegated basis can be deduced.

□

When considering $\mathbf{S}_l = \mathbf{S}_0 \cdot \mathbf{R}_l^{-1}$, where $\mathbf{R}_l$ is the product of l matrices, it can be shown that

$$GSB_l = ||\mathbf{S}_0||_{GS} \cdot \left(\sigma_{\mathbf{R}} \cdot \sqrt{2N} \cdot \eta_\epsilon^2(\mathbb{Z}^{2N})\right)^l \cdot \eta_\epsilon(\mathbb{Z}^{2N})$$

holds. This is due to the fact that multiplication by each matrix-component $Hash(ID_l)$ of the $\mathbf{R}_l$ creates an additional factor of $\sigma_{\mathbf{R}} \cdot \sqrt{2N} \cdot \eta_\epsilon^2(\mathbb{Z}^{2N})$ in the GSB_l. The $FFTsampler$, which introduces the factor of $\eta_\epsilon(\mathbb{Z}^{2N})$ into GSB_l, is only called

once per delegation in order to randomised the basis after the lattice Λ_l has been generated, and so does not exacerbate down the levels.

The practical security of Skinny Latte closely follows that of Latte [Campbell and Groves, 2017, ETS, 2019]. The next section details the security considerations to the scheme, focussing on how the Latte attacks apply to Skinny Latte.

Statistical Security: The FFT Sampler uses a short basis for a lattice Λ to produce a vector sample from a distribution that is close to the discrete Gaussian distribution $\mathcal{D}_{\mathbf{t},0}$ over Λ. If the distribution is not sufficiently close to the target distribution, then the sampling can potentially leak the short basis.

The accuracy of the FFT Sampler depends on the size of the standard deviation σ relative to the Gram-Schmidt norm of the input basis $\mathbf{S}$. According to Prest [2017], it is sufficient to take $\sigma \leq 1.312||\mathbf{S}||_{GS}$. This is derived from Lemma 6 in that paper, which says that for any $\epsilon \in \left(0, \frac{1}{4}\right)$ if $\sigma \geq \eta_\epsilon(\mathbb{Z}^{2N}) \cdot ||\mathbf{S}||_{GS}$, then the Rényi divergence R (introduced in §2.8) between $\mathcal{D} = \mathcal{D}_{\Lambda(\mathbf{S}),\sigma,\mathbf{t},\mathbf{S}}$ and the output distribution $\mathcal{D}_\epsilon$ of the $FFTSampler$ verifies

$$R(\mathcal{D}_\epsilon||\mathcal{D}) \leq \left(1 + \frac{a(a-1)\delta^2}{2(1-\delta)^{a+1}}\right)^{\frac{1}{a-1}}$$

which, as $\delta \leftarrow 0$, becomes $\delta = 1 + \frac{a\delta^2}{2}$, where $\left(\frac{1+\frac{\epsilon}{N}}{1-\frac{\epsilon}{N}}\right)^N - 1 \approx 2\epsilon$. The reader is referred to §2.4 for definitions of the smoothing parameter $\eta_\epsilon(\Lambda)$. In this equation, a is the order of Rényi divergence used in [Prest, 2017] and ϵ is from smoothing parameter definition.

As the authors follow NIST regulations, it is justified to follow their approach, which is based on findings of Prest [2017]. Taking $\epsilon = \frac{1}{2^{32}}$, the smoothing parameter

can be computed as in Falcon and therefore the standard deviation is set as $\sigma \leq 1.312||\mathbf{S}||_{GS}$.

Decryption Failure: The Fujisaki-Okamoto transform [Fujisaki and Okamoto, 1999] converts the CPA-secure PKE scheme of Skinny Latte into a CCA-secure PKE scheme, in the random oracle model, therefore protecting against active attacks that use malformed ciphertexts. However, the probability of decryption failure needs to meet the required security level. The analysis of failure probability follows that in ETS [2019] and Peikert [2014].

The error term d in Skinny Latte decryption at all levels is:

$$\mathbf{d} = \mathbf{e_2} - \mathbf{e_0} \cdot \mathbf{t_0} - \mathbf{e_1} \cdot t_1$$

where the $\mathbf{t}_i$ have coefficients sampled from a discrete Gaussian with standard deviation σ_l and the $\mathbf{e}_i$ have coefficients sampled from an approx discrete Gaussian with standard deviation $\sigma_{\mathbf{e}}$. Therefore the coefficients of $\mathbf{d}$ can be modelled as Gaussian with standard deviation

$$\tau = \sqrt{\sigma_e^2 + 2N\sigma_l^2\sigma_e^2}$$

From this point, the analysis follows that of Latte in ETS [2019]. Decryption will fail if for some $i \in \{0, ..., 255\}$,

$$|d_{iu}| + ... + |d_{i(u+1)-1}| > \frac{uq}{4}$$

as this will cause the rounding step to fail. The Chernoff-Cramer bound states the probability that $\langle \mathbf{d}, \mathbf{v} \rangle \geq k$ for any $\mathbf{v} \in \mathbb{R}_n$ is $< e^{-k^2/2\tau^2||v||^2}$ by the case used in Alkim et al. [2016]. Restricting to vectors $\mathbf{v}$ with entries $(v_{iu}, ..., v_{i(u+1)-1}) \in \{1, -1\}$ and 0

elsewhere implies the probability that

$$|d_{iu}| + ... + |d_{i(u+1)-1}| = \max_{v} \langle \mathbf{d}, \mathbf{v} \rangle > k$$

for a fixed $i \in \{0, ..., 255\}$ is

$$< 2^u e^{-k^2/2\tau^2 u}$$

and therefore the probability that the bound is exceeded for any $i \in \{0, ..., 255\}$ is $< 2^{u+8} e^{-k^2/2\tau^2 u}$.

Master Key Recovery: Recovering the master key requires solving the NTRU problem, i.e. recovering the polynomials $\mathbf{f}, \mathbf{g}$ from $\mathbf{h}$, where $\mathbf{h} = \mathbf{g} \cdot \mathbf{f}^{-1}$. The security analysis for this is identical to that of Latte and can be found in Annex C of ETS [2019]. This analysis is outlined here.

Lattice reduction with block size β will find the target vector provided that its projection onto the vector space spanned by the final β Gram-Schmidt vectors are shorter than the length of the $(2N - \beta + 1)$ Gram-Schmidt vector. This corresponds to

$$\sigma_0 \sqrt{\beta} \leq GH(\beta)^{\frac{2\beta - 2N}{\beta - 1}} \cdot det(\Lambda_0)^{1/2N}$$

where $GH(\beta) = \Gamma(\frac{\beta}{2} + 1)^{1/N} / \sqrt{\pi}$. As $det(\Lambda_0) = q^N$ and $\sigma_0 = \sqrt{\frac{qe}{4N}}$, an inequality over N can be reached and solved for $min(\beta)$.

The estimated attack costs are given in Table 5.10 using the security cost estimation formula given in ETS [2019]:

classical security = $2^{0.292\beta + 16.5}$ and *quantum security* = $2^{0.265\beta + 16.4}$

Table 5.10 LATTE Master Key Recovery Estimated Costs in bits Campbell and Groves [2017]

N	β	**Classical Security**	**Quantum Security**
1024	972	300	273
2048	1960	588	535
4096	3943	1167	1061

Delegated Key Recovery: Delegation to a sub-KMS at level l involves finding several sufficiently short vectors (s_0, s_1) in the lattice Λ_l with public basis:

$$\mathbf{B}_l = \begin{bmatrix} \mathcal{A}(\mathbf{h}) & \mathbf{I}_N \\ q\mathbf{I}_N & \mathbf{O}_N \end{bmatrix} \cdot \mathbf{R}_l^{-1}$$

of length $\sigma_l\sqrt{2N}$.

Lattice reduction with block size β will find a vector of the required length provided that it is longer than the first Gram-Schmidt vector. This corresponds to:

$$\sigma_l \cdot \sqrt{2N} \geq GH(\beta)^{\frac{2N}{N-1}} \cdot det(\Lambda_l)^{\frac{1}{2N}}$$

where the following hold:

$$det(\Lambda_l) = q^N$$

$$\sigma_l = \sigma_{l-1} \cdot \sqrt{2N} \cdot \eta_\epsilon^3(\mathbb{Z})$$

$$\sigma_0 = \sqrt{\frac{qe}{4N}}$$

The inequality can then be solved for $min(\beta)$ over N and l, and resulting security levels are given in Table 5.11.

Table 5.11 Skinny Latte Estimated cost of delegated key recovery

N	l	β	**Classical Security**	**Quantum Security**
1024	1	941	291	266
	2	370	125	114
2048	1	2064	619	563
	2	859	267	244
	3	497	162	148
4096	2	1907	573	522
	3	1140	349	319
	4	781	245	223

User Key Recovery: Extraction of user key involves solving CVP in lattice Λ_l by finding a short solution to equation

$$\begin{bmatrix} \mathcal{A}(\mathbf{h}) & \mathbf{I}_N \\ q\mathbf{I}_N & \mathbf{O}_N \end{bmatrix} \cdot \mathbf{R}_l^{-1} \begin{bmatrix} \mathbf{t}_0 \\ \mathbf{t}_1 \end{bmatrix} = b$$

An attacker must find $(\mathbf{t}_0, \mathbf{t}_1)$ with length such that

$$\sigma_l \cdot \sqrt{2(l+2)} \cdot \sqrt{2N} \geq GH(\beta)^{\frac{(2\beta-2N)}{\beta-1}} \cdot det(\Lambda_l)^{\frac{1}{2N}}$$

where $\sigma_l, GH(\beta)$ and $det(\Lambda_l)$, are as above and so minimal β depends on only N. For each parameter set, these, along with estimated cost of lattice reduction, are given in Table 5.12.

Table 5.12 Skinny Latte Estimated cost of user key recovery

N	l	β	**Classical Security**	**Quantum Security**
1024	1	733	231	211
	2	312	108	99
2048	1	1659	501	456
	2	746	234	214
	3	444	146	134
4096	2	1685	509	463
	3	1035	319	291
	4	720	227	207

Message Recovery: Message recovery can be executed via the primal-CVP attack. This recovers the ephemeral private key $\mathbf{e}$, and from that computes the other $\mathbf{e_0}, \mathbf{e_1}$. In the attack, the dimension of the lattice can be reduced to $m + N$. For Skinny Latte, it is required that $m \leq N$ as the dimension of Λ_l is $2N$ for all hierarchical levels. The estimated cost for this uses the method of Alkim et al. [2016]. The estimated primal-CVP attack costs are given in Table 5.13.The other attack is dual-SVP, again in a reduced lattice of dimension $m + N$. Estimated attack costs are given in Table 5.14, with $m \leq N$ as required for Skinny Latte.

Table 5.13 Skinny Latte ($l = 1$) Message Recovery via Primal-CVP Estimated Costs

N	**$\log_2$q**	**m**	β	**Classical Security**	**Quantum Security**
1024	**24**	2037	1403	410	372
2048	**25**	4092	3062	895	812
4096	**26**	8171	6549	1915	1737

Table 5.14 Skinny Latte ($l = 1$) Estimated cost of message recovery Dual-SVP attack

N	$\log_2 q$	m	β	Classical Security	Quantum Security
1024	**24**	2043	1395	370	289
2048	**25**	4087	3045	890	807
4096	**26**	8191	6512	1904	1727

Setting parameters

The scheme parameters are:

- dimension of the ring = $2N$
- modulus = q
- standard deviation of key generation sampler : $\sigma_0 = \sqrt{\frac{qe}{4N}}$
- standard deviation of delegation/extract samplers at level l: $\sigma_l = \sigma_{l-1} \cdot \sqrt{2N} \cdot \eta_\epsilon^2(\mathbb{Z}^{2N})$, due to basis delegation result in Theorem 1
- The GS-norm of output delegated basis is checked to be as expected, by performing a GS-norm check in the FFT Sampler. The GS-norm bound is

$$GSB_l = ||\mathbf{S}_{l-1}||_{GS} \cdot \sigma_{\mathbf{R}} \cdot \sqrt{2N} \cdot \eta_\epsilon^3(\mathbb{Z}^{2N})$$

  The GSB must be such that the user keys which are extracted from the delegated basis are sufficiently small for the required security levels.

- Noise parameters also need to be decreased as in ABB-scheme, so $\alpha_l = \frac{1}{\sigma_l \cdot \sqrt{2N}}$.

- Standard deviation of sampler for $\mathbf{R}$: $\sigma_{\mathbf{R}}$

Considering all the possible attacks addressed in this section, the parameter recommendations for Skinny Latte are given in Table 5.15.

Table 5.15 Proposed Skinny Latte Parameters

Set	**Security**			N	q
	$l=1$	$l=2$	$l=3$		
SK-LATTE-1	231	108	-	1024	$2^{24}-2^{14}+1$
SK-LATTE-2	501	234	146	2048	$2^{25}-2^{12}+1$
SK-LATTE-3	509	319	227	4096	$2^{26}-2^{14}+1$

Provable Security

The provable security of Skinny Latte in the ROM (as discussed in §2.2.2) relies on two simulation tools.

Proofs of security of the ABB-scheme make use of an algorithm $SampleRwithBasis$. A corresponding tool is proposed for Skinny Latte as Algorithm 33. On input of a lattice basis $\mathbf{A} = \begin{bmatrix} \mathcal{A}(\mathbf{h}) & \mathbf{I}_N \\ q\mathbf{I}_N & \mathbf{O}_N \end{bmatrix}$, it outputs a low-norm matrix R and a short basis for the lattice defined by $A \cdot R^{-1}$.

Algorithm 33: $SampleRwithBasis$

Data: h which defines basis A
Result: R, short basis $\mathbf{S}_B$ for $\Lambda(AR^{-1})$,

1 Run NTRUGen to generate an NTRU lattice defined by uniform-looking basis $\mathbf{B}$ and short basis $\mathbf{S}_B$ s.t. $||\mathbf{S_B}||_{GS} \leq \sigma_{\mathbf{R}}/\omega(\sqrt{log2n})$
2 For $i=1,2, j=1,2$: sample $\mathbf{r}_{i,j}$ using FFT sampler to find vectors in $\mathbf{B}$ with centre $\mathbf{h}$ and s.d. $\sigma_{\mathbf{R}}$
3 Set $\mathbf{R}$ to be the public NTRU basis matrix with polynomial $\mathbf{r}$
4 **return** $\mathbf{R}$, $\mathbf{S_B}$

The security proof also relies on a simulator $\mathcal{S}^{\mathcal{A}}$ which can answer key delegation and extraction queries without knowledge of the secret key.

Indistinguishability under chosen plaintext adaptive identity attack (ID-IND-CPA) is proven using a ROM adaptively secure HIBE security game for Skinny Latte. Section 5.3 of Agrawal et al. [2010c] presents the underlying security of their adaptively secure HIBE (on which Skinny Latte is built). A corresponding theorem and security game is presented for this scheme. To prove the HIBE security, a security game is defined as follows:

For a security parameter λ, let M_λ denote the message space, and C_λ denote the ciphertext space. The following security game between the challenger and adversary is used to define the security notion.

Setup: The challenger runs the key generation algorithm and generates a master public/private key pair (pk, sk).

Phase 1: The adversary is allowed to query oracles for delegation and extraction .

Challenge: The adversary chooses a target sequence of identifiers $(ID_1^*, ..., ID_l^*)$, where the inputs to the oracle queries did not include $(ID_1^*, ..., ID_i^*)$ for any $i \leq l$, and issues a plaintext $\mathbf{m} \in M_\lambda$ upon which it wishes to be challenged.

The challenger picks a random bit $r \in \{0, 1\}$ and a random ciphertext $\mathbf{c} \in C_\lambda$. If $r = 0$, $\mathbf{c}^* = Encrypt(\mathbf{m})$. If $r = 1$, $\mathbf{c}^* = \mathbf{c}$. The element $\mathbf{c}^*$ is sent to the adversary as the challenge ciphertext.

Phase 2: The adversary is allowed additional adaptive queries to the oracle, where they do not include $(ID_1^*, ..., ID_l^*)$ for any $i \leq l$. The challenger responds as in Phase 1.

Guess: The adversary outputs a guess $r' \in \{0, 1\}$ and wins if $r = r'$, i.e. determines whether $\mathbf{c}^*$ was the encryption of $\mathbf{m}$ or a random ciphertext.

An adversary $\mathcal{AD}$'s advantage is:

$$Adv_{HIBE}(\mathcal{AD}) = |p_1(\mathcal{AD}) - p_0(\mathcal{AD})|$$

where $p_1(\mathcal{AD})$ is the probability that the adversary guesses correctly, and $p_0(\mathcal{AD})$ is the probability that it guesses incorrectly.

In Skinny Latte, key extraction at level l aims to find a short solution $(\mathbf{t_0}, \mathbf{t_1})$ to the equation

$$\mathbf{S}_l \cdot R_l^{-1} \begin{bmatrix} \mathbf{t}_0 \\ \mathbf{t}_1 \end{bmatrix} = \mathbf{b}.$$

A simulator $\mathcal{S}^{\mathcal{AD}}$ which is able to answer delegation and key extraction queries without access to the private key is utilised. The bound on the advantage for an adversary $\mathcal{AD}$ becomes

$$ADV_{HIBE}(\mathcal{AD}) \leq l\mathcal{Q}_H^{l-1} ADV_{PKE}(\mathcal{S}^{\mathcal{AD}}) + negl(N)$$

where $\mathcal{Q}_H$ is the number of calls to the random oracle, l is the hierarchical level and $ADV_{PKE}(\mathcal{S}^{\mathcal{AD}})$ is the advantage of $\mathcal{S}^{\mathcal{AD}}$, the adversary for the underlying encryption scheme. This places a bound on the advantage of adversary $\mathcal{AD}$. The security proof showing the reduction to R-LWE using this game concept is now given.

Theorem 2. *If, in the random oracle model, there exists a Probabilistic Polynomial Time (PPT) adversary $\mathcal{AD}$ with IND-ID-CPA PPT advantage $\epsilon = \Omega(\lambda^k)$ for some $k > 0$ against the Skinny Latte HIBE scheme, then there exists a PPT algorithm b that decides the*

R-LWE problem with advantage $\epsilon' = \Omega(\lambda^k/Q_H^d)$, where Q_H is the number of queries made by $\mathcal{AD}$ to the random oracle H and d is the hierarchy depth.

Proof. Let $\mathcal{AD}$ be an IND-ID-CPA attacker. It will be shown that a non-negligible advantage ϵ in the IND-ID-CPA game can be used to decide the R-LWE problem with advantage ϵ/Q_H^d. This will prove that under the R-LWE assumption no polynomial-time attacker can have non-negligible advantage in the IND-ID-CPA game. Recall that R-LWE is about recognising an oracle $\mathcal{O}$.

Instance: b requests from $\mathcal{O}$ and receives, for $i = 0, ..., 2N$ a fresh R-LWE instance $(S_0, v_i) \in \mathbb{Z}_q \times \mathbb{Z}_q$, where $S_0 = (\mathbf{s}_i)_{1 \leq i \leq N}$ is an NTRU polynomial.

Setup: b prepares a simulated attack environment for adversary $\mathcal{AD}$ as follows.

1. For each level of the hierarchy, i where $1 \leq i \leq l$, select a random integer $Q_i^* \in [Q_H]$ where $[Q_H]$ is the maximum number of random-oracle queries to H that $\mathcal{AD}$ can make.
2. The public master key is $B_0 = \begin{bmatrix} \mathcal{A}(\mathbf{h}) & \mathbf{I}_N \\ q\mathbf{I}_N & \mathbf{O}_N \end{bmatrix}$ and the private keys are the corresponding $\mathbf{f}, \mathbf{g}$ (from NTRU instance, so that $\mathbf{h} = \mathbf{g}\mathbf{f}^{-1}$).

Random-oracle hash queries: When the oracle H is queried with input $(ID_1, ..., ID_l)$, it runs $SampleRwithBasis(\mathbf{h})$ to obtain a random $\mathbf{R} \in \mathcal{D}_N$ and a short basis $\mathbf{T_B}$ for $\mathbf{B} = \mathbf{S}_0 \cdot \mathbf{R}^{-1} \mod q$ and programs the output of the oracle to be $H(ID_1||...||ID_l) = R_l$, such that (public basis at level$l$) $= \mathbf{B}_0 \cdot \mathbf{R}_l^{-1}$.

Queries 1: $\mathcal{AD}$ makes interactive key extraction queries on arbitrary identities, chosen adaptively. When the delegation oracle is queried with input $(ID_1, ..., ID_l)$, the simulator uses the known short basis for the lattice corresponding to $\mathbf{S}_l$ to produce a delegated basis for the delegated lattice in the same way as normal delegation.

When the extraction oracle is queried with input $(ID_1, ..., ID_l)$, the simulator uses the known short basis for the lattice corresponding to A_l to find a short solution $(\mathbf{t_0}, \mathbf{t_1})$ to the equation

$$\mathbf{S}_0 \cdot \mathbf{R}_l^{-1} \cdot \begin{bmatrix} \mathbf{t}_0 \\ \mathbf{t}_1 \end{bmatrix} = \mathbf{b}.$$

Challenge: $\mathcal{AD}$ announced to b the identity ID^* on which it wishes to be challenged. say ID^* is at level l. It is required that ID^* not be equal to, or a descendant of, any identity ID for which a private key has been or will be requested in any preceding and subsequent key extraction query. $\mathcal{AD}$ also submits a message bit $b^* \in \{0, 1\}$ to be encrypted.

Then b proceeds as follows.

1. Let $v_0 = v_0$ and $\mathbf{v}^* = \begin{bmatrix} \mathbf{v}_1, ..., \mathbf{v}_n \end{bmatrix} \in \mathbb{Z}_q^n$ from the R-LWE instance.
2. Blind the message bit by letting $c_0^* = v_0 + b^* \lfloor \frac{q}{2} \rceil \in \mathbb{Z}_q$
3. Set $\mathbf{c}_1^* = \mathbf{v}^* \in \mathbb{Z}_q^N$
4. Choose a random bit $r \leftarrow \{0, 1\}$. If $r = 0$, set $CT^* = (c_0^*, \mathbf{c}_1^*)$ and send to the adversary. If $r = 1$, choose a random $(c_0, \mathbf{c}_1) \in \mathbb{Z}_q \times \mathbb{Z}_q^N$ and send this to $\mathcal{AD}$.

Queries 2: $\mathcal{AD}$ makes more extraction queries answered by b in the same manner as before.

Guess: After being allowed to make additional queries, $\mathcal{AD}$ guesses whether CT^* was an encryption of b^* for ID^*. Upon receiving $\mathcal{AD}$'s guess, b ends the simulation and outputs its answer to R-LWE. If $\mathcal{AD}$ guesses correctly, b answers pseudo-random. If $\mathcal{AD}$ guesses incorrectly, b answers random. □

5.5 Conclusions

This chapter introduces a novel HIBE scheme, Skinny LATTE, over structured lattices which keeps the dimension constant. The rational behind this is that together with the compact form of the public keys, the sampling process behind the user key extraction and delegation functions can also be accelerated. The first performance results of NCSC-proposed HIBE scheme Latte are given, and it is speculated that this may not be practical for use in IoT devices as the hierarchical level is increased. This is because, for example for parameter set LATTE-1, the ciphertext reaches XkB and extract takes Ys at level 3. Skinny LATTE aims to fill this gap. A theoretical description and analysis of the scheme is given, and parameter sets are proposed. These results suggest that Skinny LATTE will be a more appropriate solution for applications where there are many levels, yet LATTE maybe be more applicable to scenarios with a wide, but not deep, hierarchical tree.

CHAPTER 6

Conclusion and Future Work

This chapter summarises the main findings of the thesis and outlines potential avenues for extending this research.

6.1 Conclusion Summary

The overall aim of this thesis has been achieved, particularly in improving confidence in lattice-based cryptography by considering both basic primitives submitted to the NIST process, and more advanced primitives which are only offered by lattices. The feasibility of these schemes for real-world deployment has been demonstrated and security against known attacks has been considered and incorporated into the interpretation of the results. This includes both algorithmic attacks and physical attacks. Software implementations of all the covered schemes have been presented, which in fact outperform traditional forms of cryptography and include protection against recent attacks in literature.

For all the research in this thesis, the proposed software designs are optimised in ANSI C, which allows for portability, and are intended for general-purpose applications

ranging from high-end 64-bit Intel Xeon servers to 32-bit ARM Cortex-M embedded systems. Designs improve over previous research in lattice-based cryptography and elliptic-curve cryptography. Moreover, the proposed software designs outperform previous research either in terms of a reduction in the runtimes, a reduction in memory consumption, an improvement in physical attack resistance, or a combination of these. The performance improvements have been achieved by utilising the underlying structure of lattices for speedups, considering alternative algorithms for key generation and sampling, the use of NTT for polynomial multiplication by modifying parameters, and relaxing certain conditions which had a negligible effect on security.

Additionally, advanced lattice-based primitives were developed. As lattices are the only post-quantum contender to support these primitives, and with developments in technology demanding more sophisticated primitives, these algorithmic and software designs were the first of their kind. Throughout all of the research, attack countermeasures such as constant-time components and checks were considered and integrated into the designs where possible.

The conclusions from Chapter 3, Chapter 4, and Chapter 5, are now presented in the following sections, with Section 6.2 detailing several future research avenues for lattice-based cryptography.

6.1.1 Chapter 3

Chapter 3 provides a comprehensive evaluation of three lattice-based signature schemes, two NIST candidates Falcon [Prest et al., 2017] and Dilthium [Lyubashevsky et al., 2017] and one long-standing lattice-based signature BLISS-B [Ducas, 2014a]. Physical attack vulnerabilities of each of the schemes are examined, countermeasures proposed,

and a performance evaluation of each of the schemes with applicable counter-measures in place is given.

The main contribution of this chapter is the first fault attack on the Falcon signature scheme, submitted to the NIST process. Falcon employs a novel trapdoor sampler, which can be viewed as a recursive form of the GPV sampler in the FFT domain. A attacked termed Bearz is presented, which extracts the secret lattice basis by introducing a fault into the recursive sampler, resulting in faulty signatures. Given enough of these faulty signatures, a basis spanning the lattice defined by secret polynomial $\mathbf{f}$ can be obtained, and from this the remainder of the secret basis can be recovered.

Discrete Gaussian sampler are utilised in both the BLISS-B and Falcon schemes. Gaussian samplers are known to be vulnerable to side-channel attacks, as recognised in [Bruinderink et al., 2016b, Pessl, 2016a, Saarinen, 2017]. In particular, the attack on BLISS(-B)[Bruinderink et al., 2016b] (Flush, Gauss Reload), was able to recover the secret key with 3500 signatures by exploiting cache access patterns to learn a few coefficients of the sampled noise vector. In order to protect against these types of attacks, countermeasures such as constant-time samplers and BlindVector techniques are proposed. Falcon is also vulnerable to these attacks and so we recommend incorporating them into this design also.

Despite Dilithium being considered more side-channel secure due to its utilisation of a uniform rather than Gaussian sampler, attacks targeting the signing procedure exist [Ravi et al., 2018, Bruinderink and Pessl, 2018]. [Ravi et al., 2018] mounts a power analysis attack on the polynomial multiplication component. Counter-measures were proposed to improve resistance to these attacks.

Performance results of each scheme with the recommended counter-measures are given. This aims to provide a more realistic and fair benchmarking platform, as in practise the counter-measures would have to be taken into account. Interestingly, Dilithium is found to have the greatest drop in performance when its associated countermeasures are applied, almost 20% compared to 5% of Falcon. BLISS-B remains the fastest signature scheme.

6.1.2 Chapter 4

This chapter presents the first practical implementation of an advanced lattice-based cryptographic primitive. The DLP-IBE scheme [Ducas et al., 2014a] was the first scheme to show potential for real world deployment, and the contribution of this chapter demonstrates this by improving upon the PoC implementation by over 180%.

Algorithmic design methods include the use of Barrett reduction, Brown's modular GCD, relaxation of the updating of standard-deviation during the short vector sampling process and enforcing sampled-basis norm checks in the key generation. Software design elements include multi-precision arithmetic, constant-time processes, use of the CDT discrete Gaussian sampler, and entropy coding.

Results show extract is $3\times$ as fast as the PoC, encrypt $5\times$ and decrypt $12\times$ as such. Furthermore, the results are two orders of magnitude faster than existing pairings-based schemes, indicating that lattice-based IBE would be a suitable quantum-safe alternative.

6.1.3 Chapter 5

This chapter extended the IBE schemes further, to allow finer grained access control and to offload workload of the KMS. The Latte HIBE scheme was introduced in 2017 [Campbell and Groves, 2017] as an alternative to the pairings-based IBE scheme Mikey-Sakke [Groves, 2012], currently used within UK government communications. However, the increase in lattice dimension during the delegation process causes the user public key and ciphertext to grow at each level, so by level 3 at 160-bit security, they are over 48kB each and so the memory and power consumption would be too high for practical use. This is verified through the first implementation of the scheme, presented in this chapter.

To address this problem, Skinny Latte is proposed. This fixes the dimension of the lattice across all levels, at the expense of a drop in security. A security analysis and proof is given, alongside parameter recommendations and expected key and ciphertext sizes. These show that Skinny Latte is a promising HIBE system for deployment.

6.2 Topics For Future Research

This thesis has shown the potential of lattice-based cryptography for real world use, particularly on constrained devices such as those used in IoT. As the field of LBC, and QSC as a whole, is still very new, there are a lot of open problems and unexplored territory. The purpose of this section is to suggest extensions of the research given here, and related topics which are opportunities for future research.

6.2.1 QSC + QKD

Another area gaining traction with regards to the post-quantum era is that of Quantum Key Distribution (QKD). QKD creates key material by using the quantum properties of light to transfer information from one user to another. After the first authenticated key, all following sessions are deemed secure, and the key length can be increased each session without a loss in security [Cobourne and Cid, 2011]. However, quantum cryptography systems are expensive, difficult to implement in a communication network, not fully integrated with conventional algorithmic cryptography and lack a process for security evaluation and certification. A solution for this is a hybrid QSC/QKD construction. Current work on hybrid schemes include [aqu, 2018]. An ideal outcome of this project would be a quantum-safe protocol integrating QSC and QKD, to which existing applications could smoothly transition.

6.2.2 Further advanced primitives

This thesis investigated the first practical quantum-safe (H)IBE schemes. Lattices are unique in that they are able to support other advanced schemes such as attribute-based encryption, functional encryption, deterministic encryption, proxy re-encryption, homomorphic encryption, and secure multi-party computations. All of these are open research areas with regards to creating practical schemes whose security we have confidence in. Additional research on HIBE could be carried out to develop certificate-less PKE, which mitigates the key escrow security risk. This concept is based on IBE, but the user identity does not solely determine the user public key. Rather, the key generation process is divided between the key generator and the user. For instance, the user will generate some randomness to complete the user key. This

mitigates the danger of key escrow. The public key can be authenticated using the user identity. An extension of this research would be to develop DLP-IBE into such a scheme. Additionally, an KEM scheme can be constructed. This consists of a KEM together with a digital signature scheme. Therefore we can use Pino et al. [Pino et al., 2016] KEM (based on NTRU) with this digital signature scheme to form a AKE scheme. Both these additional schemes offer the hardness properties and quantum-resilience of lattice-based primitives.

6.2.3 Gaussian Sampling over the lattice

The Gaussian Sampler is a main bottleneck of the schemes explored. There is scope to investigate other variants of the GPV sampler, by computing the memory-heavy Gram-Schmidt orthogonalisation on the fly or further acceleration by properties of the NTRU basis structure. The Gram-Schmidt orthogonalisation process is performed within the sampling process. As highlighted in [Lyubashevsky and Prest, 2015], one can either choose to precompute the GS-basis and store in $O(n^2)$ elements with sampling taking $O(n^2)$ time, or else compute during the sampling process, which will then take $O(n^3)$ time. As the orthogonalisation process does not retain the cyclic structure of ideal lattice bases, the storage remains the same in this scenario. However, Lyubashevsky and Prest [Lyubashevsky and Prest, 2015] propose a method of computing the Gram-Schmidt orthogonalisation in $O(n^2)$, with additional speed-ups for NTRU lattices. It would be an interesting study to investigate the effects of these alternative sampling algorithms when incorporated within the DLP-IBE scheme.

6.2.4 Tightening security

Much of the security of lattice-based primitives is based on strong assumptions, for example structured lattices are only assumed to be as secure as standard lattices as there have *not yet been any feasible attacks*. But this has not been proven, and so standard lattices are usually recommended in use cases where security has to hold long-term and can not be easily patched, for instance within satellite communication systems, government requirements, and long term data storage. The reduction of standard LWE to R-LWE would be a ground-breaking result. However, further cryptanaylsis of these structures primitives would still be an interesting research area. Module lattices have only been used since quite recently and so would benefit from more attention from this aspect. Efforts towards determining the concrete hardness of LWE problems been seen in [Albrecht et al., 2015, Albrecht, 2018] and there is room for development due to more recent attacks and new proposed structures, such as module lattices. Refining security assumptions and parameter recommendations would lead to the tightening of security of multiple lattice-based primitives.

6.2.5 Further NIST evaluation

The current focus of the QSC community is on the Round 2 candidates. NIST have specified [Moody, 2019] that when making the selections for a potential Round 3, the focus will be on performance of the candidates on both hardware and software, agility of the protocols and how they fit into existing applications, and factors such as misuse-resistance. Continued cryptanalysis of the schemes will be valued, and perhaps investigations into migration strategies, for example hybrid QSC-ECC-type schemes, for post-standardisation deployment. The switch to quantum-safe forms of

cryptography is a given and so even if we do not yet know what the algorithms are, we can begin to prepare for the switchover, or even to start using what is available alongside existing forms of cryptography to pre-empt any problem-solving surrounding the roll-out of quantum-safe systems.

References

Terrestrial Trunked Radio (TETRA). Standard, The European Telecommunications Standards Institute, Sophia-Antipolis, France, January 2005.

Security of mission critical push to talk (mcptt) over lte. Standard, Third Generation Partnership Project (3GPP), Sophia-Antipolis, France, May 2016.

AQuaSec: Agile quantum safe communications. https://gtr.ukri.org/projects?ref=104615, 2018.

The PALISADE lattice cryptography library. https://git.njit.edu/palisade/PALISADE, 2018.

Quantum-Safe Identity-based Encryption. Standard, The European Telecommunications Standards Institute, Sophia-Antipolis, France, 2019.

Asmaa Abdallah and Xuemin Sherman Shen. A lightweight lattice-based homomorphic privacy-preserving data aggregation scheme for smart grid. *IEEE Transactions on Smart Grid*, 9(1):396–405, 2016.

Shweta Agrawal and Xavier Boyen. Identity-based encryption from lattices in the standard model. *Manuscript, July*, 2009.

Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (h)ibe in the standard model. In *EUROCRYPT*, pages 553–572, 2010a.

Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (h)ibe in the standard model. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 553–572. Springer, 2010b.

Shweta Agrawal, Dan Boneh, and Xavier Boyen. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical ibe. In *Annual Cryptology Conference*, pages 98–115. Springer, 2010c.

Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *STOC*, pages 99–108, 1996.

Miklós Ajtai and Cynthia Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *Proceedings of the Twenty-ninth Annual ACM Symposium on Theory of Computing*, STOC '97, pages 284–293, New York, NY, USA, 1997. ACM. ISBN 0-89791-888-6. doi: 10.1145/258533.258604. URL http://doi.acm.org/10.1145/258533.258604.

Sedat Akleylek, Nina Bindel, Johannes A. Buchmann, Juliane Krämer, and Giorgia Azzurra Marson. An efficient lattice-based signature scheme with provably secure instantiation. In *AFRICACRYPT*, pages 44–60, 2016. doi: 10.1007/978-3-319-31517-1_3. URL http://dx.doi.org/10.1007/978-3-319-31517-1_3.

Martin Albrecht. Security estimates for the learning with errors problem. https://bitbucket.org/malb/lwe-estimator/src/master/, 2018.

Martin R Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. *Journal of Mathematical Cryptology*, 9(3):169–203, 2015.

Erdem Alkim, Nina Bindel, Johannes A Buchmann, Özgür Dagdelen, and Peter Schwabe. Tesla: Tightly-secure efficient signatures from standard lattices. *IACR Cryptology ePrint Archive*, 2015:755, 2015.

Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange - A new hope. In *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016.*, pages 327–343, 2016. URL https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/alkim.

Joël Alwen and Chris Peikert. Generating shorter bases for hard random lattices. In *26th International Symposium on Theoretical Aspects of Computer Science STACS 2009*, pages 75–86. IBFI Schloss Dagstuhl, 2009.

Hyeongcheol An, Rakyong Choi, Jeeun Lee, and Kwangjo Kim. Performance evaluation of liboqs in open quantum safe project (part i). In *2018 Symposium on Cryptography and Information Security (SCIS 2018)*. IEICE Technical Committee on Information Security, 2018.

Daniel Apon, Xiong Fan, and Feng-Hao Liu. Compact identity based encryption from lwe. *IACR Cryptology ePrint Archive*, 2016.

Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In *Annual International Cryptology Conference*, pages 595–618. Springer, 2009.

Nicolas Aragon, Paulo Barreto, Slim Bettaieb, Loic Bidoux, Olivier Blazy, Jean-Christophe Deneuville, Phillipe Gaborit, Shay Gueron, Tim Guneysu, Carlos Aguilar Melchor, Rafael Misoczki, Edoardo Persichetti, Nicolas Sendrier, Jean-Pierre Tillich, and Gilles Zemor. Bike. Technical report, National Institute of Standards and Technology, 2017a. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

Nicolas Aragon, Phillipe Gaborit, Adrien Hautevillle, Olivier Ruatta, and Gilles Zemor. Ranksign. Technical report, National Institute of Standards and Technology, 2017b. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

Sanjeev Arora, László Babai, Jacques Stern, and Z Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. *Journal of Computer and System Sciences*, 54(2):317–331, 1997.

AC Atici, Benedikt Gerlachs, and Ingrid Verbauwhede. Power analysis on NTRU implementations for RFIDs: First results. In *Workshop on RFID Security*, 2008.

Derek Atkins, Iris Anshel, Dorian Goldfeld, and Paul E Gunnells. Walnutdsa. Technical report, National Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

Daniel Augot, Matthieu Finiasz, and Nicolas Sendrier. A family of fast syndrome based cryptographic hash functions. In *International Conference on Cryptology in Malaysia*, pages 64–83. Springer, 2005.

László Babai. On lovász' lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1):1–13, 1986a.

László Babai. On Lovasz lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1):1–13, 1986b.

Shi Bai and Steven D. Galbraith. An improved compression technique for signatures based on learning with errors. pages 28–47, 2014.

Shi Bai, Tancrède Lepoint, Adeline Roux-Langlois, Amin Sakzad, Damien Stehlé, and Ron Steinfeld. Improved security proofs in lattice-based cryptography: using the rényi divergence rather than the statistical distance. *Journal of Cryptology*, 31(2): 610–640, 2018.

Gustavo Banegas, Paolo S. L. M. Barreto, Brice Odilon Boidje, Pierre-Louis Cayrel, Gilbert Ndollane Dione, Kris Gaj, Cheikh Thiecoumba Gueye, Richard Haeussler, Jean Belo Klamti, Ousmane N'diaye, Duc Tri Nguyen, Edoardo Persichetti, and Jefferson E. Ricardini. Dags. Technical report, National Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

Paulo SLM Barreto, Patrick Longa, Michael Naehrig, Jefferson E Ricardini, and Gustavo Zanon. Sharper ring-lwe signatures. *IACR Cryptology ePrint Archive*, 2016:1026, 2016.

Daniel J. Bernstein, Josh Fried, Nadia Heninger, Paul Lou, and Luke Valenta. Post-quantum rsa-encryption. Technical report, National Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

Pauline Bert, Pierre-Alain Fouque, Adeline Roux-Langlois, and Mohamed Sabt. Practical implementation of ring-sis/lwe based signature and ibe. In *International Conference on Post-Quantum Cryptography*, pages 271–291. Springer, 2018.

Nina Bindel, Johannes Buchmann, and Juliane Krämer. Lattice-based signature schemes and their sensitivity to fault attacks. In *Fault Diagnosis and Tolerance in Cryptography (FDTC), 2016 Workshop on*, pages 63–77. IEEE, 2016.

Nina Bindel, Sedat Akleylek, Erdem Alkim, Paulo S. L. M. Barreto, Johannes Buchmann, Edward Eaton, Gus Gutoski, Juliane Kramer, Patrick Longa, Harun Polat, Jefferson E. Ricardini, and Gustavo Zanon. qtesla. Technical report, National Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

Johannes Blömer, Ricardo Gomes Da Silva, Peter Günther, Juliane Krämer, and Jean-Pierre Seifert. A practical second-order fault attack against a real-world pairing implementation. In *Fault Diagnosis and Tolerance in Cryptography (FDTC), 2014 Workshop on*, pages 123–136. IEEE, 2014.

Dan Boneh and Xavier Boyen. Efficient selective-id secure identity-based encryption without random oracles. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 223–238. Springer, 2004.

Dan Boneh and Matt Franklin. Identity-based encryption from the Weil pairing. In *Annual International Cryptology Conference*, pages 213–229. Springer, 2001a.

Dan Boneh and Matt Franklin. Identity-based encryption from the weil pairing. In *Annual international cryptology conference*, pages 213–229. Springer, 2001b.

Dan Boneh and Matthew K. Franklin. Identity-based encryption from the weil pairing. *SIAM J. Comput.*, 32(3):586–615, 2003.

Dan Boneh, Matt Franklin, Ben Lynn, Matt Pauker, Rishi Kacker, and Gene Tsudik. IBE Secure E-mail.

Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 440–456. Springer, 2005.

Dan Boneh, Craig Gentry, Sergey Gorbunov, Shai Halevi, Valeria Nikolaenko, Gil Segev, Vinod Vaikuntanathan, and Dhinakaran Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit abe and compact garbled circuits. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 533–556. Springer, 2014.

Xavier Boyen. Attribute-based functional encryption on lattices. In *TCC*, pages 122–142, 2013.

Matt Braithwaite. Experimenting with Post-Quantum Cryptography . https://security.googleblog.com/2016/07/experimenting-with-post-quantum.html/, 2016. [Online; accessed May-2019].

Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) lwe. *SIAM Journal on Computing*, 43(2):831–871, 2014.

Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. *ACM Transactions on Computation Theory (TOCT)*, 6(3):13, 2014.

Séamus Brannigan, Neil Smyth, Tobias Oder, Felipe Valencia, Elizabeth OÕSullivan, Tim Güneysu, and Francesco Regazzoni. An investigation of sources of randomness within discrete Gaussian sampling. Cryptology ePrint Archive, Report 2017/298, 2017. http://eprint.iacr.org/2017/298.

W Steven Brown. On euclid's algorithm and the computation of polynomial greatest common divisors. In *Proceedings of the second ACM symposium on Symbolic and algebraic manipulation*, pages 195–211. ACM, 1971.

Leon Groot Bruinderink and Peter Pessl. Differential fault attacks on deterministic lattice signatures. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 21–43, 2018.

Leon Groot Bruinderink, Andreas Hülsing, Tanja Lange, and Yuval Yarom. Flush, Gauss, and reload-a cache attack on the BLISS lattice-based signature scheme. In *International Conference on Cryptographic Hardware and Embedded Systems*, pages 323–345. Springer, 2016a.

Leon Groot Bruinderink, Andreas Hülsing, Tanja Lange, and Yuval Yarom. Flush, gauss, and reload - A cache attack on the BLISS lattice-based signature scheme. In *CHES*, pages 323–345, 2016b. doi: 10.1007/978-3-662-53140-2_16. URL http://dx.doi.org/10.1007/978-3-662-53140-2_16.

Johannes Buchmann, Erik Dahmen, and Andreas Hülsing. Xmss-a practical forward secure signature scheme based on minimal security assumptions. In *International Workshop on Post-Quantum Cryptography*, pages 117–129. Springer, 2011.

Johannes Buchmann, Daniel Cabarcas, Florian Göpfert, Andreas Hülsing, and Patrick Weiden. Discrete Ziggurat: A time-memory trade-off for sampling from a Gaussian distribution over the integers. In *SAC*, pages 402–417, 2013a.

Johannes Buchmann, Daniel Cabarcas, Florian Göpfert, Andreas Hülsing, and Patrick Weiden. Discrete ziggurat: A time-memory trade-off for sampling from a Gaussian distribution over the integers. In *International Conference on Selected Areas in Cryptography*, pages 402–417. Springer, 2013b.

Peter Campbell and Michael Groves. Practical post-quantum hierarchical identity-based encryption, 2017. available at https://www.qub.ac.uk/csit/FileStore/Filetoupload,785752,en.pdf.

David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 523–552. Springer, 2010.

David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. *Journal of cryptology*, 25(4):601–639, 2012.

Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. Csidh: an efficient post-quantum commutative group action. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 395–427. Springer, 2018.

Yuanmi Chen and Phong Q. Nguyen. Bkz 2.0: Better lattice security estimates. In *Proceedings of the 17th International Conference on The Theory and Application of Cryptology and Information Security*, ASIACRYPT'11, pages 1–20, Berlin, Heidelberg, 2011. Springer-Verlag. ISBN 978-3-642-25384-3. URL http://dx.doi.org/10.1007/978-3-642-25385-0_1.

Pengqi Cheng, Yan Gu, Zihong Lv, Jianfei Wang, Wenlei Zhu, Zhen Chen, and Jiwei Huang. A performance analysis of identity-based encryption schemes. In *INTRUST*, pages 289–303. Springer, 2011.

Jung Hee Cheon, Kyoohyung Han, Andrey Kim, Miran Kim, and Yongsoo Song. A full rns variant of approximate homomorphic encryption. In *International Conference on Selected Areas in Cryptography*, pages 347–368. Springer, 2018.

Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachene. *A Homomorphic LWE Based E-voting Scheme*, pages 245–265. Post-Quantum Cryptography. Springer, 2016.

Sheila Cobourne and Carlos Cid. Quantum key distribution: Awesome or pointless? https://cdn.ttgtmedia.com/rms/pdf/RHUL_Cobourne_Final.pdf, 2011.

Clifford Cocks. An identity based encryption scheme based on quadratic residues. In *IMA International Conference on Cryptography and Coding*, pages 360–363. Springer, 2001a.

Clifford Cocks. An identity based encryption scheme based on quadratic residues. In *IMA International Conference on Cryptography and Coding*, pages 360–363. Springer, 2001b.

James W Cooley and John W Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90):297–301, 1965.

Nicolas T Courtois, Matthieu Finiasz, and Nicolas Sendrier. How to achieve a mceliece-based digital signature scheme. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 157–174. Springer, 2001.

Wei Dai, Yarkın Doröz, Yuriy Polyakov, Kurt Rohloff, Hadi Sajjadpour, Erkay Savaş, and Berk Sunar. Implementation and evaluation of a lattice-based key-policy abe scheme. *IEEE Transactions on Information Forensics and Security*, 13(5):1169–1184, 2017.

Ivan Damgård, Valerio Pastro, Nigel Smart, and Sarah Zakarias. Multiparty computation from somewhat homomorphic encryption. In *Annual Cryptology Conference*, pages 643–662. Springer, 2012.

Dipayan Das and Vishal Saraswat. Compact lattice signatures. In *ICETE*, 2018.

Ruan De Clercq, Sujoy Sinha Roy, Frederik Vercauteren, and Ingrid Verbauwhede. Efficient software implementation of ring-lwe encryption. In *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition*, pages 339–344. EDA Consortium, 2015.

Xinyue Deng. An introduction to lenstra-lenstra-lovasz lattice basis reduction algorithm. Available at http://https://math.mit.edu/~apost/courses/18.204-2016/18.204_Xinyue_Deng_final_paper.pdf (2020/05/14).

Jintai Ding, Ming-Shing Chen, Albrecht Petzoldt, Dieter Schmidt, and Bo-Yin Yang. Rainbow. Technical report, National Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

Irit Dinur, Guy Kindler, and Shmuel Safra. Approximating-cvp to within almost-polynomial factors is np-hard. In *Proceedings 39th Annual Symposium on Foundations of Computer Science (Cat. No. 98CB36280)*, pages 99–109. IEEE, 1998.

Danny Dolev, Cynthia Dwork, and Moni Naor. Nonmalleable cryptography. *SIAM review*, 45(4):727–784, 2003.

Léo Ducas. Accelerating Bliss: the geometry of ternary polynomials. *IACR Cryptology ePrint Archive*, 2014:874, 2014a. URL http://eprint.iacr.org/2014/874.

Léo Ducas. Accelerating bliss: the geometry of ternary polynomials. *IACR Cryptology ePrint Archive*, 2014:874, 2014b.

Léo Ducas and Thomas Prest. Fast fourier orthogonalization. In *Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation*, pages 191–198. ACM, 2016a.

Léo Ducas and Thomas Prest. Fast fourier orthogonalization. In *Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation*, pages 191–198. ACM, 2016b.

Léo Ducas, Alain Durmus, Tancrède Lepoint, and Vadim Lyubashevsky. Lattice signatures and bimodal Gaussians. In *CRYPTO (1)*, pages 40–56, 2013a. Full version: https://eprint.iacr.org/2013/383.pdf.

Léo Ducas, Alain Durmus, Tancrède Lepoint, and Vadim Lyubashevsky. Lattice signatures and bimodal Gaussians. In *Advances in Cryptology–CRYPTO 2013*, pages 40–56. Springer, 2013b.

Léo Ducas, Vadim Lyubashevsky, and Thomas Prest. Efficient identity-based encryption over ntru lattices. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 22–41. Springer, 2014a.

Léo Ducas, Vadim Lyubashevsky, and Thomas Prest. Efficient identity-based encryption over NTRU lattices. In *ASIACRYPT*, pages 22–41, 2014b.

Léo Ducas, Vadim Lyubashevsky, and Thomas Prest. Efficient identity-based encryption over ntru lattices. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 22–41. Springer, 2014c.

Leo Ducas, Tancrede Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehle. Crystals–dilithium: Digital signatures from module lattices. *IACR Cryptology ePrint Archive*, 2017:633, 2017.

Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS-Dilithium: a lattice-based digital signature scheme. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2018 (1):238–268, 2018.

Léo Ducas. Accelerating BLISS: the geometry of ternary polynomials. Cryptology ePrint Archive, Report 2014/874, 2014c. https://eprint.iacr.org/2014/874.

Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, 1985.

T. Espitau, P. Fouque, B. Gérard, and M. Tibouchi. Loop-abort faults on lattice-based signature schemes and key exchange protocols. *IEEE Transactions on Computers*, 67 (11):1535–1549, Nov 2018. ISSN 0018-9340. doi: 10.1109/TC.2018.2833119.

Thomas Espitau, Pierre-Alain Fouque, Benoît Gérard, and Mehdi Tibouchi. Loop-abort faults on lattice-based fiat-shamir and hash-and-sign signatures. In *International Conference on Selected Areas in Cryptography*, pages 140–158. Springer, 2016.

Thomas Espitau, Pierre-Alain Fouque, Benoît Gérard, and Mehdi Tibouchi. Side-channel attacks on bliss lattice-based signatures: Exploiting branch tracing against strongswan and electromagnetic emanations in microcontrollers. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1857–1874. ACM, 2017.

J.-C. Faugere, L. Perret, and J. Ryckeghem. Dualmodems. Technical report, National Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Advances in Cryptology?CRYPTO?86*, pages 186–194. Springer, 1986.

FIPS 186-4. Digital Signature Standard. Standard, National Institute of Standards and Technology, Gaithersburg, MD, January 2013.

Jean-Bernard Fischer and Jacques Stern. An efficient pseudo-random generator provably as secure as syndrome decoding. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 245–255. Springer, 1996.

Ronald Aylmer Fisher, Frank Yates, et al. Statistical tables for biological, agricultural aad medical research. *Statistical tables for biological, agricultural aad medical research.*, 1938.

O. The FPLLL Development Team. fplll, a lattice reduction library. Available at https://github.com/fplll/fplll, 2016. URL https://github.com/fplll/fplll.

Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *Annual International Cryptology Conference*, pages 537–554. Springer, 1999.

Philippe Gaborit, Cédric Lauradoux, and Nicolas Sendrier. Synd: a fast code-based stream cipher with a security reduction. In *2007 IEEE International Symposium on Information Theory*, pages 186–190. IEEE, 2007.

Steven D Galbraith, Christophe Petit, Barak Shani, and Yan Bo Ti. On the security of supersingular isogeny cryptosystems. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 63–91. Springer, 2016.

Nicolas Gama and Phong Nguyen. Predicting lattice reduction. *Advances in Cryptology EUROCRYPT 2008*, pages 31–51, 2008.

Oscar Garcia-Morchon, Zhenfei Zhang, Sauvik Bhattacharya, Ronald Rietman, Ludo Tolhuizen, Jose-Luis Torre-Arce, Hayo Baan, Markku-Juhani O. Saarinen, Scott Fluhrer, Thijs Laarhoven, and Rachel Player. Round5. Technical report, National Institute of Standards and Technology, 2019. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.

Alexandre Gélin and Benjamin Wesolowski. Loop-abort faults on supersingular isogeny cryptosystems. In *International Workshop on Post-Quantum Cryptography*, pages 93–106. Springer, 2017.

Craig Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, pages 169–178, 2009.

Craig Gentry and Alice Silverberg. Hierarchical id-based cryptography. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 548–566. Springer, 2002.

Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions, 2007. http://eprint.iacr.org/.

Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, pages 197–206, 2008.

Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *Annual Cryptology Conference*, pages 75–92. Springer, 2013.

Danilo Gligoroski and Kristian Gjosteen. Edon-k. Technical report, National Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Public-key cryptosystems from lattice reduction problems. *Electronic Colloquium on Computational Complexity (ECCC)*, 3(56), 1996.

Oded Goldreich, Daniele Micciancio, Shmuel Safra, and J-P Seifert. Approximating shortest lattice vectors is not harder than approximating closest lattice vectors. *Information Processing Letters*, 71(2):55–61, 1999.

Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of computer and system sciences*, 28(2):270–299, 1984.

Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Comput.*, 17(2):281–308, apr 1988. ISSN 0097-5397.

Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *ACM Conference on Computer and Communications Security*, pages 89–98, 2006.

Michael Groves. Mikey-sakke: Sakai-kasahara key encryption in multimedia internet keying (mikey). Technical report, 2012.

T. Güneysu and T. Oder. Towards lightweight identity-based encryption for the post-quantum-secure internet of things. In *2017 18th International Symposium on Quality Electronic Design (ISQED)*, pages 319–324, 2017. ISBN 1948-3287. doi: 10.1109/ISQED.2017.7918335. ID: 1.

Tim Güneysu, Vadim Lyubashevsky, and Thomas Pöppelmann. Practical lattice-based cryptography: A signature scheme for embedded systems. In *CHES*, pages 530–547, 2012a.

Tim Güneysu, Vadim Lyubashevsky, and Thomas Pöppelmann. Practical lattice-based cryptography: A signature scheme for embedded systems. Cryptographic Hardware and Embedded Systems–CHES 2012, pages 530–547. Springer, 2012b.

Mike Hamburg. Three bears. Technical report, National Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

G. Hanrot, X. Pujol, and D. Stehle. http://caramel.loria.fr/sem-slides/201105051030.pdf.

Juan Pedro Hecht and Jorge Alejandro Kamlofsky. Hk17. Technical report, National Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

Martin Henk. Note on shortest and nearest lattice vectors. *Information Processing Letters*, 61(4):183–188, 1997.

M Jason Hinek. Lattice attacks in cryptography: A partial overview. *School of Computer Science, University of Waterloo, Canada*, 2004.

Philip Hodgers, Francesco Regazzoni, Richard Gilmore, Ciara Moore, and Tobias Oder. State-of-the-art in physical side-channel attacks and resistant technologies. *Technical report*, 2016.

W Hoeffding. Probability inequalities for sums of bounded random variables. j. amer. statist. assoc. 58. 1963.

Jeff Hoffstein, Jill Pipher, John M Schanck, Joseph H Silverman, William Whyte, and Zhenfei Zhang. Choosing parameters for ntruencrypt. In *Cryptographers' Track at the RSA Conference*, pages 3–18. Springer, 2017.

Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A ring-based public key cryptosystem. In *ANTS*, pages 267–288, 1998a.

Jeffrey Hoffstein, Jill Pipher, and Joseph H Silverman. Ntru: A ring-based public key cryptosystem. In *International Algorithmic Number Theory Symposium*, pages 267–288. Springer, 1998b.

Jeffrey Hoffstein, Nick Howgrave-Graham, Jill Pipher, Joseph H. Silverman, and William Whyte. NTRUSign: Digital signatures using the NTRU lattice. In *CT-RSA*, pages 122–140, 2003. ISBN 3-540-00847-0.

Jeremy Horwitz and Ben Lynn. Toward hierarchical identity-based encryption. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 466–481. Springer, 2002.

James Howe, Thomas Pöppelmann, Maire O'Neill, Elizabeth O'Sullivan, and Tim Güneysu. Practical lattice-based digital signature schemes. *ACM Transactions on Embedded Computing Systems (TECS)*, 14(3):41, 2015.

James Howe, Ayesha Khalid, Ciara Rafferty, Francesco Regazzoni, and Máire O'Neill. On Practical Discrete Gaussian Samplers For Lattice-Based Cryptography. *IEEE Transactions on Computers*, 2016.

David A Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40(9):1098–1101, 1952.

Andreas Hulsing, Daniel J. Bernstein, Christoph Dobraunig, Maria Eichlseder, Scott Fluhrer, Stefan-Lukas Gazdag, Panos Kampanakis, Stefan Kolbl, Tanja Lange, Martin M Lauridsen, Florian Mendel, Ruben Niederhagen, Christian Rechberger, Joost Rijneveld, and Peter Schwabe. Sphincs+. Technical report, National Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In *International Workshop on Post-Quantum Cryptography*, pages 19–34. Springer, 2011.

David Jao, Reza Azarderakhsh, Matthew Campagna, Craig Costello, Luca De Feo, Basil Hess Amir Jalali, Brian Koziel, Brian LaMacchia, Patrick Longa, Michael Naehrig, Joost Renes, Vladimir Soukharev, and David Urbanik. Sike. Technical report, National Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

David Jao, Reza Azarderakhsh, Matthew Campagna, Craig Costello, Luca De Feo, Basil Hess, Amir Jalali, Brian Koziel, Brian LaMacchia, Patrick Longa, Michael Naehrig, Joost Renes, Vladimir Soukharev, David Urbanik, and Geovandro Pereira. SIKE. Technical report, National Institute of Standards and Technology, 2019. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.

Gregory Kabatianskii, Ernst Krouk, and Ben Smeets. A digital signature scheme based on random error-correcting codes. In *IMA International Conference on Cryptography and Coding*, pages 161–167. Springer, 1997.

Angshuman Karmakar, Sujoy Sinha Roy, Oscar Reparaz, Frederik Vercauteren, and Ingrid Verbauwhede. Constant-time discrete gaussian sampling. *IEEE Transactions on Computers*, 2018.

Shuichi Katsumata and Shota Yamada. Partitioning via non-linear polynomial functions: More compact ibes from ideal lattices and bilinear maps. In *Advances in Cryptology ASIACRYPT 2016: 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part II 22*, pages 682–712. Springer, 2016.

Ayesha Khalid, Tobias Oder, Felipe Valencia, Maire O'Neill, Tim Güneysu, and Francesco Regazzoni. Physical protection of lattice-based cryptography: Challenges and solutions. In *Proceedings of the 2018 on Great Lakes Symposium on VLSI*, pages 365–370. ACM, 2018.

Philip Klein. Finding the closest lattice vector when it's unusually close. In *Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms*, pages 937–941. Society for Industrial and Applied Mathematics, 2000.

D. E. Knuth and A. C. Yao. The complexity of nonuniform random number generation. *Algorithms and complexity: new directions and recent results*, pages 357–428, 1976a.

Donald E Knuth and Andrew C Yao. The complexity of nonuniform random number generation. *Algorithms and complexity: new directions and recent results*, pages 357–428, 1976b.

Takeshi Koshiba and Katsuyuki Takashima. New assumptions on isogenous pairing groups with applications to attribute-based encryption. In Kwangsu Lee, editor, *Information Security and Cryptology – ICISC 2018*, pages 3–19, Cham, 2019. Springer International Publishing. ISBN 978-3-030-12146-4.

Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. *Designs, Codes and Cryptography*, 2014. ISSN 0925-1022.

Mun-Kyu Lee, Jeong Eun Song, Dooho Choi, and Dong-Guk Han. Countermeasures against power analysis attacks for the NTRU public key cryptosystem. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 93(1):153–163, 2010.

Dongxi Liu, Nan Li, Jongkil Kim, and Surya Nepal. Compact lwe. Technical report, National Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

Adriana López-Alt, Eran Tromer, and Vinod Vaikuntanathan. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 1219–1234. ACM, 2012.

Granit Luzhnica. *Pairing based cryptography and implementation in Java.* na, 2011.

V. Lyubashevsky. Round 2 official comment: qtesla, 2019. URL https://groups.google.com/a/list.nist.gov/d/msg/pqc-forum/HHnavSx4f5Q/fRsujb9ACgAJ.

V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. *Journal of the ACM (JACM)*, 60(6):43:1–43:35, 2013.

Vadim Lyubashevsky. Fiat-shamir with aborts: Applications to lattice and factoring-based signatures. In *ASIACRYPT*, pages 598–616, 2009.

Vadim Lyubashevsky. Lattice signatures without trapdoors. In *EUROCRYPT*, pages 738–755, 2012.

Vadim Lyubashevsky. Digital signatures based on the hardness of ideal lattice problems in all rings. In *Advances in Cryptology–ASIACRYPT 2016: 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part II 22*, pages 196–214. Springer, 2016.

Vadim Lyubashevsky. Lattice cryptography in the NIST standardization process. NIS Summer School 2018, 2018.

Vadim Lyubashevsky and Daniele Micciancio. Asymptotically efficient lattice-based digital signatures. In *Theory of Cryptography Conference*, pages 37–54. Springer, 2008.

Vadim Lyubashevsky and Thomas Prest. *Quadratic time, linear space algorithms for Gram-Schmidt orthogonalization and Gaussian sampling in structured lattices*, pages 789–815. Advances in Cryptology–EUROCRYPT 2015. Springer, 2015.

Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *EUROCRYPT*, pages 1–23, 2010a.

Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 1–23. Springer, 2010b.

Vadim Lyubashevsky, Leo Ducas, Eike Kiltz, Tancrede Lepoint, Peter Schwabe, Gregor Seiler, and Damien Stehle. CRYSTALS-Dilithium. Technical report, National Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

Vadim Lyubashevsky, Leo Ducas, Eike Kiltz, Tancrede Lepoint, Peter Schwabe, Gregor Seiler, and Damien Stehle. Crystals-dilithium. Technical report, National Institute of Standards and Technology, 2019. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.

George Marsaglia, Wai Wan Tsang, et al. The Ziggurat method for generating random variables. *Journal of statistical software*, 5(8):1–7, 2000.

Sarah McCarthy, Neil Smyth, and Elizabeth O?Sullivan. A practical implementation of identity-based encryption over ntru lattices. In *IMA International Conference on Cryptography and Coding*, pages 227–246. Springer, 2017.

Sarah McCarthy, James Howe, Neil Smyth, Seámus Brannigan, and Máire O'Neill. Bearz attack falcon: Implementation attacks with countermeasures on the falcon signature scheme. In *SECRYPT*, volume 2019, 2019.

Robert J. McEliece. A public-key cryptosystem based on algebraic coding theory. *DSN progress report*, 42(44):114–116, 1978.

Ralph C Merkle. A certified digital signature. In *Conference on the Theory and Application of Cryptology*, pages 218–238. Springer, 1989.

Daniele Micciancio and Shafi Goldwasser. *Complexity of lattice problems: a cryptographic perspective*, volume 671. Springer Science & Business Media, 2012.

Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *EUROCRYPT*, pages 700–718, 2012.

Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on Gaussian measures. In *FOCS*, pages 372–381, 2004.

Microsoft. LatticeCrypt. https://www.microsoft.com/en-us/research/project/lattice-cryptography-library/, 2016.

Dustin Moody. What was NIST thinking? Oxford post-quantum cryptography workshop, 2019.

Gordon E Moore et al. Cramming more components onto integrated circuits, 1965.

Michele Mosca. Cybersecurity in an era with quantum computers: will we be ready? *IEEE Security & Privacy*, 16(5):38–41, 2018.

David Naccache, Phong Q. Nguyen, Michael Tunstall, and Claire Whelan. Experimenting with Faults, Lattices and the DSA. In *International Workshop on Public Key Cryptography*, pages 16–28. Springer, 2005.

Michael Naehrig, Erdem Alkim, Joppe Bos, Leo Ducas, Karen Easterbrook, Brian LaMacchia, Patrick Longa, Ilya Mironov, Valeria Nikolaenko, Christopher Peikert, Ananth Raghunathan, and Douglas Stebila. Frodokem. Technical report, National Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

Dalit Naor, Moni Naor, and Jeff Lotspiech. Revocation and tracing schemes for stateless receivers. In *Annual International Cryptology Conference*, pages 41–62. Springer, 2001.

JAM Naranjo, JA López-Ramos, and LG Casado. Applications of the extended euclidean algorithm to privacy and secure communications. In *Proc. of 10th International Conference on Computational and Mathematical Methods in Science and Engineering*, pages 702–713, 2010.

Phong Q. Nguyen and Oded Regev. Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures. *J. Cryptology*, 22(2):139–160, 2009.

Harald Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. *Prob. Control and Inf. Theory*, 15(2):159–166, 1986.

NIST. Post-quantum crypto project. http://csrc.nist.gov/groups/ST/post-quantum-crypto/, 2016. Accessed: 27.07.2017.

Máire O'Neill, Elizabeth O'Sullivan, Gavin McWilliams, Markku-Juhani Saarinen, Ciara Moore, Ayesha Khalid, James Howe, Rafaël Del Pino, Michel Abdalla, Francesco Regazzoni, Felipe Valencia, Tim Güneysu, Tobias Oder, Adrian Waller, Glyn Jones, Anthony Barnett, Robert Griffin, Andrew Byrne, Bassem Ammar, and David Lund. Secure Architectures of Future Emerging Cryptography *SAFEcrypto*. In *Proceedings of the ACM International Conference on Computing Frontiers, CF'16, Como, Italy, May 16-19, 2016*, pages 315–322, 2016.

Máire O'Neill, Elizabeth O'Sullivan, Gavin McWilliams, Markku-Juhani Saarinen, Ciara Moore, Ayesha Khalid, James Howe, Rafaël Del Pino, Michel Abdalla, Francesco Regazzoni, Felipe Valencia, Tim Güneysu, Tobias Oder, Adrian Waller, Glyn Jones, Anthony Barnett, Robert Griffin, Andrew Byrne, Bassem Ammar, and David Lund. Secure architectures of future emerging cryptography (safecrypto). Springer, 2020.

Sebastien Ordas, Ludovic Guillaume-Sage, Karim Tobich, Jean-Max Dutertre, and Philippe Maurine. Evidence of a larger em-induced fault model. 11 2014. doi: 10.1007/978-3-319-16763-3_15.

Dan Page and Frederik Vercauteren. A fault attack on pairing-based cryptography. *IEEE Transactions on Computers*, 55(9):1075–1080, 2006.

Chris Peikert. An efficient and parallel Gaussian sampler for lattices. In *CRYPTO*, pages 80–97, 2010a.

Chris Peikert. An efficient and parallel Gaussian sampler for lattices. In *Annual Cryptology Conference*, pages 80–97. Springer, 2010b.

Chris Peikert. Lattice cryptography for the internet. In *international workshop on post-quantum cryptography*, pages 197–219. Springer, 2014.

Chris Peikert. He gives c-sieves on the csidh. 2019.

Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. In *Annual international cryptology conference*, pages 554–571. Springer, 2008.

Yossi (Joseph) Peretz and Nerya Granot. Srtpi. Technical report, National Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

Peter Pessl. Analyzing the shuffling side-channel countermeasure for lattice-based signatures. In *Progress in Cryptology - INDOCRYPT 2016 - 17th International Conference on Cryptology in India, Kolkata, India, December 11-14, 2016, Proceedings*, pages 153–170, 2016a. doi: 10.1007/978-3-319-49890-4_9. URL http://dx.doi.org/10.1007/978-3-319-49890-4_9.

Peter Pessl. Analyzing the shuffling side-channel countermeasure for lattice-based signatures. In *International Conference in Cryptology in India*, pages 153–170. Springer, 2016b.

Peter Pessl, Leon Groot Bruinderink, and Yuval Yarom. To BLISS-B or not to be: Attacking strongSwan's Implementation of Post-Quantum Signatures. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1843–1855. ACM, 2017.

Christophe Petit. Faster algorithms for isogeny problems using torsion point images. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 330–353. Springer, 2017.

Rafael Del Pino, Vadim Lyubashevsky, and David Pointcheval. The whole is less than the sum of its parts: Constructing more efficient lattice-based AKEs. In *International Conference on Security and Cryptography for Networks*, pages 273–291. Springer, 2016.

Thomas Plantard, Arnaud Sipasseuth, Cedric Dumondelle, and Willy Susilo. Drs. Technical report, National Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

Thomas Pöppelmann, Léo Ducas, and Tim Güneysu. Enhanced lattice-based signatures on reconfigurable hardware. In *CHES*, pages 353–370, 2014. Full version: https://eprint.iacr.org/2014/254.pdf.

Thomas Poppelmann, Erdem Alkim, Roberto Avanzi, Joppe Bos, Leo Ducas, Antonio de la Piedra, Peter Schwabe, and Douglas Stebila. Newhope. Technical report, National Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

Thomas Pornin and Thomas Prest. More efficient algorithms for the ntru key generation using the field norm. In *IACR International Workshop on Public Key Cryptography*, pages 504–533. Springer, 2019.

Thomas Prest. Lattice-ibe. Available at https://github.com/tprest/Lattice-IBE, 2016. URL https://github.com/tprest/Lattice-IBE.

Thomas Prest. Sharper bounds in lattice-based cryptography using the rényi divergence. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 347–374. Springer, 2017.

Thomas Prest, Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. Falcon. Technical report, National Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

Thomas Prest, Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. Falcon. Technical report, National Institute of Standards and Technology, 2019. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.

Robert Primas, Peter Pessl, and Stefan Mangard. Single-trace side-channel attacks on masked lattice-based encryption. In *International Conference on Cryptographic Hardware and Embedded Systems*, pages 513–533. Springer, 2017.

Charles Rackoff and Daniel R Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *Annual International Cryptology Conference*, pages 433–444. Springer, 1991.

Prasanna Ravi, Mahabir Prasad Jhanwar, James Howe, Anupam Chattopadhyay, and Shivam Bhasin. Side-channel assisted existential forgery attack on dilithium-a nist pqc candidate. 2018.

O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):34:1–34:40, 2009.

Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, pages 84–93, 2005. ISBN 1-58113-960-8.

Oded Regev. The learning with errors problem. *Invited survey in CCC*, 2010.

Alfréd Rényi et al. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*. The Regents of the University of California, 1961.

Ronald L Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

C. B. Roellgen and G. Brands. Rvb. Technical report, National Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

Sujoy Sinha Roy, Oscar Reparaz, Frederik Vercauteren, and Ingrid Verbauwhede. Compact and side channel secure discrete Gaussian sampling. *IACR Cryptology ePrint Archive*, 2014:591, 2014a.

Sujoy Sinha Roy, Frederik Vercauteren, Nele Mentens, Donald Donglong Chen, and Ingrid Verbauwhede. Compact ring-lwe cryptoprocessor. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 371–391. Springer, 2014b.

Markku-Juhani O. Saarinen. tiny_sha3. https://github.com/mjosaarinen/tiny_sha3, 2015.

Markku-Juhani O. Saarinen. Arithmetic coding and blinding countermeasures for lattice signatures. Cryptology ePrint Archive, Report 2016/276, 2016. http://eprint.iacr.org/2016/276.

Markku-Juhani O. Saarinen. Arithmetic coding and blinding countermeasures for lattice signatures. *Journal of Cryptographic Engineering*, pages 1–14, 2017.

SAFEcrypto. libSAFEcrypto. https://github.com/safecrypto/libsafecrypto, 2018.

Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 457–473. Springer, 2005.

Ryuichi Sakai and Masao Kasahara. ID based cryptosystems with pairing on elliptic curve. *IACR Cryptology ePrint Archive*, 2003:54, 2003.

Simona Samardjiska, Ming-Shing Chen, Andreas Hulsing, Joost Rijneveld, and Peter Schwabe. Mqdss. Technical report, National Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

Nicholas Saunderson. *The Elements of Algebra, in Ten Books*, volume 2. Printed at the University Press, 1740.

Claus-Peter Schnorr and Martin Euchner. Lattice basis reduction: improved practical algorithms and solving subset sum problems. *Mathematical Programming*, 66(1-3): 181–199, 1994.

Peter Schwabe, Roberto Avanzi, Joppe Bos, Leo Ducas, Eike Kiltz, Tancrede Lepoint, Vadim Lyubashevsky, John M. Schanck, Gregor Seiler, and Damien Stehle. Crystals-kyber. Technical report, National Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

Michael Scott. A note on the implementation of the number theoretic transform. In *IMA International Conference on Cryptography and Coding*, pages 247–258. Springer, 2017.

Adi Shamir. Identity-based cryptosystems and signature schemes. In *Workshop on the Theory and Application of Cryptographic Techniques*, pages 47–53. Springer, 1984.

Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, October 1997. ISSN 0097-5397.

Victor Shroup. Ntl library. https://www.shoup.net/ntl/, 1996.

Joseph H Silverman and William Whyte. Timing attacks on NTRUEncrypt via variation in the number of hash calls. In *CT-RSA*, pages 208–224. Springer, 2007.

Douglas Stebila and Michele Mosca. Post-quantum key exchange for the internet and the open quantum safe project. In *International Conference on Selected Areas in Cryptography*, pages 14–37. Springer, 2016.

Damien Stehlé and Ron Steinfeld. Making NTRU as secure as worst-case problems over ideal lattices. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 27–47. Springer, 2011.

Jacques Stern. A new identification scheme based on syndrome decoding. In *Annual International Cryptology Conference*, pages 13–21. Springer, 1993.

Piotr Szczechowiak and Martin Collier. Tinyibe: Identity-based encryption for heterogeneous sensor networks. In *2009 International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, pages 319–354. IEEE, 2009.

Bill Tourloupis. Example: Drawing lattice points and vectors. http://www.texample.net/tikz/examples/lattice-points/, 2012.

VAMPIRELab. SUPERCOP. https://bench.cr.yp.to/supercop.html, 2019.

J. van de Pol and N. Smart. Estimating key sizes for high dimensional lattice-based systems. In *nbsp;Cryptography and Coding (IMACC)*, pages 290–303, 2013.

Scott Vanstone. Responses to nist's proposal. *Communications of the ACM*, 35(7):50–52, 1992.

Ingrid Verbauwhede, Duško Karaklajic, and Jörn-Marc Schmidt. The fault attack jungle-a classification model to guide you. In *2011 Workshop on Fault Diagnosis and Tolerance in Cryptography*, pages 3–8. IEEE, 2011.

Nikolay Vasilev Vizev. Side channel attacks on ntruencrypt. *University of Technology Darmstadt*, 2007.

An Wang, Xuexin Zheng, and Zongyue Wang. Power analysis attacks and countermeasures on NTRU-based wireless body area networks. *KSII Transactions on Internet and Information Systems (TIIS)*, 7(5):1094–1107, 2013.

Shota Yamada. Adaptively secure identity-based encryption from lattices with asymptotically shorter public parameters. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 32–62. Springer, 2016.

Andrew Chi-Chih Yao. Protocols for secure computations. In *FOCS*, volume 82, pages 160–164, 1982.

Yang Yu and Léo Ducas. Learning strikes again: The case of the drs signature scheme. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 525–543. Springer, 2018.

Zhenfei Zhang, Cong Chen, Jeffrey Hoffstein, and William Whyte. Ntruencrypt. Technical report, National Institute of Standards and Technology, 2017a. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

Zhenfei Zhang, Cong Chen, Jeffrey Hoffstein, and William Whyte. pqntrusign. Technical report, National Institute of Standards and Technology, 2017b. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

APPENDIX A

Reference code and algorithms

A.1 Random oracles used in DLP-IBE design

Algorithm 34: Random Oracle H(id) - convert the ID into a unique polynomial

Data: id, N, q
Result: $t \in \mathcal{R}_q$

```
1 s ← H(id)
2 t ← CSRNG(s) ∈ Z_q^N
3 return t
```

Algorithm 35: Random Oracle $H'(k)$ - create a byte stream to use as a one-time pad

Data: k, N, q
Result: $t \in \{0,1\}^m$

```
1 s ← H'(k)
2 t ← CSRNG(s) ∈ {0,1}^m
3 return t
```

A.2 FFT Sampler [Prest et al., 2017]

```
static void
ffSampling_fft(samplerZ samp, void *samp_ctx,
fpr *restrict z0, fpr *restrict z1,
const fpr *restrict tree,
const fpr *restrict t0, const fpr *restrict t1, unsigned logn,
fpr *restrict tmp)
{
size_t n, hn;
const fpr *tree0, *tree1;

n = (size_t)1 << logn;
if (n == 1) {
fpr x0, x1, sigma;

x0 = t0[0];
x1 = t1[0];
sigma = tree[0];
z0[0] = fpr_of(samp(samp_ctx, x0, sigma));
z1[0] = fpr_of(samp(samp_ctx, x1, sigma));
return;
}

hn = n >> 1;
```

```
tree0 = tree + n;
tree1 = tree + n + ffLDL_treesize(logn - 1);

/*
 * We split t1 into z1 (reused as temporary storage), then do
 * the recursive invocation, with output in tmp. We finally
 * merge back into z1.
 */
falcon_poly_split_fft(z1, z1 + hn, t1, logn);
ffSampling_fft(samp, samp_ctx, tmp, tmp + hn,
tree1, z1, z1 + hn, logn - 1, tmp + n);
falcon_poly_merge_fft(z1, tmp, tmp + hn, logn);

/*
 * Compute tb0 = t0 + (t1 - z1) * L. Value tb0 ends up in tmp[].
 */
memcpy(tmp, t1, n * sizeof *t1);
falcon_poly_sub_fft(tmp, z1, logn);
falcon_poly_mul_fft(tmp, tree, logn);
falcon_poly_add_fft(tmp, t0, logn);

/*
 * Second recursive invocation.
 */
falcon_poly_split_fft(z0, z0 + hn, tmp, logn);
```

```
ffSampling_fft(samp, samp_ctx, tmp, tmp + hn,
tree0, z0, z0 + hn, logn - 1, tmp + n);
falcon_poly_merge_fft(z0, tmp, tmp + hn, logn);
}
```

A.3 Merge FFT

```
void
falcon_poly_merge_fft(fpr *restrict f,
const fpr *restrict f0, const fpr *restrict f1, unsigned logn)
{
size_t n, hn, qn, u;

n = (size_t)1 << logn;
hn = n >> 1;
qn = hn >> 1;

/*
 * An extra copy to handle the special case logn = 1.
 */
f[0] = f0[0];
f[hn] = f1[0];

for (u = 0; u < qn; u ++) {
fpr a_re, a_im, b_re, b_im;
```

```
fpr t_re, t_im;

a_re = f0[u];
a_im = f0[u + qn];
FPC_MUL(b_re, b_im, f1[u], f1[u + qn],
fpr_gm_tab[((u + hn) << 1) + 0],
fpr_gm_tab[((u + hn) << 1) + 1]);
FPC_ADD(t_re, t_im, a_re, a_im, b_re, b_im);
f[(u << 1) + 0] = t_re;
f[(u << 1) + 0 + hn] = t_im;
FPC_SUB(t_re, t_im, a_re, a_im, b_re, b_im);
f[(u << 1) + 1] = t_re;
f[(u << 1) + 1 + hn] = t_im;
}
}
```

APPENDIX B

Falcon family

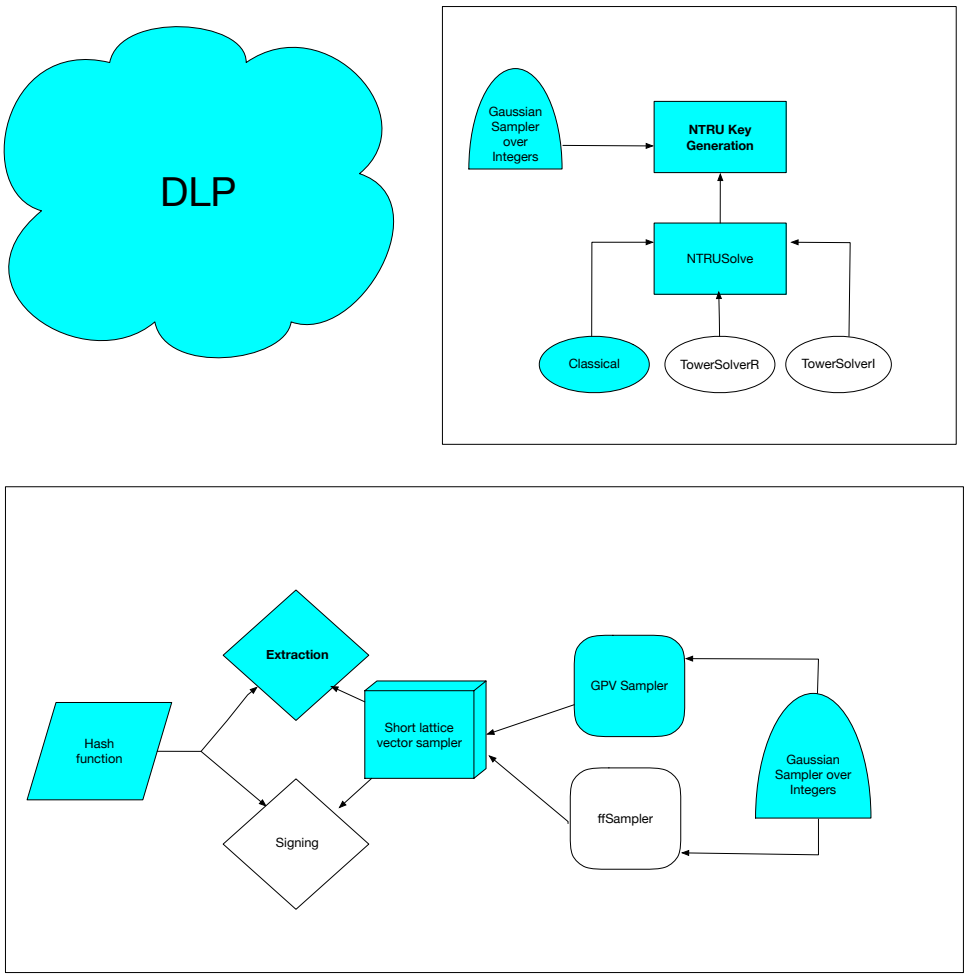


Fig. B.1 The components of the DLP IBE scheme

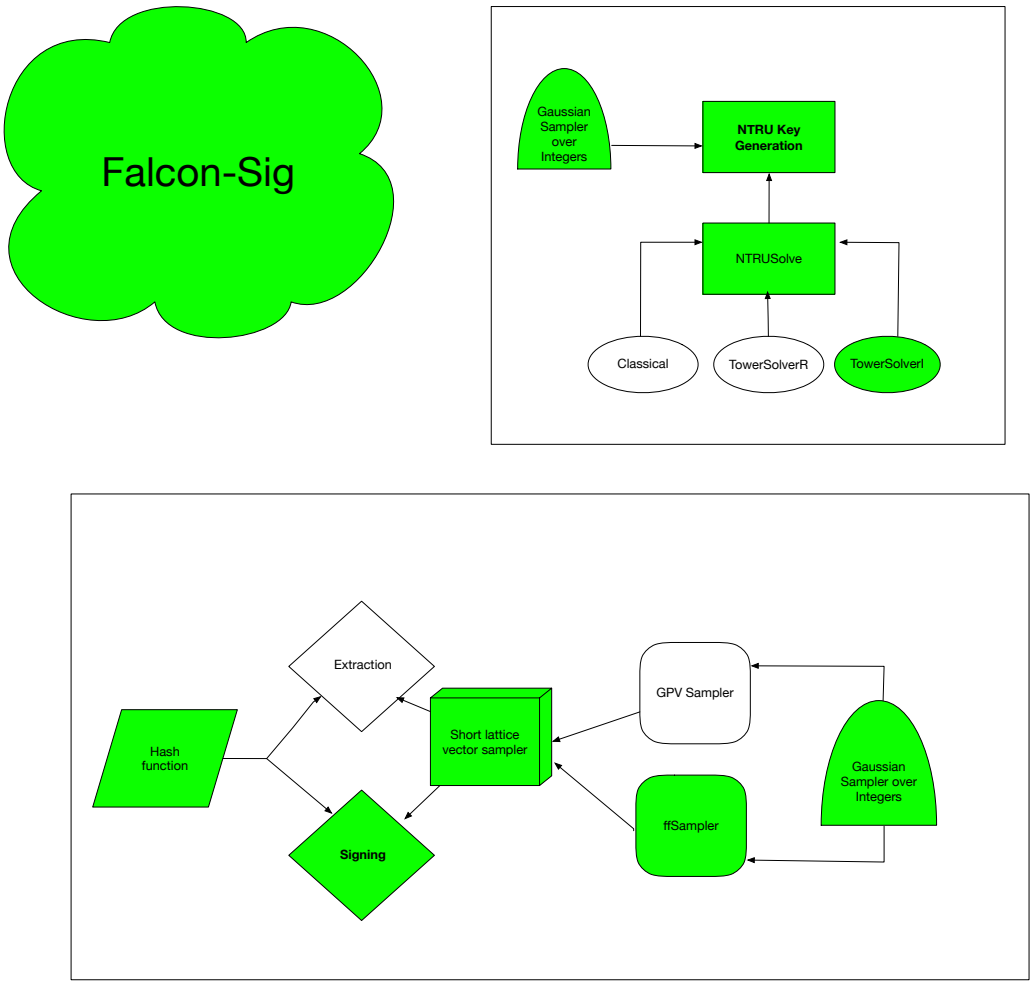


Fig. B.2 The components of the FALCON Signature scheme

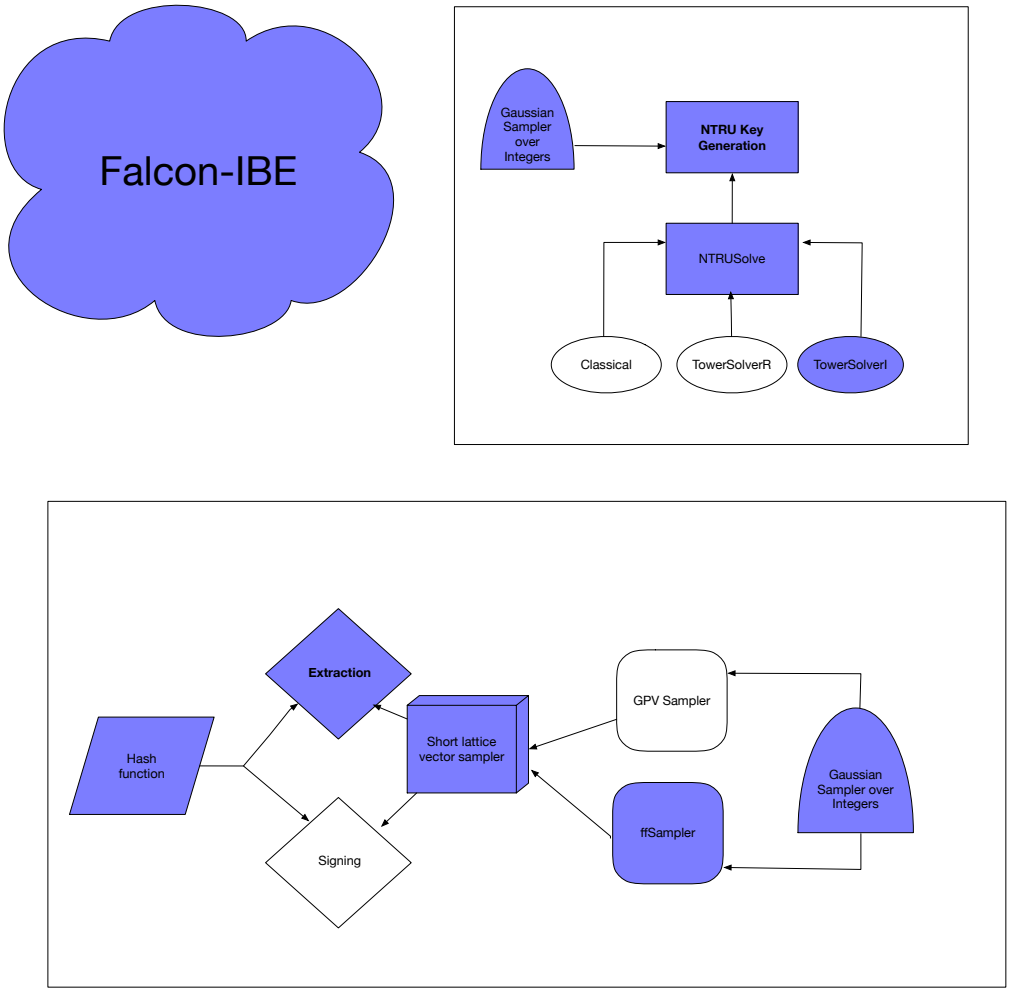


Fig. B.3 The components of the FALCON IBE scheme

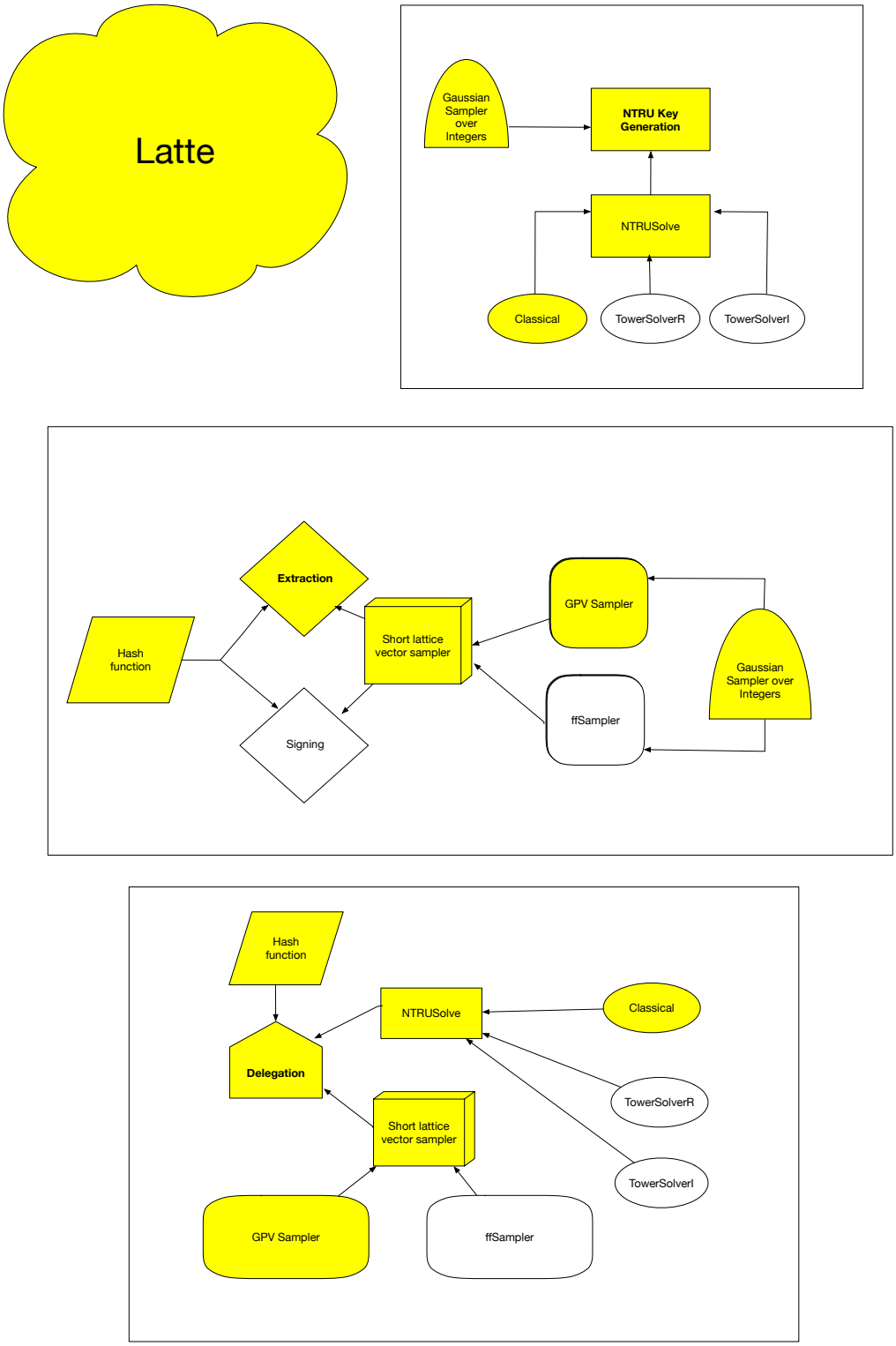


Fig. B.4 The components of the LATTE HIBE scheme

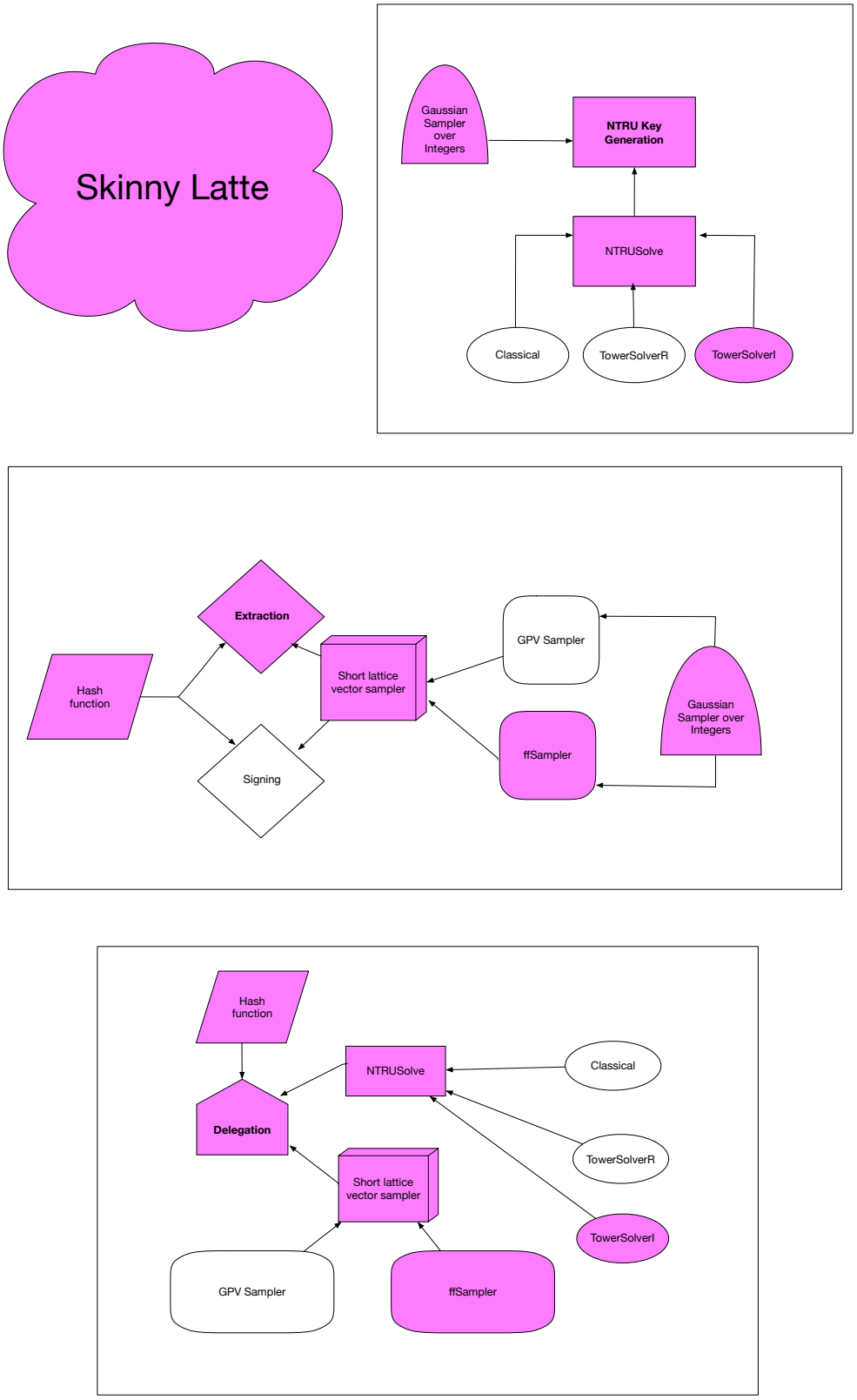


Fig. B.5 The components of the Skinny Latte HIBE scheme

APPENDIX C

Additional cryptographic algorithms

C.1 LWE PKE [Peikert et al., 2008]

Algorithm 36: LWE-based PKE Key Generation

Data: $n, m, q, l, t, r \in \mathbb{Z}$ and $\sigma > 0$
Result: $sk = \mathbf{S} \in \mathbb{Z}_q^{n \times l}, pk = (\mathbf{A}, \mathbf{B}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^{m \times l}$

1 $\mathbf{S} \leftarrow \mathcal{U}(\mathbb{Z}_q^{n \times l})$
2 $\mathbf{A} \leftarrow \mathcal{U}(\mathbb{Z}_q^{m \times n})$
3 $\mathbf{E} \leftarrow \mathcal{D}_\sigma(\mathbb{Z}_q^{m \times l})$
4 Set $\mathbf{B} = \mathbf{AS} + \mathbf{E}$
5 **return** $\mathbf{S}, (\mathbf{A}, \mathbf{B})$

Algorithm 37: LWE-based PKE Encryption

Data: Message $\mu \in \mathbb{Z}_t^l, pk = (\mathbf{A}, \mathbf{B}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^{m \times l}$
Result: Ciphertext $(\mathbf{u}, \mathbf{c}) \in (\mathbb{Z}_q^n, \mathbb{Z}_q^l)$

1 Choose $\mathbf{a} \leftarrow \mathcal{U}(\{-r, -r+1, ..., r\}^m)$
2 Compute $\mathbf{u} = \mathbf{A^T a}$
3 Compute $\mathbf{c} = \mathbf{B^T a} + \mu$
4 **return** $(\mathbf{u}, \mathbf{c})$

Algorithm 38: LWE-based PKE Decryption

Data: $(\mathbf{u}, \mathbf{c}) \in (\mathbb{Z}_q^n, \mathbb{Z}_q^l)$, $sk = \mathbf{S} \in \mathbb{Z}_q^{n \times l}$
Result: Decrypted message μ'
1 Compute $\mu' = \mathbf{c} - \mathbf{S^T u}$
2 **return** μ'

C.2 R-LWE DSS [Güneysu et al., 2012a]

Algorithm 39: GLP Key Generation

Data: $\mathcal{R}, n, p, k$
Result: $\mathbf{s_1}, \mathbf{s_2} \in \mathcal{R}_p^{2n}$, $\mathbf{t} \in \mathcal{R}_q^n$
1 $\mathbf{s_1}, \mathbf{s_2} \leftarrow \mathcal{U}(\mathcal{R}_p^{2n})$,
2 $\mathbf{a} \leftarrow \mathcal{U}(\mathcal{R}_p^n)$,
3 $\mathbf{t} = \mathbf{a}\mathbf{s_1} + \mathbf{s_2}$
4 **return**

Algorithm 40: GLP Sign

Data: $\mu, \mathbf{a}, \mathbf{s_1}, \mathbf{s_2} \in \mathcal{R}^{3n}$ *with coefficients in range* $[-1, 1]$,$H : \{0, 1\}^* \rightarrow D_{32}^n$, *where* D_{32}^n for $n \geq 512$ consists of all polynomials of degree $n - 1$ that have all zero coefficients except for at most 32 coefficients that are ± 1.
Result: $(\mathbf{z_1}, \mathbf{z_2}, \mathbf{c}) \in \mathcal{R}_p^{3n}$
1 Choose $\mathbf{y_1}, \mathbf{y_2} \leftarrow \mathcal{R}_p^n$ *with coefficients in range* $[-k, k]$
2 Compute $\mathbf{c} \leftarrow H(\mathbf{a}\mathbf{y_1} + \mathbf{y_2}, \mu)$
3 Compute $\mathbf{z_1} \leftarrow \mathbf{s_1}\mathbf{c} + \mathbf{y_1}$ and $\mathbf{z_2} \leftarrow \mathbf{s_2}\mathbf{c} + \mathbf{y_2}$
4 **if** $\mathbf{z_1}$ *or* $\mathbf{z_2} \notin \mathcal{R}^n$ *with coefficients in range* $[-(k - 32), (k - 32)]$ **then**
5 go to Step 1
6 **return** $(\mathbf{z_1}, \mathbf{z_2}, \mathbf{c})$

C.3 Kyber KEM [Schwabe et al., 2017]

In this section, the following notation is used.

Algorithm 41: GLP Verify

Data: $\mu, \mathbf{z_1}, \mathbf{z_2}, \mathbf{c}, \mathbf{a}, \mathbf{t}$
Result: ACCEPT or REJECT

```
if z_1 or z_2 ∉ R^n with coefficients in range [−(k − 32), (k − 32)] and
  c = H(az_1 + z_2 − tc, μ) then
    return Accept
else
    return Reject
```

Set $\mathcal{B} = \{0, ..., 255\}$, two hash functions $H : \mathcal{B}^* \to \mathcal{B}^{32}$ and $G : \mathcal{B}^* \to \mathcal{B}^{32} \times \mathcal{B}^{32}$, a key-derivation function $KDF : \mathcal{B}^* \to \mathcal{B}^*$, an extendable output function XOF and a pseudorandom function PRF. The notation $\hat{\mathbf{x}}$ indicates the vector $\mathbf{x}$ is in the NTT domain. Details of supporting algorithms $Parse, Encode, Decode, CBD, NTT$ are not given here, and compression/decompression techniques are not included and so the reader is referred to [Schwabe et al., 2017] for further information.

Algorithm 42: Kyber PKE Key Generation [Schwabe et al., 2017]

Data: n, q, k, η, d_u, d_v
Result: $sk \in \mathcal{B}^{12 \cdot k \cdot n/8}$, $pk \in \mathcal{B}^{12 \cdot k \cdot n/8+32}$

```
d ← B^32
(ρ, σ) := G(d)
N := 0
for i = 0 → k − 1 do
    for j = 0 → k − 1 do
        Â[i][j] ← Parse(XOF(σ, i, j) such that Â ∈ R_q^{k×k}
for i = 0 → k − 1 do
    s ← B^η(R_q^k)
for i = 0 → k − 1 do
    e ← B^η(R_q^k)
ŝ := NTT(s)
ê := NTT(e)
t̂ := Â · ŝ + ê
pk = Encode(t̂)
sk = Encode(ŝ)
return (pk, sk)
```

Algorithm 43: Kyber PKE Encrypt [Schwabe et al., 2017]

Data: $pk \in \mathcal{B}^{12 \cdot k \cdot n/8+32}$, $\mu \in \mathcal{B}^3 2$, $\mathbf{r} \in \mathcal{B}^3 2$
Result: $\mathbf{c} \in \mathcal{B}^{d_u \cdot k \cdot n/8 + d_v \cdot n/8}$

```
N := 0
t̂ = Decode(pk)
ρ = pk + 12 · k · n/8
for i = 0 → k − 1 do
    for j = 0 → k − 1 do
        Â[i][j] ← Parse(XOF(σ, i, j) such that Â ∈ R_q^{k×k}
for i = 0 → k − 1 do
    r ← B^η(R_q^k)
for i = 0 → k − 1 do
    e_1 ← B^η(R_q^k)
e_2 ← B^η(R_q^k)
r̂ := NTT(r)
u := NTT^{-1}(Â^T · r̂) + e_1
v := NTT^{-1}(t̂^T · r̂) + e_2 + Decode(μ)
c_1 := Encode(u)
c_2 := Encode(v)
return c = (c_1||c_2)
```

Algorithm 44: Kyber PKE Decrypt [Schwabe et al., 2017]

Data: $sk \in \mathcal{B}^{12 \cdot k \cdot n/8}$, $\mathbf{c} \in \mathcal{B}^{d_u \cdot k \cdot n/8 + d_v \cdot n/8}$
Result: $\mu \in \mathcal{B}^3 2$

```
u = Decode(c)
v = Decode(c + d_u · k · n/8)
ŝ = Decode(sk)
μ = Encode(v − NTT^{-1}(ŝ^T · NTT(u)))
return μ
```

Algorithm 45: Kyber KEM Key Generation [Schwabe et al., 2017]

Data: n, q, k, η, d_u, d_v, *Hash function* H
Result: $sk \in \mathcal{B}^{24 \cdot k \cdot n/8+96}$, $pk \in \mathcal{B}^{12 \cdot k \cdot n/8+32}$

```
z ← B^32
(pk, sk') ← KyberPKEkeygen()
sk = (sk'||pk||H(pk)||z)
return (pk, sk)
```

Algorithm 46: Kyber Encapsulation [Schwabe et al., 2017]

Data: $pk \in \mathcal{B}^{12 \cdot k \cdot n/8+32}$
Result: $\mathbf{c} \in \mathcal{B}^{d_u \cdot k \cdot n/8+d_v \cdot n/8}$, *shared key* $K \in \mathcal{B}^*$

```
1 μ ← B^32
2 μ ← H(m)
3 (K̄, r) := G(μ||H(pk))
4 c ← KyberPKEencrypt(pk, μ, r)
5 K := KDF(K̄||H(c))
6 return (c, K)
```

Algorithm 47: Kyber Decapsulation [Schwabe et al., 2017]

Data: $\mathbf{c} \in \mathcal{B}^{d_u \cdot k \cdot n/8+d_v \cdot n/8}$, $sk \in \mathcal{B}^{12 \cdot k \cdot n/8}$
Result: $K \in \mathcal{B}^*$

```
1  pk := sk + 12 · k · n/8
2  h := sk + 24 · k · n/8 + 32 ∈ B^32
3  z := sk + 24 · k · n/8 + 64
4  μ' := KyberPKEdecrypt(s, (u, v))
5  (K', r') := G(μ'||h))
6  c' = KyberPKEencrypt(pk, μ', r')
7  if c = c' then
8      K := KDF(K̄'||H(c))
9  else
10     K := KDF(z||H(c))
11 return K
```

APPENDIX D

Author's Publications and Contributions

1. S. McCarthy, M. Groves, P. Campbell. Advanced Lattice-based cryptographic primitives. Secure Architectures of Future Emerging Cryptography (SAFEcrypto) *Springer* M. O'Neill and T. Güneysu and F. Regazzoni and A. Waller. 2020. (Wrote chapter based on IBE and HIBE research and results)

2. N. Smyth, F. Regazzoni, S. McCarthy. Software Requirements and Architecture for Lattice Based Cryptography. Secure Architectures of Future Emerging Cryptography (SAFEcrypto) *Springer* M. O'Neill and T. Güneysu and F. Regazzoni and A. Waller. 2020. (Contributed to sections about the structure of the libSAFEcrypto software library which the author cowrote)

3. S. McCarthy and J.Howe and N. Smyth and S. Brannigan and Máire O'Neill. BEARZ Attack FALCON: Implementation Attacks with Countermeasures on the FALCON signature scheme *The 16th International Conference on Security and Cryptography (SECRYPT 2019).* 2019. (Sole creator of attack. Collected data for attack post-processing and proposed and implemented scheme and counter-measures)

4. A. Khalid and S. McCarthy and M. O'Neill and W. Liu. Lattice-based Cryptography for IoT in A Quantum World: Are We Ready? *IEEE International Workshop on Advances in Sensors and Interfaces.* 2019. (Wrote section on advanced cryptographic primitives)

5. S. McCarthy and N. Smyth and E. O'Sullivan. A Practical Implementation of Identity-Based Encryption over NTRU Lattices. *16th IMA International Conference on Cryptography and Coding.* 2017. (Proposed algorithmic optimisations and created initial implementation of IBE scheme. Performed parameter security analysis)