



**Research on Electronic Travel Document Protocols using Lattice-based Cryptography**

A Thesis

Presented to

The Graduate Faculty

University of Wisconsin-Platteville.

In Partial Fulfillment

Of the Requirement for the Degree

Master of Science (M. Sc) in Computer Science .

By

Chinmay Shah

2015

**1<sup>st</sup> Supervisor:** Prof. Dr. Harald Baier

**2<sup>nd</sup> Supervisor:** Prof. Dr. Omar Meqdadi

**3<sup>rd</sup> Supervisor:** Prof. Dr. Christoph Wentzel

The attached thesis proposal, by \_Chinmay shah\_, entitled \_Research on electronic travel document protocols using lattice-based cryptography\_, when completed, is to be submitted to the Graduate Faculty of the University of Wisconsin-Platteville in partial fulfillment of the requirements for the \_\_\_\_\_Masters in Computer Science\_\_\_\_\_ degree, for which \_30\_ credits shall be allowed is hereby:

Approved: \_ Harald Baier\_\_\_\_\_ Date: \_\_11/20/2015\_\_\_\_\_

Committee Chair: Prof. Dr. Harald Baier

Approved: \_ Nicolas Buchmann \_\_\_\_\_ Date: \_\_11/20/2015\_\_\_\_\_

Committee Member: Nicolas Buchmann

Approved: \_\_ Omar Meqdadi \_\_\_\_\_ Date: \_\_11/30/2015\_\_\_\_\_

Committee Member: Prof. Dr. Omar Meqdadi

## Research on Electronic Travel Document Protocols using Lattice-based Cryptography

Approved: \_ Harald Baier\_\_\_\_\_ Date: \_\_11/20/2015\_\_\_\_\_

Committee Chair: Prof. Dr. Harald Baier

Approved: \_ Nicolas Buchmann \_\_\_\_\_ Date: \_\_11/20/2015\_\_\_\_\_

Committee Member: Nicolas Buchmann

Approved: \_\_ Omar Meqdadi \_\_\_\_\_ Date: \_\_11/30/2015\_\_\_\_\_

Committee Member: Prof. Dr. Omar Meqdadi

Suggested content descriptor keywords:

\_Quantum Cryptography,

Eletronic travel documents\_\_\_\_\_

# Abstract

---

## Statement of the Problem

The world has become one globalized platform with the advent of sophisticated embedded systems as it allows people to communicate in the local town to the farthest town on this earth. This includes communicating via mobile phones, Internet or sending an instruction to microwave oven, television and other wireless systems. We make assumptions that our communication remains confidential when we communicate using mobile phones or Internet. We assume that the payment gateway will be authenticated and our banking details will remain confidential, while accessing Internet banking. We assume that our instruction sent to the wireless embedded systems will be received exactly as we had sent (Integrity) i.e. the message sent by us and the message received by the embedded system will be the same. We make these assumptions as there will be certain protocols followed to maintain security standards.

The security of these protocols is generally based on certain computationally 'hard' problems such as factoring a very large number. These are computationally hard problems to solve due to limited computational resources. If a quantum computer will be built, then these problems will no longer remain computationally hard. They will be solved in polynomial time complexity. Hence, our messages will remain neither confidential nor have integrity. Network communication will, no longer be able to authenticate. Similarly, electronic travel documents will not remain secured. Today, travel documents are an integral part of the daily routine. Among these travel documents, passports allow the holder's identity to be verified. This protects the nations against illegitimate immigrants, criminals and terrorists. Similar to the above mentioned scenario, advent of quantum computers will allow forfeiting of passports. It will also allow access to personal information. Hence, posing risk to millions of lives.

## Methods and Procedures

To conquer these issues, we propose to conduct research on replacing the current travel document protocols by using Lattice-based cryptography. This could be replicated in all other network based embedded systems. We are motivated to use Latticebased cryptography as it is one of the important classes of the post-quantum cryptography algorithms. Lattice based algorithms are important due to significant research by scientific community, well-known underlying lattice problems and computational efficiency.

Hence, we understand the cryptographic primitives behind the algorithms based on well-known hard lattice problems. We have created an exhaustive list of algorithms based on these cryptographic primitives. Learning with errors, NTRU, GGH, Ideal lattice based cryptosystem algorithms are the main candidates. Hence, we shortlist the lattice algorithms that will offer enough computational security, memory and time efficiency. After understanding the travel documents protocols and requirements, we evaluate the lattice based cryptosystems with respect to meeting these requirements of travel documents.

## **Summary of Results**

The main result is to ensure that the evaluated algorithms offer computational security, at least, as much as the current travel document algorithms with minimum changes in the existing infrastructure. In our main result, we also ensure that our proposed protocols can be implemented with minimum changes. In our result, we also evaluate key exchange mechanism and propose a new key exchange mechanism. Our research also led us to develop a new protocol that can reduce the need for large public keys in protocols. The core result of this thesis is to build protocols, which are time and memory efficient for embedded systems that would be quantum attack resistant, which would ensure network security.

# Preface

---

This thesis was prepared at the department of Informatics at the Hochschule Darmstadt and University of Wisconsin, Platteville, while working at the Center for Advanced Security Research Darmstadt (CASED) in fulfilment of the requirements for acquiring a M.Sc. in Computer Science

# Acknowledgements

---

I would like to thank Prof. Dr. Harald Baier for inviting me to do my thesis at the Center for Advanced Security Research Darmstadt (CASED). I am very grateful for inspiring and motivating me all the way through my stay there. I would like to thank Prof. Nicolas Buchmann for supervising my project at the Hochschule Darmstadt. I would also like to thank prof. Dr. Omar Meqdadi to help me with the thesis and registration for the dual master's degree in the US.

I want to express my deepest gratitude to Prof. Nicolas Buchmann and Prof. Dr. Harald Baier and Prof. Dr. Meqdadi for all the meetings, discussions and support throughout his supervision on my thesis progress. His precious suggestions and ideas have encouraged me to keep on working energized and overcome the emerging problems far earlier before they become apparent.

Last but not least, I want to thank my family for encouraging and supporting me throughout my MSc studies even though struggling financially and struggling with life taking diseases. During these two years they have always been there for me, no matter what, so they deserve to be acknowledged.

---

## Contents

### 1 Introduction

|       |                                                      |    |
|-------|------------------------------------------------------|----|
| 1.1   | Motivation.....                                      | 11 |
| 1.1.1 | Motivation of protecting travel documents.....       | 11 |
| 1.1.2 | Motivation in using lattice based cryptography ..... | 12 |
| 1.2   | Objectives .....                                     | 16 |
| 1.3   | Open Problems.....                                   | 17 |
| 1.4   | Structure of the Thesis.....                         | 17 |

### 2 Foundations of Lattice-based Cryptosystems

|       |                                                     |    |
|-------|-----------------------------------------------------|----|
| 2.1   | Quantum algorithms to break current schemes.....    | 19 |
| 2.2   | Quantum resistant classes of Cryptography.....      | 20 |
| 2.2.1 | Hash-based cryptography.....                        | 20 |
| 2.2.2 | Multivariate-quadratic-equations cryptography ..... | 20 |
| 2.2.3 | Code-Based Cryptography.....                        | 21 |
| 2.3   | Lattice and solving lattice problems.....           | 21 |
| 2.3.1 | Algorithms to solve lattice problems.....           | 21 |
| 2.4   | Lattice problems.....                               | 22 |
| 2.5   | Lattice cryptographic primitives .....              | 24 |
| 2.5.1 | Short Integer Solution.....                         | 25 |
| 2.5.2 | Learning with Errors.....                           | 27 |
| 2.5.3 | Efficiency of these functions .....                 | 29 |
| 2.6   | Efficient Key exchange mechanism .....              | 29 |
| 2.7   | Lattice cryptographic algorithms .....              | 31 |
| 2.7.1 | GGH/HNF public key cryptosystem .....               | 32 |
| 2.7.2 | NTRU Public key cryptosystem.....                   | 33 |
| 2.7.3 | Ajtai-Dwork Cryptosystem.....                       | 35 |
| 2.7.4 | Other Lattice based cryptosystems.....              | 35 |

### 3 Foundations of Electronic Travel document

|       |                                                          |                                     |
|-------|----------------------------------------------------------|-------------------------------------|
| 3.1   | Travel documents .....                                   | 37                                  |
| 3.2   | Travel documents .....                                   | 38                                  |
| 3.2.1 | Face Image .....                                         | 38                                  |
| 3.2.2 | Fingerprint.....                                         | 39                                  |
| 3.2.3 | Palmprint.....                                           | 39                                  |
| 3.2.4 | IRIS recognition .....                                   | 39                                  |
| 3.2.5 | Operations .....                                         | 40                                  |
| 3.2.6 | Security Goals .....                                     | 40                                  |
| 3.2.7 | Passport Life cycle .....                                | <b>Error! Bookmark not defined.</b> |
| 3.3   | Public Key Infrastructure.....                           | 41                                  |
| 3.3.1 | Certification Authority (CA).....                        | 41                                  |
| 3.3.2 | Registration Authority (RA) .....                        | 41                                  |
| 3.3.3 | Repository .....                                         | 41                                  |
| 3.3.4 | Public Key Certificates .....                            | 42                                  |
| 3.3.5 | Certificate Revocation List (CRL) .....                  | 42                                  |
| 3.3.6 | PKI Users .....                                          | 42                                  |
| 3.4   | Travel documents protocols .....                         | 43                                  |
| 3.4.1 | Passive Authentication .....                             | 43                                  |
| 3.4.2 | Active Authentication.....                               | 43                                  |
| 3.4.3 | Basic Access Control .....                               | 44                                  |
| 3.4.4 | Second Generation ePassports .....                       | 45                                  |
| 3.4.5 | Third Generation ePassports .....                        | 47                                  |
| 3.5   | Travel documents memory constraints and efficiency ..... | 48                                  |
| 3.5.1 | Memory constraints .....                                 | 49                                  |
| 3.5.2 | Operations and Efficiency.....                           | 49                                  |

## 4 Efficient Lattice based Algorithms

|       |                                                            |    |
|-------|------------------------------------------------------------|----|
| 4.1   | Efficiencies of the cryptosystems .....                    | 51 |
| 4.1.1 | Small Integer problem One Time Digital signatures.....     | 51 |
| 4.1.2 | Learning with errors symmetric encryption.....             | 52 |
| 4.1.3 | Learning with errors asymmetric encryption .....           | 54 |
| 4.1.4 | GGH/HNF public key cryptosystem- Hermite Normal form ..... | 55 |
| 4.1.5 | GGH/HNF public key cryptosystem- Optimized scheme .....    | 56 |

|                                                                           |    |
|---------------------------------------------------------------------------|----|
| 4.1.6 NTRU public key cryptosystem .....                                  | 61 |
| 4.1.7 Ajtai-Dwork Cryptosystem .....                                      | 63 |
| 4.1.8 Ideal lattices public cryptosystem .....                            | 63 |
| 4.2 Efficiencies of the key exchange schemes.....                         | 64 |
| 4.2.1 Our Proposed key exchange scheme .....                              | 64 |
| 4.2.2 Efficient Key exchange mechanism .....                              | 64 |
| 4.3 Efficiencies of other cryptosystems .....                             | 68 |
| <br>5 Proposed Protocols for Travel Documents                             |    |
| 5.1 Elected cryptosystems-public schemes .....                            | 69 |
| 5.1.1 Learning with Errors asymmetric scheme .....                        | 69 |
| 5.1.2 GGH scheme .....                                                    | 70 |
| 5.1.3 NTRU .....                                                          | 71 |
| 5.1.4 Ideal lattice Cryptosystem.....                                     | 71 |
| 5.2 Proposed Protocols in the travel documents .....                      | 73 |
| 5.2.1 Proposed Public key Infrastructure .....                            | 73 |
| 5.2.2 Proposal for Passive authentication.....                            | 74 |
| 5.2.3 Proposed Active authentication (optional protocol).....             | 74 |
| 5.2.4 Proposed Basic Access Control.....                                  | 74 |
| 5.2.5 Proposed Chip Authentication (mandatory EAC specification) .....    | 76 |
| 5.2.6 Proposed Terminal Authentication.....                               | 77 |
| 5.2.7 Proposed Password Authenticated Connection Establishment(PACE)..... | 77 |
| 5.3 Proposed Usage of Asymmetric cryptosystem .....                       | 78 |
| <br>6 Conclusion .....                                                    | 78 |
| Index of Figures.....                                                     | 84 |
| Index of Tables.....                                                      | 84 |
| Glossary and Abbreviations.....                                           | 85 |
| References .....                                                          | 83 |

# Chapter 1

## Introduction

---

This chapter provides an overview of the motivation, the objectives, open problems and the structure of the thesis

### 1.1 Motivation

If a perfectly secured algorithm is used then the decryption/private key for the scheme needs to be random (i.e. uniform probability key distribution and probability for each of the bits in the key to be 0 is 0.5) and as long as the message. It is inefficient in practice due to the long key size. Hence, computational secured schemes are used in practical scenario, which ensure that the schemes implemented are based on the problems, which are hard to solve in polynomial time, today and if they are solved, the private/decryption key is obtained and the security provided by the algorithm is lost.

Most common algorithms like RSA and Diffie Hellman key exchange scheme are computationally secured schemes and are based on the hard problem of factorizing a large number and solving the discrete logarithm problem respectively but with the invention of Shor's algorithm, it would solve both these hard problems in polynomial time using quantum computers, if built. A quantum computer making use of large qubit registers can put the RSA and DH algorithms out of practice as theoretically, Shor's algorithm with large number of quantum gates of quantum computer can solve the factorization problem in  $\lg N^3$  time approximately which is a polynomial factor [BCDP '96] This would put the entire internet domain at risk along with usage of travel documents at major risk. There are important classes of algorithms which are considered to be quantum attack resistant based on the underlying problems and Lattice-Based cryptography is one of the important classes. Hence there are two motivations for this thesis; one is with respect to protecting the travel documents integrity and the other one being the selection of lattice-based cryptography among the classes of algorithms that are supposed to be post-quantum attacks resistant.

#### 1.1.1 Motivation of protecting travel documents

The most commonly used travel document is passport which has multiple purposes. The electronic passports may hold biometric information like finger prints and encoded eye details (retina scan) along with personal information. There are several protocols in the passport, which have their own motivations. We are describing passive authentication, which is one of the protocols along with how it can easily be broken by quantum-attacks

Passive authentication checks for authenticity and integrity which is usually achieved by the digital signature. When the passport is read by the terminal, it reads the security object (hashed data groups and the signature), retrieves the Certificate from the signer and verifies the signature after computing the hash values of all the data groups. It mainly uses RSA and DSA algorithms. It is similar to the concept of integrity and authentication provided by the digital signatures.

By quantum attacks in existing scenario, the private key can be computed from the public key by solving factorization problem and fake hashes can be signed using this private key and when the terminal will obtain this signature and verify with the fake hashes computed or obtained, verification process would be completed. Hence, a chip with fake data groups and fake hashes will not be detected.

The scenario can be extended to other protocols of the passport. Quantum attacks on existing passport protocols will not ensure authentication, integrity, confidentiality, terminal and chip authentication. Hence, travel documents need to be quantum attack resistant and that would mean to use one of the classes of post-quantum cryptography algorithms. Hence, in the next section, we have described as to why we chose lattice-based cryptography among the whole set of classes of post-quantum cryptosystems.

### 1.1.2 Motivation in using lattice based cryptography

Depending on the current research of cryptanalysis using quantum computers, important classes of cryptographic systems are not solved because of the underlying hardness problem of these classes of algorithms. Among these classes, the most interesting is using lattice-based cryptography because there have been quite some progresses when compared to the other classes of post-quantum algorithms, which are explained in the following paragraphs along with certain features of the lattice cryptography which involves, simple computations, provably secure, based on well-known lattice problems and the security is based on worst case problem rather than the average case in RSA (A factorization algorithm working just on few values of  $N$  can be coded but such an algorithm cannot be written to solve lattice problems), which is also explained below as the underlying problems are different as it would be explained in the later chapters

#### Algorithms for solving lattice problems

There have been no algorithms that solve the lattice problem in polynomial time. There have been certain experiments and it was discovered that for small dimensions, the algorithms seem to have a non-negligible probability of outputting the shortest vector as shown in figure 1. Also, Dr. Gama and Dr. Nguyen have mentioned in their paper that “The best achievable approximation with approximation is close to  $(1.012)^n$  and the approximate hermit factor, which is needed to solve lattice problems is of  $(1.01)^n$  and is outside reach of known lattice reduction problem”. [GN ‘08] Figure 2 shows the comparison of time taken by different algorithms even below the dimensions of 55. Moreover, it has also been shown that the SVP in lattices with particular norms is NP-hard for randomized reductions of lattices.

Hence we can say that there is no polynomial time algorithm that approximates lattice problems to within polynomial factors.

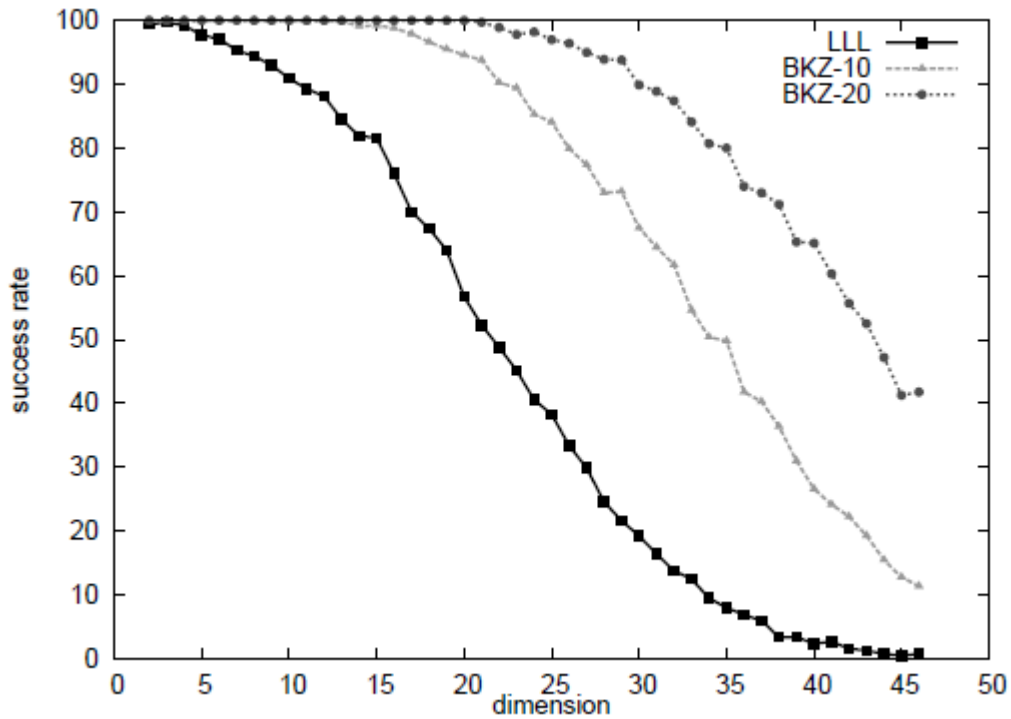


Figure 1: probabilistic SVP-oracles in lower dimensions [GN '08]

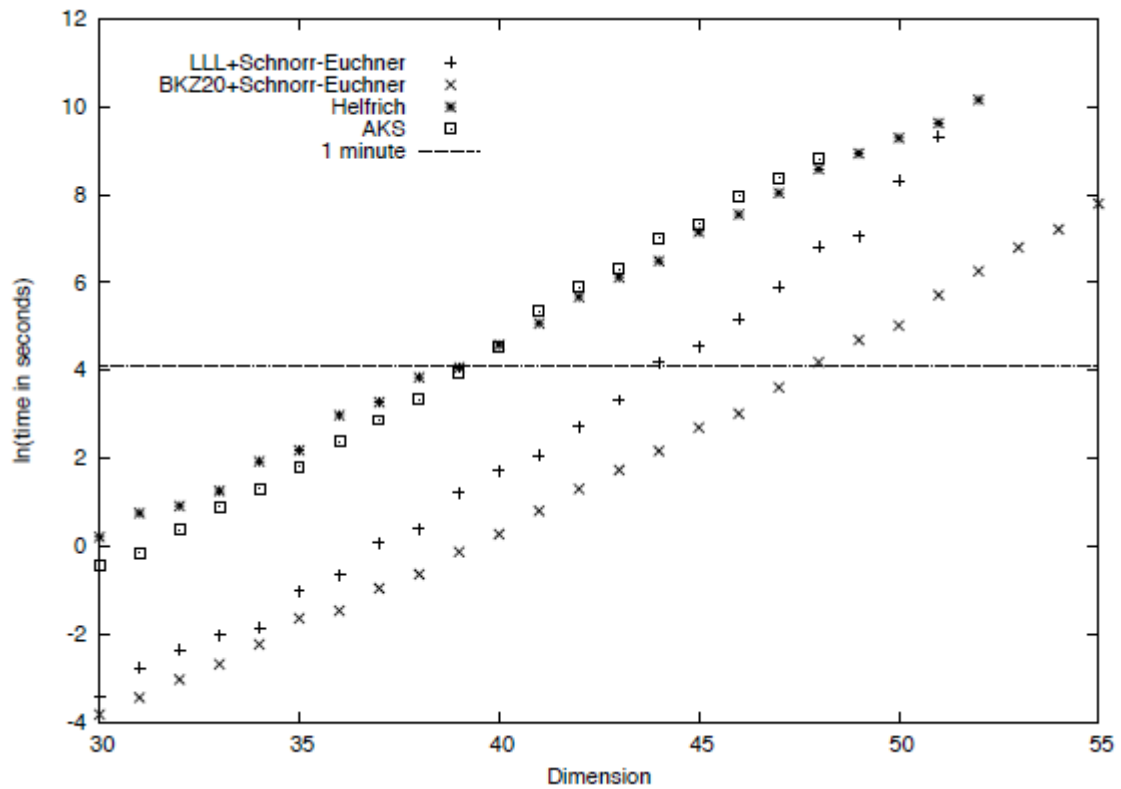


Figure 2: Comparison of efficiency of the algorithms [NV '11]

## Quantum algorithms to solve lattice-problems

There are currently no quantum algorithms, which can readily break the lattice-based problems with sufficiently high number of dimensions. The fastest known quantum algorithm Quantum search reduction algorithm approximates shortest lattice vectors up to a factor  $(k/6)^{n=2k}$  and makes use of Grover's quantum search algorithm, where  $n$  is the lattice dimension and  $k$  is an almost arbitrary parameter. The expected running time of the algorithm is  $O(n^3(k/6)^{k/8} + n^4)$  which is roughly the square root of the running time of the fastest known classical algorithm, which is approximately the complexity of  $O(n/2)^{n/16}$ .

The security analysis was conducted on the most famous lattice cryptosystem NTRU by Dr. Ludwig and it was found that an attack against NTRU-503 with algorithm required only estimated  $5.8 \cdot 10^3$  MIPS-years. Dr. Ludwig also found that "An attack with QSR against NTRU would be infeasible only if NTRU's security parameter was raised up to at least 1277, if sufficiently large quantum computer were available nowadays" [LC '03].

As mentioned above, Shor's factoring algorithm and related quantum algorithms make use of periodicity finding technique which does not really apply to lattice problems. Even though few connections were demonstrated as mentioned above in the use of Quantum search algorithm, no quantum algorithms are demonstrated to have shown any substantial result.

Hence, it can be said that there is no polynomial time quantum algorithm that solves the lattice problem with sufficiently high dimensions and appropriate approximation factor in polynomial factors.

## Efficiency vs provably security

In terms of security, lattice based cryptosystems are divided into two parts a) the ones which are really efficient but they lack the supporting proof of security. The second type admits strong provable security guarantees based on worst case hardness. The core goal is to have an efficient system which also has the computational security at least as much as the currently used algorithms in the travel documents along with efficiency.

The worst case security ensures that breaking a system is at least as hard as solving several lattice problems in polynomial time within appropriate approximations in the worst case. In case of most other systems for ex: in RSA, the factorization problem is at least as hard as the RSA problem so one can come up with an algorithm that solves RSA problem ( $e$ -th root of the cipher mod  $N$ ) when particular public key is used but does not solve the factorization problem for all the numbers. In RSA, it might also be possible to factor numbers chosen according to a certain distribution but does not factorize all numbers as it is based on the average-case hardness. Worst case hardness ensures durability of the cryptosystem and it helps us in choosing concrete parameters for the cryptosystem.

## Lattice-Based Cryptosystems

At this point from the above paragraphs, we have given a proof of how there are no quantum or classical algorithms to break the lattice problems in polynomial time but we need to understand if we can use these problems to form the cryptographic primitives and practically implement the lattice-based cryptosystems.

It is said that lattice-based cryptography has seen huge progresses compared to other classes of post-quantum cryptography and the best example can be as of Gentry's algorithm, which is the first ever fully homomorphic algorithm based on lattice-based cryptography. A fully homomorphism algorithm allows computation of both addition and multiplication on encrypted text and the decryption after the computation would be same as the plaintext after the computation. This would allow the third party to compute on data without knowing the data itself. The algorithm is based on lattice problems and the subset problem, where, the goal is to find a subset of the bases that sum to a target  $T \bmod M$  where the inputs are  $N$  bases and the target  $T$ . Clearly, the hardness depends on  $N$  and  $M$  as  $M$  forms the integer ring over group  $Z_M$ . There has also been significant progress in the areas of cryptosystems and digital signature schemes.

One of the famous schemes called the NTRU, which allows encryption and signing data is based on lattice-based cryptography, is based on lattice problems, which is significantly faster than the RSA and has small key sizes but does not offer the security of proof.

Learning by errors is a scheme that was discovered based on the parity problem and it was proved to be provably secured but there were large public keys making the system quite inefficient. Parity problem has the goal to calculate the function  $f(x)$  given some samples and the function  $f$  computes the parity at some locations. There has been research to have provably security with smaller key sizes and the RING-learning with errors have been evolved, which is based on the hard problem of subset sum and the lattice based problems. The problem in simple words can be defined as the input of  $g, t$  such that  $t = gs + e$  and  $s$  and  $e$  have small coefficients and  $s$  and  $e$  must be found. The multiplication and addition is to ensure that there is a uniformity of the linearly independent vectors. The computations in lattice-based cryptography are simple can be shown by this example: The encryption in ring-LWE is in the following steps

- 1) The message is converted into binary code
- 2) Pick random  $r$  from the Ring with small coefficients
- 3) Output  $(v = rg + e_2, w = rt + e_3 + sm)$  This step ensures uniformity and when the dimensions are quite high, ensuring that cryptanalysis finds uniformly random rows and columns all over.  $rg$  can be row matrix multiplication, column matrix multiplication or a hybrid matrix multiplication over very high dimensions

The decryption works as  $w - vs$  is equal to the binary code of the message and  $s$ , a very small coefficient chosen from the ring is known only to the receiver. The algorithms mentioned above are some of the candidates, which we need to prove that it matches the need of the travel documents with the constraints of high efficiency and low memory.

By describing the algorithm in number of steps, we want to show that systems based on lattice based cryptography involve simple computations and there have been significant progresses in developing practical schemes unlike the other classes of post-quantum cryptography.

## ***1.2 Objectives***

In the previous section, the motivation of this thesis has been explained along with the motivation for replacing the current algorithmic schemes in travel documents and the motivation for using lattice-based cryptography. This section outlines the concrete objectives of this thesis.

One of the main objectives is to thoroughly research the lattice-based cryptography to understand their requirements. The lattice-based problems and the cryptographic primitives behind the lattice-based cryptosystems also need to be well-understood. This would enable us to understand the advantages and draw backs of particular lattice-based cryptosystems. This would allow us to have a sample of algorithms as candidates, which would have computational security at least as much as the current travel documents algorithms

Another important objective is to test the efficiency and memory requirements of these algorithms. The requirements for the protocols in the passport need to be well understood with respect to the efficiency needs and the memory, which can be supported. Each of the candidates needs to be tested with respect to these constraints of speed and memory. We must also mention how the precise algorithm that has the desired computation security, efficiency and memory needs, can be implemented in the current protocols of the travel documents.

Furthermore, we also need to research a key exchange mechanism based on the hardness of lattice cryptosystem that would allow setting up a communication channel to ensure confidentiality as it is a desired feature in the travel documents for symmetric encryption as it would allow communication between terminal and the travel documents. We must also ensure that the algorithm, we selected offers digital signatures and cryptography else we need to look for a digital signature, which has the hardness based on the lattice-based cryptography and offers desired efficiency, computation security and matches with our memory constraints.

This would allow us to have the complete research document of how lattice-based cryptosystem can be used to replace the current travel document protocol algorithms and yet maintain the security of the same along with efficiency. Minimum changes in the existing public key infrastructure must also be ensured during the research. The protocols at the same time should stay practical using the lattice-based scheme. The public key must also stay competitive enough when compared to the current schemes i.e. the public key length must stay in a practical limit. The requirements of each protocol of the travel document must also be met at the same time.

In Summary, the research document will do the whole requirement analysis, design and architecture for the travel documents using lattice-based cryptosystem that one just needs to

read this document and implement the new system for the travel documents and yet maintain all requirements of protocol and the system should maintain the same efficiency and match the memory constraints and yet, ensure quantum attack resistance.

### 1.3 Open Problems

The lattice cryptography is just computationally secured but should stay secured at least for the time being as perfectly secured scheme would need the use of uniformly random key length as long as the message, which would make it extremely inefficient. All the schemes being in use make use of computationally secured schemes. And we don't know for how long, it might stay as computationally secured scheme

As mentioned before, lattices problem solving algorithms make use of reduction techniques and since a lot of polynomial mathematics is used to speed up the computational procedures, there might be better reductions or reductions to other problems which makes lattices easier to solve in the future as we might be looking in the wrong direction but we still have to consider that there is no algorithm in quantum or classical system that solves lattice problems in polynomial time with appropriate approximation (around  $1.01^n$ ) with sufficiently high dimension  $n$ .

Even for a non-perfectly secured lattice-based cryptosystem, the key length runs into large sizes.

### 1.4 Structure of the Thesis

Now that the motivation and the objectives in this thesis were presented, the remaining parts of this thesis are structured in the following way:

Chapter 2 shall describe the necessary fundamentals of the Lattice based cryptography, Lattice-based problems, underlying ideas in Lattice solving algorithms, cryptographic primitives and the lattice cryptosystems. A section also short lists the algorithms that are at least computationally secured as the current schemes implemented in the travel documents.

In Chapter 3, the fundamentals of the travel documents, protocols in the travel documents, the underlying requirement with efficiency and memory constraints will be generated.

Chapter 4 then would describe how the data to check the efficiency and memory constraints of each algorithm was generated. This includes the description of the chosen simulation environment as well. Key exchange algorithms and digital signatures must also be tested to match the requirements, which are also described in this chapter.

The following chapter 5 then provides the analysis of these algorithms and outlines with the conclusion of the possible features representing the results of this analysis. It also focuses on

the other requirements of the protocols in the travel documents and mentions about the feasibility test to ensure that there are minimum changes in the existing infrastructure of the travel document protocols, which is then followed by the necessary steps needed to implement the short listed algorithms for cryptosystems, key exchange and digital signatures.

In chapter 6, the design and architecture of the new infrastructure based on lattice-based cryptosystem is mentioned and a section is dedicated on how to replace the existing algorithms with the lattice-based algorithms. There are number of other issues with the existing travel documents that need to be mentioned in this chapter

The final chapter provides a short summary, outlines the contributions of this thesis and describes open points for further work in this field.

## Chapter 2

---

# Foundations of Lattice-based Cryptosystems

---

This chapter provides an overview of lattice-based cryptosystems as well as foundations of the underlying technologies of this thesis.

After an introduction to lattices and an associated history, the next topic covered by this chapter would be lattice-based problems. The underlying primitives would be followed next. The list of lattice-based cryptosystems would be the last topic of this chapter. This list should help us in determining the set of algorithms, which we should be looking at for the rest of the thesis. The small introduction would help in understanding the lattices followed with the cryptographic primitives, which lead to the development of the lattice-based cryptography.

## 2.1 Quantum algorithms to break current schemes

Shor's algorithm was published, which gave the solution to both factorizing large numbers and solving discrete log problem in polynomial time. The algorithm can be simply described as finding period in classical computers. The algorithm takes a number smaller  $A$  than  $N$  such that greatest common divisor  $\gcd(A, N) = 1$  and finds the period of  $N$  with respect to  $A$ , whose order  $R$  is computed by taking  $A^{(x+R) \bmod N} = A^{x \bmod N}$  and then  $\gcd((\text{square root of } (A^R + 1)), N)$  would lead to the factor of  $N$ . But it takes exponential time when using it via classical systems as in classical system, the probabilities are not dependent on the amplitudes but it is based on the occurrences but when Quantum bits are used which allows the system to perform in quantum states, the algorithm's working changes completely as in Quantum states, the probability is just the square of the amplitude and paths cancel out each other and all the paths must be considered known as superimposition. For Ex: The amplitude of  $-0.5$  and  $0.5$  would cancel out each other and if a needle is to be found in a hay then just follow all the paths and make dead ends disappear, which have negative interference. In theory, Shor's algorithm computes the period of the large composite number  $N$  (which is the product of two large primes) by evaluating functions at all points at the same time but this requires the system to calculate the functions in all Quantum states, which also needs the linear transformation of quantum bits (qubits) into the frequency domain and we obtain the period for the large number  $N$  and we need a computer which can maintain large number of Qubit states, which is the state not being in either of the two states ( $0$  or  $1$ ) as opposed to the classical system. Once the period has been obtained and factorization problem has been solved. Similarly, discrete logarithm problem can also be broken using Shor's algorithm. Hence, The quantum computer making use of large qubit registers can put the RSA and DH algorithms out of practice as theoretically, Shor's algorithm with large number of quantum gates of quantum computer can solve the factorization problem in  $\lg N^3$  time approximately which is a polynomial factor [BCDP'96].

The machine readable passports make use of RSA signature schemes and DH algorithm. Hence, the invention of a quantum computer would not ensure the protection of travel documents protocols. Also, Grover's algorithm, which involves the searching through the unsorted database with  $N$  entries, can be performed in square root of  $N$  allowing the AES 128 and triple DES to be computed in less than  $2^{64}$  time units, which is much faster than the current results. Hence, for well-known symmetric algorithms, the key length will have to be at least 200 bits, which is double than the current security standards and key sharing will become extremely hard [BBBV'97].

## 2.2 Quantum resistant classes of Cryptography

Quantum Cryptography is not into research just because Shor's Algorithm breaks the factorization and discrete logarithm problem putting RSA and DH algorithms useless, if quantum computers would be in practice. Classes of Quantum Cryptography can help in detection of eavesdropping, too. Grover's algorithm, a quantum search algorithm reduces the search space by a factor of square root than the classical searching algorithms. It is just to emphasize on the power of quantum algorithms. There have been certain classes of algorithms that have been said to have no existing classical or quantum algorithm to break the underlying problems upon which the cryptosystems could be built.

### Hash-based cryptography

Hash based cryptography is based on the usage of hash trees coupled with one time signature schemes. The binary hash tree can hold  $2^n$  leaves. The original idea was to allow the hash tree to concatenate both the child nodes to form the inner node. This would allow signing a limited number of messages with one public key even though a one-time signature scheme is used. But for it to become practical, it must be extendable to infinite leaf nodes to ensure the complete advantage of the system. But due to the advent of Shor's algorithm, one time signature schemes based on large hash trees has seen progress as the memory and efficiency of the systems have increased over the last few decades allowing to use store large hash trees and run one time signature schemes fast but there have been no great progresses, which can potentially replace current infrastructure.

### Multivariate-quadratic-equations Cryptography

This has been one of the most recently developed classes of algorithm. The underlying problem is to solve multivariate (many unknown variables) quadratic equations over the finite field (large quadratic tuples, cyclic rings, etc.) Certain set of such problems are classified as NP Complete problem (Every such problem can be solved by brute force in exponential time and when any of the NP-complete solution would be solved, all such problems would be solved) and no quantum algorithm is also able to break this underlying problem of solving the equation when in large dimension making it a candidate for post quantum cryptography.

Unfortunately, there have been no efficient cryptosystem developed but there have been signature schemes developed.

## Code-Based Cryptography

The underlying problem is the error correcting code problem. When the encoded data is transmitted by the sender, it is possible to correct some of the errors. The public key schemes make use of error-correcting code for which an efficient decoding algorithm is known (Usually linear codes from algebraic curves over finite fields are selected) and this error-correcting code would be the private key and public key is generated from the private key by disguising it as a large linear code.

Over all, Lattice-based cryptography has many advantages over these classes as practical efficient key encapsulation mechanism and Signature schemes have been established [PC '14] along with other advantages mentioned in the motivation of lattice based cryptography when compared to these classes of algorithms.

## Lattice and solving lattice problems

Micciancio and Regev describe a Lattice as 'A set of points in  $n$ -dimensional space with a periodic structure such as one described below. Given  $n$ -linearly independent vectors known as basis, a lattice is generated, which is the set of vectors'.

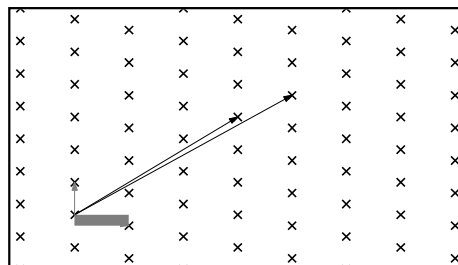


Figure 3: A lattice in  $\mathbb{R}^2$  and two of its bases [RO '14]

Lattices were usually used in the cryptanalysis but Atjai invented the Atjai seminal function [12], which speeds up constructions of the lattice based cryptosystems.

### 2.2.1 Algorithms to solve lattice problems

There are many algorithms that have attempted to solve lattice problem with in polynomial time but have not been able to do so. These algorithms can be divided into two main approaches for solving exact lattice problems: enumeration and sieving-reduction. Enumeration algorithms, given a lattice basis  $B$ , systematically explore a region of space (centered on the origin for SVP, or around the target vector for CVP) that is guaranteed to contain a nonzero

shortest lattice vector. The running time of these algorithms is roughly proportional to the number of lattice points in that region, which, in turn depends on the quality of the input basis.

LLL algorithm is a lattice reduction algorithm, which outputs the LLL-reduced lattice basis by following the lattice reduction method after performing the basis of the vector into orthonormal set i.e. they are orthogonal (independent of other vectors) and are unit vectors. We are yet to research more in this area. LLL could not solve the problem but could approximate the basis. Schorr's algorithm was presented to improve the approximation factor but it increased the run-time. But approximation does not really solve the problem as it must be close to exact. The most efficient algorithm known as the heuristic AKS sieve algorithm also takes  $2^{\text{complexity}(N)}$  to solve the SVP even with in approximation factors and the authors of this algorithm clearly mention that the algorithm needs exponential space not allowing the algorithm to solve SVP for more than  $N=50$  dimensions. There have been no other algorithms, which can solve the lattice problem with better results.

Dr. Voulgaris in his paper states "Using an LLL reduced basis, the enumeration algorithm achieves complexity of  $2^{O(n^2)}$  for both problems. This was improved to  $n^{O(n)}$  by Kannan using a basis reduction algorithm. A careful analysis of the algorithm further reduced the complexity to  $n^{0.184n}$  for SVP and  $n^{0.5n}$  for CVP. Kannan's algorithm remains the best algorithm for CVP and the best deterministic algorithm for SVP. The AKS Sieve lowers the running time complexity of SVP to a simple exponential function  $2^{O(n)}$  using randomization and  $2^{O(n)}$  space" [VP '11] .

Hence we can say that there is no polynomial time algorithm that approximates lattice problems to within polynomial factors.

## 2.3 Lattice problems

The lattice-based cryptographic constructions are based on presumed hardness of lattice problems; the most common accepted problem is the shortest vector problem (SVP). The input is the lattice with arbitrary  $n$ -linearly independent vectors (none of these vectors can be calculated by linear combination of other vectors) and the goal is to obtain the shortest nonzero vector in it. There are many variants of this problem. A variant can be of having goal of finding the nearest and shortest non-zero vector to the target vector  $t$ , which is also the input. There are several other well-known problems in the lattice. Closest vector problem is another problem, where the lattice with  $N$ -dimension is the input along with a target vector  $t$  and a radius  $r$  and the goal is to find lattice point closest to  $t$  with in the radius  $r$ .

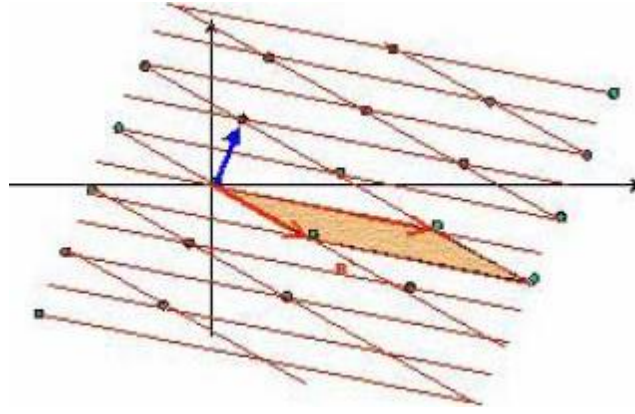


Figure 4: Shortest vector problem [3]

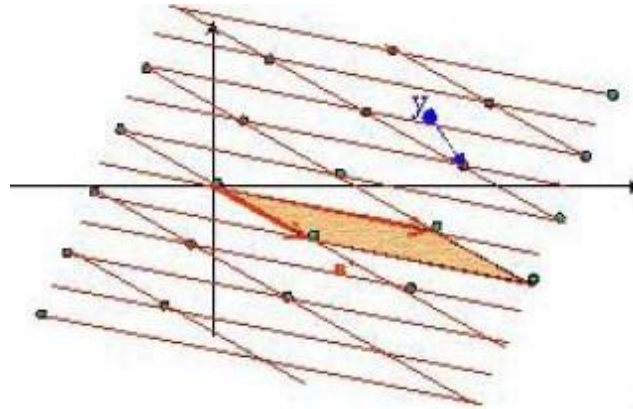


Figure 5: closest vector problem [3]

In ‘Complexity of Lattice problems’, Daniele Micciancio and Shafi Goldwasser have mentioned that no efficient algorithm has been found to solve these two lattice problems. When either of the problems is solved, the other problem would also be solved. When the target is the center point of the lattice, the CVP would be reduced to SVP when the shortest non zero vector in the whole lattice is to be discovered. Hence, a CVP oracle can answer for the SVP. It can be practically implemented by doubling one of the basis and passing that as the basis input and the target point being the original basis point, the output  $X$  being the CVP then the SVP would be  $X$ -the actual basis point and by adding a target point and the closest non zero vector nearest to it to SVP, CVP is a generalization of SVP.

Private Key cryptography and digital signatures mainly use Shortest independent vector problem and Absolute distance decoding problem. Shortest independent vector would mean to find the find  $n$  successive independent shortest non zero vector when lattice with basis from  $b_1$  to  $b_n$ . The problem can have a small variation that  $n$  independent shortest non zero vector have to calculated such that the  $n^{\text{th}}$  vector is smaller than  $\gamma \cdot (\text{nth successive minimal independent vector})$  where  $\gamma$  is the approximation factor greater than 1. Absolute distance problem, which is also used in code-based cryptography would mean to find a lattice point  $x$  given a target point  $t$  such that the distance between the  $x$  and  $t$  is less than  $\gamma \cdot \mu(L)$  where  $\mu(L)$  is the covering radius of the lattice, which is the maximum distance from any vector to the lattice.

Public key cryptography is built on the problems of GapSVP, GapSIVP and Bounded Distance decoding problem. GapSVP is used to compute or approximate the  $\lambda_1$ , which is the length of the shortest non zero vector in the lattice without finding the shortest vector. GapSIVP is used to compute or approximate the  $\lambda_n$ , which is the length of the  $n$ th shortest non zero vector in the lattice (where  $n$  is the number of dimensions in the lattice) without finding the  $n$ th minimum successive vector of the lattice. Bounded Distance decoding problem is a special case of closest vector problem, where the target  $t$  is at a distance which is at most  $2 * \lambda_1$  from the lattice and there exists at most one vector making it easier to solve than the general case of CVP.

[MG'01], [GMSS'99], [M'08], [L'87], [LM'09], [LLS'91] allows us to derive the relationship between these problems into three levels of the tree, where child node problem would also be solved when the parent problem is solved

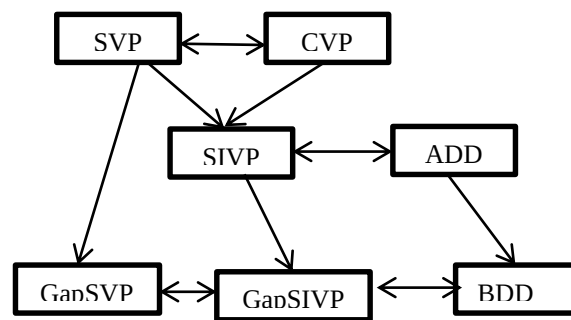


Figure 6: Relationship between Lattice problems

## 2.4 Lattice cryptographic primitives

In 1996, Atjaj discovered the Atjaj function, which lead to a renewed interests in the area of lattice cryptography. When there is an input  $X$  where  $X \in \{0, 1\}^m$  and key  $A \in \mathbb{Z}^{n \times m} \bmod q$  ( $\mathbb{Z}$  is the set of integers,  $m$  is the number of columns and  $n$  is the number of rows) and  $m > n \cdot \log q$  then  $X * A \bmod q = f_A(x)$  is a one way function. This discovery allows building many applications using this property. This function makes use of the subset sum problem. Subset problem assures that given a large set of  $A$  and  $T$  is the sum of a random subset of  $A$  then finding a subset of  $A$  that sums to  $T \bmod m$  is hard and the problem gets harder with increasing size of  $n$ .

It is well established that both ring-SIS and ring-LWE enjoy provable hardness guarantee. It is established that the average hardness is based on the idea that the Shortest Vector problem cannot be solved by quantum computers for the ideal lattices over the corresponding ring. The ideal lattice parameters and the proof is given by Atjaj. Hence, cryptosystems and key exchange primitives based on ring SIS and ring -LWE should be taken into consideration for meeting the goal of this thesis.

### 2.4.1 Short Integer Solution

PQCrypto Summer school had a conference held by the Institute for Quantum Computing based on which, this section is derived from the lecture of Daniele Micciancio's lattice based public key cryptography.

Parameters:  $m, n, q \in \mathbb{Z}$

Key:  $A \in \mathbb{Z}_q^{n \times m}$

Input:  $x \in \{0,1\}^m$

Output  $f_A(x) = Ax \bmod q$

From the above parameters, it can be inferred that a matrix  $A$  with  $n$  rows and  $m$  columns is the key and an input  $x$  has  $m$  columns and a single row and the output is obtained with multiplication of the matrix with the input and modulus  $q$  is taken to form a corresponding ring structure. Short Integer solution problem is based on the idea that given the output and key, it is hard to find the input  $x$ . It is hard to find as the underlying hardness is based on the subset sum problem i.e. given a target  $T$ , it is hard to find a subset of the set such that it sums to the target  $t \bmod m$ . It is based on subset sum problem as matrix multiplication over binary values is essentially an addition of a particular subset of the matrix. It can easily be implied that the hardness depends on the number of rows as that would make the set larger allowing more combinations of subsets obtaining the same value.

Properties

Compression:

The output  $f_A(x) = Ax \bmod q$  would be of the form  $n$  rows over  $\bmod q$ . Hence, an input of  $m$  bits columns would result into output of  $n \log_2 q$  bits ( $n * \log_2 q$  as it is a ring formation over  $q$  and each row could hold  $\log_2 q$  bits. Decimal number converted into the binary values) If  $n * \log_2 q$  bits is smaller than  $m$  bits then it is a compression function. Also, to solve this function, which is to obtain the input  $x$  (or the message in an application), the shortest independent vectors have to be found, which is the particular subset for the matrix  $A$ . It is equivalent of finding the shortest independent vectors. Hence, this function can be applied to cryptographic application of hashing that it is a one way function (as shown above that SIVP is hard to solve) and message  $m$  is equivalent to the input  $x$  (i.e. message is converted into binary values) and using the compression method (i.e. when  $m$  bits of message  $> n * \log_2 q$  with  $n$  as the security parameter), a hashing application can be built.

It can be shown that this application of hashing is collision resistant by the following:

Two different messages  $x$  and  $y$  with  $x \neq y$  cannot have the same hash values for collision resistance to be true. Assume that the function mentioned above is not collision resistant so there exists a message  $y$  such that  $f_A(x) = f_A(y)$ . Add this value to a random row of the key  $A$  to

obtain  $B = a + f_A(y)$ . Now hash the messages  $x$  and  $y$  over this new key to check for a collision. But  $A(x-y) \bmod q$  and  $B(x-y) \bmod q$  would be equal to  $f_A(y)$ , which is contradicting to the assumption that  $f_A(x) = f_A(y)$  as that would mean  $A(x-y) \bmod q = 0$  and  $B(x-y) \bmod q = 0$ . Hence, the hash function is collision resistant

Another property of SIS functions is the regularity property. When two messages  $m$  and  $y$  are such that  $x \neq y$  then  $f_A(x)$  and  $f_A(y)$  are independent over a random key  $A$  such that given either of the values, probability to guess the value of the other function over different message is zero. When the messages  $x, y$  are different, then  $f_A(x)$  gives no information about  $f_A(y)$  and vice versa establishing the pair wise independence. Left Over hash Lemma has establishes that pair wise independence and compression fulfils the property of regularity that if the input is chosen with a particular probability then the output of the function is also chosen with the same probability. For example, if input is uniformly distributed then the output is also distributed uniformly. Zero knowledge systems or perfectly hiding commitments could be built over regularity property i.e. neither of the parties gain any knowledge over the communication but one party shows to another that it has the knowledge of a particular event that was to happen. The analogy could be that a message is locked in a box with a key and the box is handed to the receiver and after the event, the key is given to the receiver to show that the event was in synchronization with the message.

The application can be implemented as follows:  $A$  and  $B$  are the key matrices. The message  $M$  and a random  $r$  is taken and converted into binary and split into  $m$  columns and the function  $f_A(M) = AM \bmod q + Br \bmod q$  is computed and there is an hiding property as  $Br \bmod q$  cannot be computed due to solving subset sum problem and whenever, the message is to be revealed to the receiver,  $r$  is send to the receiver and the appropriate computation to reveal the message would be performed

One of the most important properties of SIS is that it offers linear homomorphism

i.e.  $f_A(x) + f_A(y) = f_A(x+y)$  and  $f_A(x) + f_B(x) = f_{A+B}(x)$

Efficient one time Digital signature scheme, which is quantum resistant, can be built over SIS due to the above mentioned three properties. One time digital signature as the private key and public key would be chosen just for one message to ensure that the patterns for the key are not leaked.

Two small inputs  $X, x$  are chosen as the secret key.  $X \in \{0,1\}^{m \times m}$  and  $x \in \{0,1\}^{1 \times m}$ . The public key is chosen as  $f_A(X) = |f_A(X_1), \dots, f_A(X_m)|$  (each of the  $X_i$  is a matrix of  $1 \times m$  elements) and  $f_A(x)$ . The message  $M \in \{0,1\}^{1 \times m}$  is to be signed such that integrity, authenticity and non-repudiation is maintained. Signature algorithm would compute  $X * M + x$ .  $X * M$  would output  $m$  columns added with  $x$ . This would be the message  $y \in \{0,1\}^{1 \times m}$  for the SIS function.  $f_A(y)$  would be the one time digital signature for the message  $m$ .

The equation for the signature is simplified as follows:

$$f_A(y) = f_A(XM + x)$$

Due to the linear homomorphism, it is equal to

$$f_A(y) = f_A(X)M + f_A(x)$$

$f_A(X)$ ,  $f_A(x)$  are the public keys. Hence, the receiver just has to multiply the public key  $f_A(X)$  and the received message  $m$  and add the second public key  $f_A(x)$  to it. This digital signature is based on the fact that  $f_A(X)$  and  $f_A(x)$  are one way functions and as shown above, it is collision resistant so an adversary cannot select another private key to receive the same signature for an altered message. A small size for the private key is chosen to ensure to maintain the property of SIVP (shortest independent vectors) as finding  $n$  shortest vectors ( $x$  in  $f_A(x)$  are set of shortest vectors) in lattice is hard. This digital signature involves simple computation, efficient algorithm as SIS function involves matrix multiplication and addition and the public key  $f_A(X)$  and  $f_A(x)$  has the same length as of the message bits passed which is the column size  $m$ . As mentioned above, it is quantum resistant digital signature.

### 2.4.2 Learning with Errors

Parameters:  $m, n, q \in \mathbb{Z}$

Key:  $A \in \mathbb{Z}_q^{m \times n}$  ( $n$  is the number of columns for the matrix)

Input:  $s \in \mathbb{Z}_q^{1 \times n}$ ,  $e \in E^{m \times 1}$  where  $E$  is the error term with  $|E|$  terms

Output  $g_A(x) = (As + e) \bmod q$

Theorem (Regev' 05) states that it is hard to invert the function on the average assuming SIVP is hard to approximate in the worst case even for quantum computers. The main security parameter would be the number of columns  $n$  as that would make the  $A \cdot S$  complex and increase the entropy for the input  $s$ . The public parameter for Learning with Errors is similar to that of the SIS public parameter. The input  $s$  is a one row and  $n$  columns input making to  $n \cdot \log_2 q$  bits ( $n$  columns) and the error term making  $m \cdot \log_2 |E|$  bits ( $m$  rows) and the output  $g_A(x)$  would be  $m \cdot \log_2 q$  bits ( $m$  rows). This means that there is an expansion or compression for the output bits as the input of  $m \cdot \log_2 |E|$  bits is converted into an output of  $m \cdot \log_2 q$  bits. Theorem (Regev' 05) states that  $|E|$  should be greater than square root ( $n$ ). Theorem also states that  $|E|$  follows a Gaussian non uniform distribution pushing the function to be injective and an expansion function. The expansion factor is  $\log_2 q / \log_2 |E|$  (output bits / input bits).

Pseudo randomness is an important property offered by learning with error. When an efficient algorithm  $D$  cannot distinguish between a random input and output from our function, it can be classified as Pseudo Random. An efficient algorithm  $D$  will predict with exactly 50% probability that the next bit would be 0 or 1 for a truly random string. To break the learning with error function, the efficient algorithm must distinguish the pair  $(a_i, a_i \cdot s + e_i)$  and obtain the secret value  $s$ . The adversary to obtain the secret key will have to choose a random  $r$  and choose a secret value  $s$  with probability  $1/q$  ( $q$  possibilities for secret value  $s$ ) and obtain  $v = r \cdot s$  and when it is passed to an efficient algorithm with the pair  $(a_i + r, g_{a_i}(s, e_i) + v)$  and the efficient algorithm has to identify if it was a valid output from the function. Hence, based on this

property of pseudorandomness, a symmetric encryption can be built over Learning with error. The symmetric encryption must ensure that adversary cannot distinguish between zero's encryption and the encrypted text.

### Symmetric encryption

Secret key  $s \in \mathbb{Z}_q^{1 \times n}$ , message  $m$  is encoded into binary  $o$  and  $1$

Encryption algorithm:  $\text{Enc}(s, m) = \{a_i, c_i = g_{a_i}(s, e_i) + E(m)\}$  where  $E(m) = q/2 * m$

Decryption algorithm:  $\text{Dec}(s, \{a_i, c_i\}) = c_i - a_i * s = e_i + E(m)$ . For correction of the scheme, it is expected that  $e_i$  is less than  $q/4$ . Hence the next step is to subtract  $q/4$  to remove the error term  $e_i$ , since encrypted message could be  $o$  or  $q/2$ , it could very well be identified if the bit was  $0$  or  $1$ .

If  $g_a(s, e_i)$  were uniformly random, the encrypted text  $c$  would also be uniform and independent of  $m$ . If the message were as long as the number of columns  $n$ , it would provide unconditional security, which would mean that the secret key were as long as the message. Overall, the scheme offers security as it is computationally hard for adversary even with quantum computers to distinguish between encrypted text and truly random bits.

Important property of learning with errors is the homomorphic property, i.e.

$$g_{A1}(s, x_1) + g_{A2}(s, x_2) = g_{A1+A2}(s, x_1 + x_2)$$

Hence, this applies to the encrypted text in the following manner

$$\text{Enc}(s, m_1) + \text{Enc}(s, m_2) = \text{Enc}(s, m_1 + m_2)$$

$$\text{i.e. } (a_1, g_{A1}(s, e_1) + q/2 * m_1) + (a_2, g_{A2}(s, e_2) + q/2 * m_2)$$

$$= (a_1 + a_2, g_{A1+A2}(s, e_1 + e_2) + q/2 * (m_1 + m_2))$$

From the above mentioned logic of homomorphism, public key scheme could be built.

Private key: choose a private key  $s \in \mathbb{Z}_q^{1 \times n}$

Public key: Encrypt  $p_i = (s, 0)$  to make  $p_1 \dots p_n$ . That is equal to  $p_i = ((s * a_i + e_i \bmod q) + (0 * q/2))$

Encryption algorithm: Pick a random  $r_i$  and encrypt the message with the public key and add the message part, which would be as follows:

$$\sum_{i=0..n} \{p_i * r_i\} + q/2 * m = \sum \{r_i * \text{Enc}(s, 0)\} + q/2 * m \text{ (as each of the public key is encryption of 0)}$$

$$= \text{Enc}(s, \sum r_i * 0) + q/2 * m \text{ (using the homomorphic property as } \sum \{r_i * \text{Enc}(s, 0) + m\} \text{ would lead to } n \text{ times and the encryption term can be written by adding up the message values)}$$

$$= \text{Enc}(s, m)$$

Hence, encrypting using the public key mentioned above is same as encrypting with the secret key using the specific homomorphic property.

Decryption algorithm:  $\text{Dec}(s, \{a_i, c_i\}) = c_i - a_i * s = e_i + E(m)$  (i.e. same as the symmetric scheme algorithm as the secret key is known but the secret key is known only to the message receiver)

### 2.4.3 Efficiency of these functions

For the SIS function, if  $n=64$  (rows),  $q=2^8$  and  $m=1024$ (columns)

The function would be mapping 1024 bits of input into  $n * \log_2 q$  i.e. 512 bits

The Key size (Public parameter A) would be  $n * m$  (The whole matrix)  $* \log_2 q$  i.e.  $2^{19}=64$  KB in our example

The two Public key size for the digital signature would be needing  $n * \log_2 q * m$  (for  $f_A(X)$ ) +  $n * \log_2 q$  (for  $f_A(x)$ )

Runtime: The matrix multiplication of  $1 * m$  matrix with  $n * m$  matrix would result in  $n * m$ , which is equal to  $2^{16}$  in our case.

Hence, it is efficient in runtime but has very high space needs

The space needs can be reduced by number of ways. One idea could be use a structured matrix i.e. take the first row and the second row would be one bit left/ right circular of the first row and so on allowing just the first row of the matrix to be stored as the public key i.e. the space complexity from  $n * \log_2 q * m$  can be dropped down to  $m * \log_2 q$ . Hermite Normal form could also be used for the public key to reduce half of the size of the matrix i.e.  $n * \log_2 q * m$  would become half of it.

## 2.5 Efficient Key exchange mechanism

Most of the Travel document protocols make use of key exchange algorithms and as mentioned earlier, are using Diffie Hellman, currently for establishing a secured symmetric encryption channel by using public key infrastructure.

The first step towards making this the reality, is to have the internet standard to use the protocols. AKE (Authentication key exchange) component can be added to any key exchange mechanism, which is analogous to Internet key exchange v2 (RFC 5996). It will be achieved by replacing a component of the internet key exchange with lattice component. But that would need to have a key exchange mechanism that is analogous to RSA-KEM (RFC 5990). Key exchange mechanism would allow the key to be transported from the receiver to the sender, securely. Peikert, has discovered a reconciliation scheme that would allow just an additive overhead to the cipher text. This would allow a fixed overhead with a bit by bit processing against an overhead added to every bit. Currently, it is under processed to be written (BCNS'14) in C but it has been theoretically established that it would provide 128 bits security.

The authentication key exchange would mean that the two parties namely sender and receiver have mutually authenticated each other and have established a consistent view of having the session completed along with the common key without having the adversary found it.

SK security with post specified peer model (CK' 02a) is designed to ensure that the responder's identity is discovered during the protocol, which is one aspect that can be applied to electronic identities. Authentication key exchange named Sign and Mac (Sigma) protocol (Kra'03) was designed especially to meet these requirements. It uses Diffie Hellman protocol, which can be replaced a lattice based Key exchange mechanism and a passively secured Key exchange mechanism. Sigma has been shown to be highly secured as it is backed by Diffie Helman, which as of today, is computationally secured. The bottom line is that with just replacing the component of DH with lattice based passively secured algorithm, it can be made efficient and be quantum attack resistant. Author of Sigma claims 'The SIGMA protocols provide perfect forward secrecy via a Diffie-Hellman exchange authenticated with digital signatures, and are specifically designed to ensure sound cryptographic key exchange while supporting a variety of features and trade-offs required in practical scenarios (such as optional identity protection and reduced number of protocol rounds). As a consequence, the SIGMA protocols are very well suited for use in actual applications and for standardized key exchange. In particular, SIGMA serves as the cryptographic basis for the signature-based modes of the standardized Internet Key Exchange (IKE) protocol (versions 1 and 2)'

A level down would be the Key Encapsulation mechanism, which works in a unique manner. The public key of receiver is passed to the encapsulation algorithm, which outputs the common key  $k$  to the sender and encapsulated message  $m$ . When this message is passed to the DE-capsulation algorithm with private key of the receiver, it outputs the common key  $k$ . The principle is to offer passive security that when the adversary has access to public key and the encapsulated message; it learns nothing about the common key  $k$ . When there is access to private key, only then, it can obtain the common key  $k$ . For this to be true, two unique and independent keys when encapsulated must remain indistinguishable. i.e. (public key, encapsulated message for key  $k_1$ ) must remain indistinguishable from (public key, encapsulated message for key  $k_2$ ) . Stronger security can be provided with active security, which has adversary access to both encapsulation and DE-capsulation oracle and yet, the common key  $k$  cannot be retrieved by the adversary. There have been claims of many actively secured KEM's. RSA-OAEP (BR'94, Shoup'01) is one of the oldest known practical KEM providing active securities, which is based on RSA. Later, RSA-KEM (RFC 5990) was discovered, which was again based on RSA. Hashed El-Gamal (BR'97, CS'98, ABR'01) based on ElGamal scheme, React transform (OP'01) based on any injective trapdoor function, FO transforms (FO'99a,b) based on any Chosen plaintext attack secured encryption were subsequently implemented in practical scenarios. Unfortunately, as all of them are based on non-quantum attack based algorithms.

Hence, the common idea would be to make use of Ring-LWE problem based on lattices to make the KEM's quantum attack resistant. Error reconciliation mechanism, as explained above can be used to develop the model. The core idea would be to ensure that active security is provided, which is essentially chosen cipher text attack secured. It has been established that

there are many possible transformations to convert a passively secured KEM into active secured KEM. The common idea would be to plug component of Ring LWE KEM into the active secured KEM's mentioned above. The empirical experiments suggest that it is inapplicable, inefficient and makes KEM insecure, if done so. When components of Hashed ElGamal are replaced, then it becomes insecure as it relies on search and decision equivalence of Diffie Hellman and ring LWE search and decision equivalence has different complexities. OAEP and React transform are inapplicable because they make use of injective trapdoor functions. The conclusion is that Fujisaki-Okamoto transformation(FO99b) is a good solution as it maintains the plain text length but Random oracle/coins are needed for the encryption and the Gaussian sampling, which does effect the practical efficiency. But, It also offers active security.

As mentioned above, Learning with error works in a simple manner. Sender encodes a message bit  $u \in (0,1) (\mathbb{Z}_2)$  and  $v = u * \text{floor}(q/2) \in \mathbb{Z}_q$ . The receiver recovers  $w \sim u * \text{floor}(q/2)$  and it is approximate as it comes from learning with errors. Receiver then computes  $u$  by rounding  $u = \text{floor function}(2/q * v) \in (0,1) (\mathbb{Z}_2)$ . The only issue is that  $\log_2 q$  factor is overhead per message bit as  $(\mathbb{Z}_2)$  bit is transformed into  $\mathbb{Z}_q$  bits. To reduce this, a bit for bit error-tolerant reconciliation mechanism can be used. The concept is similar that sender has pseudo random  $v \in \mathbb{Z}_q$ , receiver can recover  $w \sim v$ . A cross rounding function  $\langle v \rangle_2 = \text{floor function}(4/q * v) \bmod 2$ . If  $v$  is uniform, then  $v \bmod 2$  and  $\langle v \rangle_2$  would be uniform as both use similar function structures. Hence, if  $\langle v \rangle_2$  is known and any  $w \sim v$  is available to receiver,  $v \bmod 2$  can be recovered. Using this concept, the Ring-LWE (ex: cyclotomic rings using principles of associativity) KEM can be built.

Let  $R = \mathbb{Z}[X]/(X^{n+1})$  and  $R_q = R/q * R$  such that Errors in  $R$  are polynomials with small Gaussian Integer coefficients. Operations in  $R_q$  can be made efficient by using FFT algorithms

Public key is  $(a, b \sim a * s) \in R_q \times R_q$  for secret error  $s \in R$

Encapsulation process:

Choose error  $r \in R$ , let  $u = r * a, v = r * b \in R$ , output

$C = (u, \langle v \rangle_2) \in R_q \times \mathbb{Z}_2$ , key  $k = v \bmod 2 \in \mathbb{Z}_2$

DE-capsulation process:

As it can be observed that  $u, \langle v \rangle_2$  are known along with  $u, w$ . Hence, obtaining the key is possible and ring-LWE (LPR'10) proves that it is passively secured scheme and when the scheme is passively secured. Fujisaki-Okamoto transformation(FO99b) transforms it into a actively secured scheme. The process of DE-capsulation can simply be calculated as following

$w = u * s \sim r * a * s \sim r * b \sim v$  ( $\sim$  denotes the error)

## 2.6 Lattice cryptographic algorithms

Micciancio and Regev in ‘Lattice-based cryptography’ have mentioned that lattice based cryptosystems can be classified into two categories where one class does not offer the strong provable security guarantees but is too efficient whereas the other class are of theoretical interests as they offer strong security guarantees but are too inefficient. Security guarantees would mean that breaking the encryption scheme can be shown to be at least as hard as solving several lattice problems within polynomial factors in the worst case. The most notable schemes are Goldreich–Goldwasser–Halevi (GGH) cryptosystems and its Hermite Normal form (HNF) variant and NTRU. They are primarily practical proposals aimed at offering a concrete alternative to RSA and other number theoretic cryptosystems. The notable algorithms offering strong security would be Atjari-Dwork Cryptosystem, Learning with Error (LWE), recently developed Ring-LWE. The cryptosystems mentioned here are proved to offer semantic security under chosen plain text attack.

### 2.6.1 GGH/HNF public key cryptosystem

GGH cryptosystem was proposed by Goldreich, Goldwasser and Halevi [AM ‘96]. At a high level, the algorithm works as follows:

The **private key** is a lattice basis  $B$  where the basis consists of short and orthogonal vectors. This allows to efficiently solving certain instances of closest vector problem when the target is very close to the lattice.

Lattices have a property of allowing the same lattice to be developed when different equivalent basis have been selected. GGH cryptosystem makes use of the same concept. The **public key**  $H$  is the Hermite Normal Form of  $B$ . Hermite Normal form can be computed efficiently by Gaussian Elimination. Gaussian elimination allows the matrices to be in row echelon form, where the leading coefficient of the row (the first non-zero value of the row) always appear towards the right than its upper row and the matrices would be an upper or lower triangular matrices.

#### Encryption process

The message is encoded into a short noise vector  $\mathbf{r}$  and is added to a chosen lattice point  $\mathbf{v}$ .  $\mathbf{v}$  should be selected such that all coordinates of  $(\mathbf{r}+\mathbf{v})$  modulo is reduced along the diagonal of the public basis  $H$  i.e.  $\mathbf{r}+\mathbf{v} \bmod H$  would result in a matrix which has been reduced over the diagonals as the HNF diagonals would have non-zero element. This would take the advantage of the subset sum problem. This is the encryption process.

The vector resulting from this process is known as  $\mathbf{r} \bmod H$  and it is proved that it is hard for cryptanalysis as one can chose particular plain text and also choose any  $\mathbf{v}$  that belongs to the lattice and modulo would result in getting  $\mathbf{r} \bmod H$  but the chosen  $\mathbf{r}$  might not be the message as  $\mathbf{r}+\mathbf{v} \bmod H$  can be easily obtained by choosing many vectors  $\mathbf{v}$  and message vector  $\mathbf{r}$ .

#### Decryption process

When the private key mentioned above is known, basis consisting of short and orthogonal basis allow to efficiently solve certain instances of closest vector problem closest to target cipher text  $c$  and the message vector can then be obtained by computing  $c-v \bmod H=r$ , which is the encoded message vector

The cryptanalysis of this algorithm would have to solve closest vector problem closest to the target ciphertext  $C$  with the basis of the HNF form of the lattices, which is computationally hard and the hardness increases with the increasing dimensions of the lattice. One can easily distinguish the large number of cipher texts corresponding to large number of fixed messages and identify  $v$ , which can be solved by randomly pad the message.

### Problems with the cryptosystem

The concern with using the cryptosystem is that it needs large memory space as the general lattice basis need to be stored, which would mean proportionally  $n^2$  factor of memory is needed where  $n$  is the dimensions of the lattice. But no good attack to GGH is known with moderately large values of the security parameter.

However, through the paper ‘lattice based cryptography- a practical implementation’ in the year 2011, Michael Rose has optimized the GGH cryptosystem and has named it ‘An optimized GGH style Cryptosystem’ which optimizes the implementation and reduces the public basis size and improves encryption speed and also optimizes the private key space. It was implemented and tested on a 2.1 Ghz Intel core 2 Duo with 4 Gb Ram and the code is written in C++ and compiled against GNU Multi precision library [SA ‘13] But the computational time required for the original GGH cryptosystem, particularly on the encryption and decryption phase cryptosystem made the implementation impractical but this new construction drastically reduces the public key and private key sizes. The analysis, mentioned by the author is mentioned below where GGH does not make use of the HNF form and Micciancio makes use of GGH along with HNF form and the new stands for optimizing the GGH scheme

| Dimension:            | 500   |        |        | 800   |       |        |
|-----------------------|-------|--------|--------|-------|-------|--------|
|                       | GGH   | Mic    | New    | GGH   | Mic.  | New    |
| Encryption Speed (ms) | 39.60 | 8.977  | 0.1415 | 127.8 | 22.49 | 0.3155 |
| Decryption Speed (ms) | 1608  | 215.3  | 140.2  | 8153  | 851.4 | 630.0  |
| Private Key size (Mb) | 86.36 | 194.0  | 0.6188 | 374.2 | 861.7 | 1.685  |
| Public Key size (Mb)  | 37.80 | 0.3925 | 0.1711 | 149.6 | 1.085 | 0.4654 |

Figure 7: Comparison of GGH algorithms [RM ‘11]

The author also states that the through put of the encryption and decryption is competitive with RSA and decryption error attacks and side-channel Branch Prediction Analysis attacks can also not be performed on this newly designed cryptosystem

### 2.6.2 NTRU Public key cryptosystem

NTRU was discovered in 1998 by Hoffstein, Pipher and Silverman [MR '08] . The lattices used in this cryptosystem make use of lattices in even dimension  $2n$  satisfying two properties. First, they are closed under one to one linear transformations i.e. the vectors of the lattices are rotated cyclically. Secondly, they are  $q$ -ary lattices. A  $q$ -ary lattice  $L$  is defined as that in which any vector which consists of multiples of some scalar  $q$  is in the lattice  $L$ . While any lattice can be represented as a  $q$ -ary lattice (for instance if  $q$  is a multiple of  $\det(L)$  which is the volume of the  $n$ -dimensional lattice), primarily  $q$ -ary lattice cryptography is concerned with the cases where  $q$  is a small, possibly prime value. [MR '08] They have seen to be extensively used in producing provably secure lattice/based cryptosystems [AM '96]

The **private key** in NTRU is a short vector  $(f,g)$  which can be seen as the smallest convolutional modular lattice, which satisfies the partial order and the duality nature of the matrix. Other Specific details, regarding the parameters can be understood from [ 13 ].

**Public key** of NTRU follows the general GGH/HNF framework that the HNF basis of the private key is computed. The HNF public basis will have a form as mentioned below

$$H = \begin{bmatrix} I & O \\ T * H & q * I \end{bmatrix}$$

$H = (T*f)^{-1} g \pmod{q}$ ,  $I$  is the identity element, and  $T$  is the inverse element for the private key yielding identity element over multiplication group

### Encryption process

An input message is encoded as a vector just as in the case of GGH encryption scheme hence,  $m \in \{1,0,-1\}^n$  This message is then concatenated with a randomly chosen vector  $r$  of the same form. Hence, a new short vector is obtained, which is of the form  $(r,m) \in \{1,0,-1\}^{2n}$

When this vector is reduced by modulo public key  $H$ , we complete the encryption process, the matrix for the encryption and decryption process is from [HHPS '07]

$$\begin{bmatrix} r \\ m \end{bmatrix} \pmod{\begin{bmatrix} I & O \\ T * H & q * I \end{bmatrix}} = \begin{bmatrix} 0 \\ (m + [T * h]r) \pmod{q} \end{bmatrix}$$

The non-zero vector would be the cipher text

### Decryption process

The ciphertext  $c$  is decrypted by multiplying the private key modulo  $q$  which would yield the message vector, which is shown below

$$[T*f]c \pmod{q} = [T*f]m + [T*f][T*h]r \pmod{q} = [T*f]m + [T*g]r \pmod{q}$$

Before reduction modulo  $q$ , the decryption process would allow recovering the exact values of  $[T*f]m$  and  $[T*g]r$  and when reduced over  $P$ , the exact message vector would be produced as

$P$  is the encoder for obtaining the private key such that  $f$  and  $g \in \{P, 0, -P\}^n$ . Hence the message vector would be produced as shown below

$$[T*f]m + [T*g]r \bmod P = I*m + 0*r = m \text{ i.e. the message vector}$$

### Problems with the cryptosystem

Based on Howgrave-Graham's hybrid attack [HHPS '07], NTRU had to increase the security bits and as of now for achieving security level for 80 bits, the dimension of the lattice should be 449 with  $q$  above  $2^8$  and the public key size should be 3592 bits and for the estimated security of 256, the dimension of lattices should be 14303 and key size should be 114424 bits [MR '08]

### 2.6.3 Ajtai-Dwork Cryptosystem

Previously mentioned cryptosystems are unfortunately quite inefficient. Ajtai presented an efficient cryptosystem whose public key scales in the proportion of  $n^4$  and the encrypted bits remain proportional to  $n$  bits. If a session key in the scale of  $n^2$  can be agreed upon then the public key remains in the proportion of  $n$  bits. Detailed parameters are also presented by Ajtai. [AM '05]

Ajtai has also constructed a variable on Ajtai-Dwork Cryptosystem, which reduces the public key size proportional to  $n \log n$  by using a compact representation of special instances of SVP, whose security is based on a certain Diophantine approximation problem. [AM'05]

### 2.6.4 Other Lattice based cryptosystems

An efficient public key encryption based on Ideal Lattices was proposed by Stehle, Stienfeld, Tanaka and Xagawa which adapts the trap door one way function of Gentry et al (STOC'08) and is based on Learning with error problems on structured lattices. It is efficient as both keys are of length proportional to  $n$ , the security parameter and costs of encryption and decryption are in proportion to bit costs per message bit. It is the first provably secured public key encryption scheme whose security relies on the hardness of Ideal lattices SVP. Cryptosystems based on general lattices mentioned in the references rely on the average case hardness of Learning with error problem and all of them are inefficient due to overload of encryption and decryption and the efficient cryptosystems make use of breakable problems like Diffie Hellman problem [19, 20, 21, 22]. This efficient public key encryption also uses LWE problem. The scheme is divided into two blocks. The first step is use the LWE mapping as an injective trapdoor one way function with the trapdoor being the full dimensional set of vectors in lattices and the author states that it is as hard as Ideal SIS [SSTX'00] Goldreich-Levin hardcore function based on Toeplitz matrices [GO'01] is applied to the trapdoor function and is XORed with the hardcore bits to obtain a semantically secured encryption as the search version of LWE is at least as hard as quantum approximating GapSVP and SIVP on  $n$ -dimensional lattices in the worst case [Reg05]

| Cryptosystem                         | Public key       | Expansion   | Worst-case problem                                                                   | Approximation        |
|--------------------------------------|------------------|-------------|--------------------------------------------------------------------------------------|----------------------|
| LWE, $q = 2^{O(n)}$                  | $O(n^4)$         | $O(\log n)$ | GapSVP                                                                               | $\tilde{O}(n^2)$     |
| Ajtai-Dwork [AD97, AD07]             | $O(n^4)$         | $O(n)$      | unique-SVP*                                                                          | $\tilde{O}(n^2)$     |
| Regev [Reg04]                        | $O(n^4)$         | $O(n^2)$    | unique-SVP*                                                                          | $\tilde{O}(n^{1.5})$ |
| LWE, $q = \text{poly}(n)$            | $\tilde{O}(n^2)$ | $O(\log n)$ | $\frac{\text{poly}(n)\text{-to-}\gamma\text{-GapSVP}}{\text{GapSVP/SIVP (quantum)}}$ | $\tilde{O}(n^{1.5})$ |
| old CCA: $q = \text{poly}(n)$ [PW08] | $n^{2+\delta}$   | $O(\log n)$ | same two choices $\uparrow$                                                          | $n^{5+\delta}$       |
| new CCA: $q = \text{poly}(n)$        | $n^{2+\delta}$   | $O(\log n)$ | same two choices $\uparrow$                                                          | $\tilde{O}(n^2)$     |
| new CCA: $q = 2^{O(n)}$              | $n^{4+\delta}$   | $O(\log n)$ | GapSVP                                                                               | $\tilde{O}(n^3)$     |

Figure 8: Tabular details of algorithms [PC'09]

The above table shows that all the cryptosystems mentioned have large public key sizes making it inefficient. Expansion is the additional ratio of the cipher text to the plain text.

Regev constructed an efficient lattice-based cryptosystem with shorter keys [RO'00] with a security proof that it is based on the worst case quantum hardness of SVP and SIVP. Learning with errors, an important step in the area as mentioned above in the primitives.

Kawachi, Tanaka and Xagawa has proposed a new Multi bit cryptosystems based on lattice problems [BCDP'96] They developed a universal technique derived from the general structure behind GGH, LWE and Atjai-Dwork cryptosystem. The main idea as per the author is that the mentioned cryptosystems use a general structure. The cipher texts of 0 are basically distributed according to a periodic Gaussian distribution and those of 1 according to another. The multi Bit cryptosystem embeds both the distributions into cipher text space that their peaks appear alternatively and regularly. The public key is the information for generating the periodic Gaussian distribution corresponding to 0 and for shifting the distribution to other distribution corresponding to 1 i.e. we encrypt using distribution corresponding to 0 and if the plain text was 1, we shift it using the public key. It is decrypted using the  $k*(d/p)$  where  $p$  is the plaintext space and  $k$  is the generator of the group of periodic distributions and  $k$  and  $d$  are the private keys.

## Chapter 3

# Foundations of Electronic travel documents

This chapter provides an overview of electronic travel documents as well as foundations of the underlying protocols for this thesis.

After an introduction to travel documents, the next topic covered by this chapter would be public key infrastructure of the travel documents. The underlying security protocols, security

goals would be followed next. The constraints of the electronic passports would be the last topic of this chapter. This should help us in determining the set of algorithms, which we can relook for the rest of the thesis. This is the last chapter in our research to explore various number of papers to understand lattice based cryptography and travel documents. The Chapters, following this chapter would be our own research to contribute to the community. The main contribution as mentioned in the abstract and the preliminary chapters is to research into lattice based cryptography and explore each and every requirement of the travel protocols and evaluate the algorithms to propose the lattice based algorithm that could fit in the travel documents protocols. We do not claim to have discovered any of the areas mentioned in this chapter as our goal is to set the foundation and find the underlying requirements of the travel documents

### 3.1 Travel documents

‘A passport is an internationally recognized travel document for verifying the identity and nationality of the bearer. The primary data as visible from any individual passport and that can be found on all passport are the unique passport number, national identity number, full name of passport holder, country code, color photograph, passport holder’s signature, nationality, date of issue, expiry date of passport, date of birth and place, sex and passport issuing authority and this data is written over the machine readable zone with appropriate check sums. An electronic passport contains a radio frequency ID (RFID) with encoded data. The addition of the electronic chip is intended to provide additional resistance to forgery.’ [NR’09]

Authentication can be achieved by what one has- card, token, key or what one knows- pin, password and what one is-biometric. The epassports contain encoded data of the information printed on the passport, digital picture of the passport holder for biometric facial recognition and digital signature with the chip number. Digital signing and securing of biometric and other sensitive information is achieved by RSA, Elliptical curve and other non-quantum resistant algorithms. The general procedure consists of taking the data to be stored and arranging in the logical data structure. The face is primary biometric information and fingerprint, palm print and Iris is secondary biometric information, which is subject to change. The data is signed using a public key infrastructure to protect the integrity and to prove authenticity. The data is then written to the memory of the chip.

The initial Epassports did not offer even the basic security since the data was not encrypted. It was just digitally signed allowing adversary to clone and gather data illegally through passport skimming but not modify the data. United States for border security took a major step. It mandated the adoption of International Civil Aviation Organization (ICAO) in its Visa waiver program. Inadequate data protection leads to development of new specification by ICAO. Extended Access Control (EAC) was introduced that reduced some of the privacy issues. The second generation of passports enhanced its focus on fingerprints and IRIS by using left and right first finger print in JPEG format. Finger print and Facial recognition covers a larger set of biometric rules allowing better recognition. For example, Facial biometric systems are

sensitive to changes in light, pose of the head and the head scarf, which is enough to make the facial biometric change completely. Hence, combining with fingerprint and IRIS would ensure that there are minimum effects. It led to the development of the concept of mutual authentication between terminal and the passport. Third generation passports has Password-Authenticated Connection Establishment (PACE) mandatory making use of non-quantum resistant elliptic curve Diffie-Hellman key agreement. This was added due to Bundesamt für Sicherheit in der Informationstechnik (BSI) as it released a new stack of protocols called EAC v2.1, which fixed some issues present in the original EAC proposal and asked for protocol PACE in EAC.

Juels et al (2005) discussed security and privacy issues that apply to e-passports. The main issue was that the chip's contents could be read with in a close range, even if the passport booklet was closed. The data could be obtained by adversary by simple skimming of passport or eavesdropping. We must also consider that the data stored in the passport is not very noteworthy but there was still a noteworthy privacy issue. Due to the low entropy of the secret keys (the possibilities of secret keys were limited, brute force attack was demonstrated by Laurie (2007), Karger and Kc(2005). The original biometric information could be altered (known as splicing attack) and a finger print could be stolen (fake finger print attack). This allowed the adversary to alter the data or just take the picture of finger print and clone it. The passport was used for presenting as an identity card at many places like restaurants, hotels, police verification etc. Considerable papers (Johnson, 2006; Knight, 2006; Reid, 2006) could be checked to explore on the weaknesses in epassport, which also talked about the cloning of epassport.

For the cloning attack to work, the adversary will have to steal the legitimate physical passport, undergo a plastic surgery to match the picture. The attacker will also have to modify the fingerprint details, compute new digital signature and hashes to maintain the integrity and replace the public key for the digital signature. The simpler option would be steal identity and issue a legitimate passport. However, cloning breaches the privacy issue that the adversary obtains not just the details stored on the passport but also the biometric data.

## **3.2 Travel documents**

The concept of epassport was introduced by Davida and Desmedt in 1988 and the first study on privacy issues from RFID protocols is due to Avoine and Oechslin [NBS'12] In 2006, Hoepman et al discussed more about unauthorized access and skimming over BAC protocol along with entropy and EAC protocols. Hancke and Carluccio et al. [NBS'12] performed the first experiment on the BAC key weaknesses. Later, man in the middle attack was demonstrated on active authentication. The core distinguishing factors are the biometrics in the passport. Biometrics in epassports complying with the ICAO consists of mandatory facial and fingerprints. Hence, facial prints are used by a number of countries and finger prints are used by relatively few countries. The different biometric possibilities are as follows:

### **3.2.1 Face Image**

The most common biometric characteristics used by humans to make personal recognition. It is not a very great idea as the facial images can be changed by a surgery. It is also not a very suitable means as the face images could range from static proper images to images taken while moving in a much cluttered background. Though, there are regulations that only proper images should be used. It includes a simple technology a feature set from the two dimensional image of the users image and it matches with the template in the data base. The most popular approaches to face recognition are based on either 1) location and shape of facial attributes (eyes, nose, lips etc.) and their spatial relationships or 2) overall analysis of face image that represents a face as a weighted combination of a number of canonical faces (possibility of the face). Identifying the face, extracting the features and recognizing it from general view point and extracting the feature set and matching it with the dataset is rather a difficult task. This is extracted from the paper [BTA'09]

### **3.2.2 Fingerprint**

Initially, fingerprints (ridges and furrows located on finger tips) were extracted by creating an inked impression of fingertip on paper. Today, compact sensors are replaced by the digital images. Fingerprint recognition acquires the first print by taking a live scan of the finger by direct contact with the reader and these images are passed to feature extraction module to store the feature values. The minutiae points (position and orientation of certain critical points) are used to compute the feature values. The matching process involves comparing these points later on for the verification. The main issue is that they require a large amount of computational resources. This is extracted from the paper [MJW'05]

### **3.2.3 Palmprint**

Usually, it is used only to compute the identifications of the person. The computational resources required to make a comparison with the stored palm print is large and inefficient as the minutiae points throughout the whole palm needs to be compared in order to get a good result. Hence, the palm print is passed to the system as an input with personal identification details as the expected output. The input image is then the input which is compared with the image stored in the database. Since, palm prints contain huge number of minutiae points; the relevancy is estimated with reference to the threshold value. The most relevant image is selected for identity and if it does not match then it is accepted as an unknown person. This is extracted from the paper [DJ'04]

### **3.2.4 IRIS recognition**

It is based on the distinctly colored ring surrounding the pupil of the eye. IRIS is made out of elastic connective tissue; it delivers a rich set of biometric data with 266 distinctive characteristics. These include a meshwork (the appearance of dividing the IRIS) around 173 characteristics are generally used in distinctive characteristics by the system. The system

involves a black and white high resolution image of the iris and it defines the boundaries of the iris (the area of importance) and then it establishes the area which allows verification. This paragraphs complete information is extracted from the paper [FCR'06]

### 3.2.5 Operations

An e-passport bearer presents the documents whose machine readable zone information in the passport needs to be checked through the reader and then it should be placed near the e-passport reader to fetch data from the chip. ICAO's e-passport guideline provides details on how data should be stored in the chip. The data elements are grouped as a datagroup(DG) and collectively known as the logical data structure (LDS). The guidelines as in the paper [ICAO6] segregate into 19 data groups and LDS has three sections with in. There is the mandatory data; details recorded in the MRZ (passport number, passport bearer's name, nationality, date of birth, expiry, facial biometric image, finger prints (mandatory in EU EAC) and checksum which is necessary for calculation of BAC session keys). The optional data is iris scan, contact details, proof citizenship and endorsements. There is data defined by the receiving state that includes automated border clearance, e-visas and other travel records. The LDS stored in the microchip is stored using the file system as defined in ISO/IEC 7816-4 (ICAO, 2004). DG from 1 to 16 are read only, 17 to 19 can be modified by authorities and write access is currently not implemented but ICAO has plans for it. The dedicated file (DF) in chip file system stores the encryption, MAC and private key of the e-passport bearer used in active authentication. The elementary file (EF) will store the security object descriptors (SOD) and DG. SOD is the hashing of DG individually but collectively signed by the issuing state. The above mentioned information would be used in proposing our protocol as we need to ensure that our protocol also makes use of the same infrastructure to make it feasible to implement it in real time.

The summary of the protocols are as follows; It is well known knowledge that Basic Access Control provides encrypted communication between the chip and the inspection system. Passive authentication allows border security officer reading and verifying the authenticity of e-passport content stored in the chip. Active authentication provides integrity verification of e-passport's data.. Basic Access Control does not detect chip substitution, chip copy and needs extra resources as memory and CPU. There are certain security goals to be met, which is the focus of the e-passport, which are the following

### 3.2.6 Security Goals

These security goals are well known and established in the scientific community. The common ones with respect to the e-passports are mentioned in the following paragraphs.

1. Data Confidentiality: Data confidentiality ensures the privacy of e-passport. An encryption is the common technique that provides confidentiality. The encryption could be using a public key scheme, common key scheme or public key scheme and deriving common session key scheme. In the case of e-passport, encryption is used to create a secure channel between the e-passport reader and the chip using Diffie- Hellman scheme, which is a scheme that will be obsolete, if the quantum computers would come into existence

2. **Data Integrity:** Data integrity disallows modifications of information by third party exchanged between the e-passport reader and the microchip. Also the DF, SOD and LDS should be secured against any unauthorized modifications, even when data is not being exchanged. That is third party or terminal should not be able to read them and change them

3. **Mutual Authentication:** Mutual authentication is the process where both participants prove their identities to each other. It is important for epassport to authenticate the epassport reader before exchanging any personal information as that would leak the information. This prevents an unauthorized epassport reader from obtaining biometric and personal details from an e-passport. Similarly, terminal can also read the epassport and confirm that the passport is authenticate

4. **Key Freshness and Key Integrity:** Key freshness and key integrity protects against replay attacks. The e-passport reader and e-passport must have proof that, a nonce generated in protocols is completely random and the other party has correctly got session key. The common example is of basic access control protocol. Both parties should also have proof that the other party is in possession of a valid session key, which is established by a multiple step encryption process of parts of keys in BAC protocol, explained further.

### **3.3 Public Key Infrastructure**

Public Key Infrastructure (PKI) helps to bind public keys to entities and enables other entities to verify and authenticate themselves. The infrastructure and explanation obtained from the standards for globally interoperable organization consists of the following components:

#### **3.3.1 Certification Authority (CA)**

It is the central component in a PKI, and performs the following functions - issuing certificates, maintaining certificate revocation lists and publishing certificates and revocation lists. The CA issues certificates to PKI users by digitally signing a certificate with its private key and, during verification; a user confirms the authenticity of the certificate by verifying the CA's signature using the CA's public key.

#### **3.3.2 Registration Authority (RA)**

It is an entity trusted by the CA, to register or validate the identity of users to a CA, that is, its primary responsibility is to verify whether the certificate contents reflect the information presented by the entity requesting the certificate.

#### **3.3.3 Repository**

A repository is a database of active (valid) digital certificates. The repository provides information, to allow users who receive digitally signed messages to confirm the status of the digital certificates.

### 3.3.4 Public Key Certificates

The CA issues a public key certificate for each identity. A digital certificate typically includes the public key, information about the identity of the entity holding the corresponding private key, the validity of the certificate, and the CA's own digital signature.

### 3.3.5 Certificate Revocation List (CRL)

CAs also issues and processes certificate revocation lists (CRLs), which list revoked certificates. Every PKI user validating a certificate is also required to process the CRL to check if the certificate has been revoked.

### 3.3.6 PKI Users

PKI users are those who use and rely on PKI components to obtain and verify certificates of other entities with whom they transact. Using PKI is the dominant method for verifying an entity's public key, and thus plays an important role in both semi-passive devices and active hardware-based security devices, for example, e-Passports rely extensively on PKI for the validation of certificates.

The certificate Authorities can be divided into Country Signing Certificate Authorities (CSCA) and Country Verifying Certificate Authority (CVCA). The CSCA generates and stores a key pair ( $K_{Pu_{CSCA}}$ ,  $K_{Pr_{CSCA}}$ ). The private key is then used to sign each Document Signer (DS) in the country. Each of these DS generates and stores the key pair ( $K_{Pu_{DS}}$ ,  $K_{Pr_{DS}}$ ). The private key of DS, then, signs the security data element (SOD) of every passport, it issues. All the DS certificates can be found by ICAO's public key directory (PKD) The PKD will store only the certificates of all registered DS's. CRL are also stored in the same PKD. Every nation is responsible for updating its own repository of public certificates and CRL's by downloading them from the PKD, after which, each nation distributes this information to every DS in its jurisdiction. Inspection device can check signatures using DS certificates.

Similarly, CVCA generates and stores a key pair ( $K_{Pu_{CVCA}}$ ,  $K_{Pr_{CVCA}}$ ). The private key is then used to sign each Document Verifier (DV) in the country. Each of these DV generates and stores the key pair ( $K_{Pu_{DV}}$ ,  $K_{Pr_{DV}}$ ). The private key, in turn is used to sign each Inspection System (IS) certificate in its domain. E-passport can check authenticity of the inspection device using the DV certificate

When a DV private key is compromised, the country cannot automatically invalidate all the passports issued with this key. The passport signed by any private key is expected to last more than the duration of the passport. It is not possible to revoke thousands of passports when the key is compromised. The whole architecture would ensure that when DS private key is compromised, custom officials can be notified of passports signed by the particular DS and the passports can be used as normal. The importance is of the CSCA private key, it must be replaced every 3-5 years to ensure security though the key must be lifetime of the passport. If the DS key needs to be revoked then, CSCA must communicate within 48 hours to the PKD

and to all other countries and a full revocation list must be exchanged within 90 days [Kc and Karger 2005] ICAO, 2006 says that each participant country is responsible for downloading the latest version of the keys and ensuring that passports are indeed signed by the document signer.

### **3.4 Travel documents protocols**

ICAO published a set of guidelines (in Doc 9303) that were meant to be followed as ePassport standard. As mentioned above, the default mandatory biometric to be used is the digital image of the individual, other allowable biometrics are fingerprints and iris images. There are three Cryptographic protocols described in the first generation ICAO specifications to ensure data correctness and privacy, which are Passive Authentication, Basic Access Control, and Active Authentication. These are well known and we do not claim that we have discovered them but we researched to find out each and every protocol used in the travel documents and we took excerpts from number of papers as mentioned in the references. There is also a possibility that some words in the steps of the passport protocols might match the exact words used in this paper and it is necessary as we can not use alternative words and most steps are defined by authorities in their paper so we do not feel it the right step to re-word these steps. However it is important to have these protocols, a place in the thesis document as it is important for us to evaluate the requirements in order to set the benchmark to evaluate lattice based algorithms

#### **3.4.1 Passive Authentication**

Passive Authentication is the only mandatory cryptographic protocol in the ICAO first generation epassport specification. The primary goal of the protocol is to ensure that the data in the epassport matches the authenticity requirement. The tag, itself does not process but is just reading and verifying. It only lets the terminal ensure that the tag holds the right data but does not verify if the RFID was cloned. The process evolves as following

- 1) The terminal gets the certificate of the issuing document singer (as mentioned in the previous section), using the public key from the certificate
- 2) It verifies the digital signature used to sign the data in the LDS.
- 3) Once the validity of the signature is ensured, the reader computes the hash of each of the data elements and compares them with the hashed values stored in the SOD (elementary file). If both the hashes match, it can be established that the data on the RFID was not altered but if it was cloned cannot be confirmed by this protocol.

#### **3.4.2 Active Authentication**

Active Authentication is an optional protocol in the ICAO first generation specifications as cloning of chip is very difficult for epassports as there are also physical issues that needs to be addressed by the adversary in order to perform cloning. Using a simple challenge-response mechanism, it aims to detect if RFID has been cloned. If Active Authentication is supported,

the RFID on the ePassport stores a public key ( $K_{P_{uAA}}$ ) in Data Group 15 of the LDS (elementary file) and its hash representation in the SOD as mentioned in the earlier sections. The corresponding private key ( $K_{P_{rAA}}$ ) is stored in the secure section of memory (dedicated file). In order for the radio frequency chip to establish its authenticity, it must prove to the terminal reader that it possesses this private key.

- 1) The terminal sends a random 64 bit string to the chip
- 2) The Tag signs this string with the key  $K_{P_{rAA}}$  and sends it back to the terminal
- 3) The terminal reader gets the public key  $K_{P_{uAA}}$  stored in Data Group 15
- 4) The terminal reader uses the public key of the tag to open the signature and checks it with the string

### 3.4.3 Basic Access Control

Basic Access Control (BAC) is an optional protocol that tries to check that only terminal readers with physical access to the passport can read chip. When a terminal reader attempts to read the BAC enabled ePassport, it ensures to have terminal reader to prove knowledge of a pair of secret keys that are derived from data on the Machine Readable Zone (MRZ) of the passport i.e. only if terminal reader can read the MRZ of the passport, it can read the passport and to read the MRZ, it needs the physical access to the epassport. From these keys, a session key is then obtained using Diffie Hellman, which is used for secure messaging.

The Access Keys ( $K_{ENC}$ ,  $K_{MAC}$ ) are derived from the following data available on the MRZ: The Passport Number (Doc No), Date of Birth of the Passport Holder (DOB), Valid Until Date of the Passport (DOE), 3 Check Digits (C).

$$K_{seed} = 128\text{msb}(\text{SHA} - 1(\text{DocNo}||\text{DOB}||\text{DOE}||\text{C}))$$

$$K_{ENC} = 128\text{msb}(\text{SHA} - 1(K_{seed}||1))$$

$$K_{MAC} = 128\text{msb}(\text{SHA} - 1(K_{seed}||2))$$

The terminal reader will now enter a Challenge-Response mechanism (described below) to prove possession of the access keys and to derive a session key.

- 1) The chip computes and sends the terminal, a 64 bit string ( $R_T$ ).
- 2) The terminal reader gets  $R_T$  and generates two random 64 bit strings ( $R_R$ ,  $K_R$ ).
- 3) The terminal reader encrypts  $R_R||R_T||K_R$  using 3-DES algorithm (which is a symmetric algorithm based on the DES algorithm) to which the key  $K_{ENC}$  is used.
- 4) The terminal generates the MAC of the cipher to which the key  $K_{MAC}$  is used.
- 5) The terminal reader sends the cipher obtained in the step 3 and the MAC from the previous step to the chip.
- 6) The chip checks the MAC, decrypts the cipher (which proves that terminal has access to epassport else  $K_{ENC}$  and  $K_{MAC}$  wouldn't have been calculated). It verifies the correctness of  $R_T$  and then extracts  $K_R$ .

- 7) The chip generates another 64 bit random string  $K_T$  . Now, the chip knows  $(R_R, K_R)$  and terminal knows  $(R_T)$
- 8) The chip encrypts  $R_T || R_R || K_T$  using the 3-DES algorithm using  $K_{ENC}$  , which is obviously available to the chip as it can compute these values where  $R_R, K_R$  has remained secret and is randomly send by the terminal
- 9) The chip computes the MAC just like the terminal does in the step 4, with the key  $K_{MAC}$
- 10) The chip sends the cipher and the MAC to the terminal.
- 11) If the terminal is legitimate then the chip will know  $(R_R, K_R)$  and terminal will know  $(R_T, K_T)$  The terminal reader checks the MAC, decrypts the cipher. It verifies  $R_R$  and then extracts  $K_T$  , where by checking  $R_R$ , chip is authenticated
- 12) Both the terminal reader and the chip compute the session key seed ( $K_{seed}$ ) as  $K_R \oplus K_T$  (which as can be seen, it is a combination of two random string, where one was selected by the chip and the other one by the terminal). Now both parties generate a new session encryption key  $K_E$  and a session MAC key  $K_M$  as shown below, and since, it is based on  $K_R$  and  $K_T$ , the key will always be session keys as different strings will be selected every time

$$K_E = 128\text{msb}(\text{SHA} - 1(K_{seed}||1))$$

$$K_M = 128\text{msb}(\text{SHA} - 1(K_{seed}||2))$$

Communication is secured using the above encryption and MAC key where the strength lies on the Passport Number (Doc No), Date of Birth of the Passport Holder (DOB), Valid Until Date of the Passport (DOE), 3 Check Digits (C).

The entropy of BAC is just 56 bits, where most of the bits can be guessed. Here is the explanation of how it is just 56 bits. The 56 bits is obtained by considering the nine digit passport number providing  $10^9$ , date of birth with  $(365*100)$  (the assumption is that the person won't be older than 100 years and 365 days for selecting the day in the year in the last 100 years)) and  $365*10$  as the passport expiry date would be in the next 10 years (as passports are valid for 10 years to come) providing a total of  $365^2*10^{12}$  approximating to  $2^{56}$ . Out of which, date of birth for well known people is publicly available and attacks on german-dutch passports reduced the strength to 20-35 bits (Kalach, Quisqater, 2007)

#### 3.4.4 Second Generation ePassports

The primary goal of EAC was to provide mutual authentication to chip and terminal reader authentication as there are no other protocols as mentioned in the previous sections, which effectively meets this goal. It also aimed to call for implementation of secondary biometrics for additional security. To achieve mutual authentication, the EAC proposal introduced two new protocols called Chip Authentication and Terminal Authentication. These were used to supplement the Passive Authentication protocol, Basic Access Control protocol [NR'09] .

It was important as it was shown that active authentication is based on the fact that the original chip has knowledge of private key. However side channel attacks (power and timing attacks)

(Bogari, Zavarsky, Lindskog, Ruhl, 2005) showed that it was easy to obtain the active authentication private key from the original chip. But unfortunately, later, it was shown that it completely mitigate the side channel attacks (Blundo et al, 2009) the faraday cages could be used to avoid preventing and skimming but eavesdropping could not be checked as adversary could still eavesdrop on legitimate conversation and some information could be obtained. A more detailed explanation is in the following paper in the reference number [NR'09]

### **Chip Authentication**

The Chip Authentication protocol is a mandatory protocol in the EAC specifications. It aims to replace Active Authentication as a mechanism to detect cloned ePassports. If Chip Authentication establishes a new pair of encryption and MAC keys to replace BAC derived session keys and enable secure messaging, which has the entropy based on just on simple data published on the machine readable zone of the passport. It does this using the static Diffie-Hellman key agreement protocol, which can be broken by the quantum attacks. ePassport chip has a Chip Authentication public key (in Data Group 14) and private key (in secure memory) (Pu, Pr). The process of Chip Authentication is described below.

- 1) The chip sends the public key Pu to the terminal reader along with the Diffie-Hellman key agreement parameters (D) (prime number p and the group base g).
- 2) The terminal reader verifies the correctness of the received key using Passive Authentication. After the terminal ensures that the data in the LDS is validated, it is easy to confirm that the received key is the genuine public key
- 3) The terminal uses the D to generate its own public and private key pair (K<sub>Pu</sub>, K<sub>Pr</sub>), where  $K_{Pu} = g^{K_{Pr}} \text{ mod } p$ . In classical attacks, it is not feasible to compute  $K_{Pr} = \log_g(K_{Pu}) \text{ mod } p$  as it is equivalent of solving either the discrete logarithm problem of the diffie hellman problem
- 4) The terminal sends the generated public key K<sub>Pu</sub> to the chip
- 5) Chip computes (R<sub>Pu</sub>, R<sub>Pr</sub>) where  $R_{Pu} = g^{R_{Pr}} \text{ mod } p$  and sends it to the terminal
- 6) The Reader and Tag can now generate a new seed key (K<sub>seed</sub>) using this shared information by  $K_{seed} = (K_{Pu}^{R_{Pr}} \text{ mod } p)$ . The last few steps are the core steps of the Diffie hellman algorithm, whose underlying can be broken with the advent of quantum computers as shor's algorithm could render it obsolete
- 7) The new encryption and MAC keys are generated as described in BAC.

### **Terminal Authentication**

The Terminal Authentication protocol is a protocol necessary, if access to more sensitive data (secondary biometrics) is required. It is a challenge-response mechanism that allows the chip to check the terminal used in Chip Authentication. It ensures that terminal is authentic as the chip authentication ensures that the chip is authentic. The terminal proves to the chip using digital certificates that it has been authorized by concerned nations to read ePassport chip. The process of Terminal Authentication is described below;

- 1) The terminal reader sends an Inspection System certificate to the chip (which was received from the local DV) and the DV's certificate (which was received from the CVCA, which will validate that the inspection system certificate is valid). Both the certificates are mentioned in detail in the PKI section, above.
- 2) The chip inspects the certificates and obtains the public key ( $TKP_{uc}$ ) of the terminal from the Inspection System certificate. This will ensure that the terminal public key is not faked as DV has signed it and CVCA certificate will ensure that DV was not compromised.
- 3) The chip computes a random string (R) and sends it to the terminal.
- 4) The terminal generates the hash of  $TKP_{uc}$ , which is the public key of the chip. Chip Authentication public key is stored in the LDS (in Data Group 14)
- 5) The terminal signs the message ( $R||SHA-1(TKP_{uc})$ ) with its private key ( $TKP_{rt}$ ).
- 6) The chip verifies the correctness of R and  $TKP_{uc}$  using the key  $TKP_{ut}$  (obtained from the certificate) and grants access to sensitive data. It is quite similar to the active authentication protocol with the difference that active authentication authenticates the chip whereas terminal authentication authenticates the terminal

A major issue still pending was that the epassport chips made estimates of time from the terminal (i.e. they take time stamps from the terminal) and when the chip's time has not been updated since a long time for infrequent travelers then and terminal reader with expired certificate could have access to the chip (It can be done in the following manner. 1) The terminal is compromised 2) The terminal sets the time to an older one for which it holds the DV and CVCA certificate and then it lets the chip have the perception that the terminal is authenticate by the terminal authentication protocol and shares the important biometric information), which is also an issue in the third generation passports. The malicious terminal reader could also flood the chip with invalid certificates during the terminal authentication, due to chip's limited memory, the chip would stop functioning. By altering the terminal and chip authentication, denial of service attack was mitigated. Further details are mentioned in the paper in the reference of [NR'09]

### 3.4.5 Third Generation ePassports

In 2008, Bundesamt Für Sicherheit (BSI - Germany) called for describing new security mechanisms for electronic passports [NR'09] . It was then implemented for electronic identity cards, electronic signatures in Germany; it made its way to the electronic passports. The third generation specification introduces a new protocol called Password Authenticated Connection Establishment (PACE). In addition to PACE, the Terminal Authentication and Chip Authentication protocols were also updated, which still used Diffie Hellman algorithm. The PACE (Password Authenticated Connection Establishment) protocol is introduced as a replacement to the Basic Access Control mechanism.

#### Password Authenticated Connection Establishment (PACE)

PACE replaces the Basic Access Control protocol as a mechanism allowing chip to verify that the terminal reader has authorized access to the electronic passport, which is also the goal behind Basic Access Control. The chip and the terminal reader share a common password ( $\pi$ ) which is obtained by the Diffie-Hellman key agreement protocol to provide a stronger session key. The entire process is as follows;

- 1) The chip encrypts random nonce using the key  $K\pi$ . Here,  $K\pi$  is  $\text{SHA-1}(\pi||3)$ . It is mentioned below as to how the password is computed
- 2) The Tchip sends the encrypted nonce and the Diffie Hellman key agreement static domain parameters (D- the group  $g$  and the large prime number) to the Terminal.
- 3) The chip reader uses the shared password ( $\pi$ ) to recover the encrypted nonce.
- 4) The chip and the terminal reader compute the Diffie-Hellman ephemeral key domain parameters using D and the nonce.
- 5) The chip generates the public private key pair given  $(P_{rT}, P_{uT})$  and sends  $P_{uT}$ .
- 6) The terminal reader generates the key pair  $(P_{rR}, P_{uR})$  and sends  $P_{uR}$ . Step 2-6 are obtained using the Diffie Hellman algorithm. The terminal and chip now communicate via a secret channel using the public key encryption using the derived public and private key to generate a seed key ( $K_{\text{seed}}$ ).
- 7) The terminal reader and chip now derive session Keys  $K_{\text{ENC}}$  and  $K_{\text{MAC}}$  in the same manner as in BAC.
- 8) The terminal reader computes an authentication token:

Token-TerminalReader=  $\text{MAC}(K_{\text{MAC}}, (P_{uT}, D'))$  where  $D'$  are the ephemeral parameters and sends it to the chip.

- 9) The chip computes an authentication token:

Token-chip=  $\text{MAC}(K_{\text{MAC}}, (P_{uR}, D'))$

When both authenticate using the token-chip, a secret communication channel has been established and PACE has been successful

The specification allows for two types of passwords to be used with electronic passports. These are Card Access Number (CAN) and Machine readable zone (MRZ) passwords. The CAN may be a short static (can be printed on the passport) or dynamic. When dynamic, chip randomly computes it and displays it on the passport using low power display technologies. The MRZ password is a static type symmetric key obtained from the MRZ, which is the similar case of the BAC though the entropy is much larger and skimming is much more difficult.

### 3.5 Travel documents memory constraints and efficiency

This section is divided into memory constraints and operations-efficiency exploring into the memory type and storage capacity and operations in real time with time efficiency for the protocols.

### 3.5.1 Memory constraints

ICAO maintains that the RFID system implemented in epassports follow the ISO 14443 standard, which specifies the use of 13.56 Mhz radio frequency for communication. The physical features are defined by ISO 7810 ID-3 standard, which mentions the use of 125mm\*88mm along with having antenna built around the chip. The chips can store 32-144 KB of Electrically Erasable Programmable Read-Only Memory (EEPROM) memory. Specifications for Electronically Enabled Passports with Biometric Identification Capability (ICAO, 2006 under the Secretary General) mentions that the total memory size for the elementary file, dedicated files and other files is around 32 KB. EEPROM ensures that when power is removed, the memory is still saved in the chip. When the terminal reader comes in contact with the chip, there is an inductive coupling to transfer power from the reader to the chip. The antennas of both reader and the chip couple a circuit and both the capacitors and inductance complete the circuit. The power needed for the chip is 55 mW and is supplied by Infineon technologies. By using the physics behind 55 mW and load resistance of 550 ohms, the reader and chip need a minimum distance of 9.8 cm to complete the circuit. (Sarma, Weis, Engels, 2002)

### 3.5.2 Operations and Efficiency

Regulation (EC) No 562/2006 of the European Parliament and of the Council of 15 States March, 2006 establishing a Community Code on the rules governing the movement of persons across borders (Schengen Borders Code) states the following;

‘All persons shall undergo a minimum check in order to establish their identities on the basis of the production or presentation of their travel documents. Such a minimum check shall consist of a rapid and straightforward verification, where appropriate by using technical devices and by consulting, in the relevant databases, information exclusively on stolen, misappropriated, lost and invalidated documents, of the validity of the document authorizing the legitimate holder to cross the border and of the presence of signs of falsification or counterfeiting’. [FA’11] also mentions that ‘Though it does not mention regarding the second line checks, whenever there is a doubt of counterfeiting of documents, there is a check for additional biometrics after terminal authentication’. [FA’11] has conducted interviews with European authorities to draw a conclusion on that inspection of the passport is taken by either of these methods;

- 1) First line visual inspection (booklet) only by border guard. It specially applies to passports without the radio frequency chip. ICAO has also described specific physical properties of a travel document to check forgery of travel documents
- 2) First line inspection of booklet and chip by border guard. Biometric comparison of facial image from chip by border guard/ terminal reader

In case of suspicion, either of the steps is generally followed;

- 1) Second line visual inspection (passport booklet) by border guard to ensure that passport is completely genuine along with the person
- 2) Second line inspection of booklet and chip by border guard in combination with an inspection system

Current European biometric-enabled ePassports allow access to basic information to terminals to successfully complete the PACE protocol [BSI10] with the chip. International Civil Aviation Organization (ICAO) regulations ensure that BAC must also be supported; there is also support for the weaker BAC protocol [ICA06] that also allows access to basic data. However, BAC is expected to be replaced by the PACE-based supplemental access control (SAC) protocol by 2018 [ICA13]. For access to the sensitive data, such as biometrics of IRIS and fingerprint, European ePassports require third generation ePassports. The process works as follows;

- 1) PACE allows a secure communication channel
- 2) Passive authentication is then performed along with chip authentication to ensure that chip is not cloned
- 3) If the second line check is needed, terminal authentication is performed, and the sensitive biometric data is read

Different countries have different regulations for the efficiency along with different regulation within (for diplomats and non-diplomats) but a general scenario is that PACE performed takes in the range of 4-15 seconds with passive authentication and chip authentication taking around 10-30 seconds to reach a sufficient throughput of people at border crossing without compromising on security. It also includes the capturing and comparing the facial patterns for around 10 seconds which has 3.3%-70% false non match rate and 0.3-5% false match rate. The second level check needs about 45 seconds with 9 to 19 seconds for fingerprints and 12 seconds for comparing iris patterns. These three biometric checks would ensure authentication of the passport holder (Kumar, Shrinivasan, 2013)

However, there have been many proposals to speed up the processes. Bender et al. [BFK13] proposed to combine PACE with active authentication as a combined process to reduce the computations needed for each of them. Buchmann et al. [BPBP13] propose BioPACE protocol where the ePassport holder's biometrics are used in combination with a biometric template protection scheme as input for PACE (which is the first step in the PACE) in place of the ePassport's Machine Readable Zone (MRZ) or CAN. This removes the need for terminal authentication as the access to sensitive data is not needed then. There is also a proposal to completely eliminate the PACE (Peeters, Hermans, Mennink; 2013) and replace it with a terminal authentication and elliptic mutual authentication scheme [PHF13] based on forward secrecy encryption to establish a secure channel and then all sensitive information can be read along with establishing passive authentication. Hence, the passport protocols might change in the future but as of now, PACE is an important protocol.

Germany and Russia uses elliptic curve digital signature algorithms with sha1 for passive authentication signature. Great Britain, Finland, Poland and Sweden uses Sha256 with RSA encryption and Spain used Sha1 with RSA encryption [FA'11]

This chapter was an account to show that each of the protocols used in the epassports are vulnerable to quantum attacks as the signature algorithm, key exchange algorithm and cryptosystems are all based on RSA-ECDSA-SHA signatures, Diffie Hellman and RSA respectively. It was also needed to mention the memory and efficiency constraints that the new algorithm needs to offer in order to replace the current algorithm as designing epassport authentication protocols needs one to take into account, efficiency and cost constraints on the chip side, which is mentioned in the last section of this chapter. Even after replacing the algorithms with lattice based algorithms, the overall epassport authentication procedure, should maintain the standards of ICAO and other necessary standards. Hence, it is important, that newly proposed algorithms are at least as efficient as the current solution when providing improved security features. Additionally, to keep the cost low, the newly proposed algorithms should also be able to run on the currently available hardware.

## Chapter 4

# Efficient Lattice based algorithms

---

This chapter is to differentiate between different lattice based algorithms mentioned in the second chapter.

The idea of this chapter is to argue about the efficiencies and memory constraints of the lattice based algorithms. It would also focus on expanding on the key exchange mechanism discussed in the earlier chapter and also expand on some of the practically implementable lattice based algorithms.

### 4.1 Efficiencies of the cryptosystems

#### 4.1.1 Small Integer problem One Time Digital signatures

**Inputs:**  $X \in \{0,1\}^{mxm}$  and  $x \in \{0,1\}^{1xm}$ .

$$f_A(x) = Ax \bmod q. \quad A \in \mathbb{Z}_q^{n \times m}$$

**Public key:**  $f_A(X) = |f_A(X_1), \dots, f_A(X_m)| \in \mathbb{Z}_q^{1 \times m}$  (each of the  $X_i$  is a matrix of  $1 \times m$  elements) and  $f_A(x) \in \mathbb{Z}_q^{1 \times m}$ .

**Message**  $M \in \{0,1\}^{1 \times m}$

**Signature algorithm:**  $f_A(y) = X^*M + x^* X^*M \bmod q$  (output of  $m$  columns)  $\in \{0,1\}^{1 \times m}$ .

One time digital Signature for message  $m$ :  $f_A(y)$

**Receiver:** Computes  $f_A(X)^*m + f_A(x)$  and verifies with  $f_A(y)$

**Space complexity:** The two Public key size for the digital signature would be needing  $n \cdot \log_2 q \cdot m$  (for  $f_A(X)$ ) +  $n \cdot \log_2 q$  (for  $f_A(x)$ )

**Runtime:** The matrix multiplication of  $1 \times m$  matrix with  $n \times m$  matrix would result in  $n \times m$

If the data groups of the travel documents were to be hashed and then digitally signed for the passive authentication to take place. The data group hashes would be digitally signed so hashed data can be matched with these hashes after checking the Digital signature.

Considering data of 256 bits (using Sha-256 for each data group) for 19 DG, it would give out message  $m$  of  $256 \times 19 = 4864$  bits. Considering  $n$  as 128 for the security parameter and  $q$  as  $2^8$ . The public key size would work out to be  $4864 \times 128 \times 8$  comes to around 608 KB for first public key and 1KB for the second. Since, this key would not be stored in the chip but be used only in the certificate; it is not a major concern. The time complexity would be 622592, which should take just few seconds for computing the digital signature.

Our suggestion is to use a structured matrix i.e. take the first row and the second row would be one bit left/ right circular of the first row and so on allowing just the first row of the matrix to be stored as the public key i.e. the space complexity from  $n \cdot \log_2 q \cdot m$  can be dropped down to  $m \cdot \log_2 q$ .

i.e.  $4864 \times 8 = 4.5$  KB for the first public key and 1 KB for the second public key. In this way, the public key size is quite reasonable though it increases the computation by an additional factor of 128 as each of these rows needs to be left/right circular of the first row. The computation would still take just a few seconds.

Hermite Normal form could also be used for the public key to reduce half of the size of the matrix i.e.  $n \cdot \log_2 q \cdot m$  would become half of it, which will maintain hardness as the SIS problem. This would need 308 KB for the first public key and 1 KB for the second.

#### 4.1.2 Learning with errors symmetric encryption

**Parameters:**  $m, n, q \in \mathbb{Z}$

**Key:**  $A \in Z_q^{m \times n}$  ( $n$  is the number of columns for the matrix)

**Input:**  $s \in Z_q^{1 \times n}$ ,  $e \in E^{m \times 1}$  where  $E$  is the error term with  $|E|$  terms

**Output**  $g_A(x) = (As+e) \bmod q$

**Secret key**  $s \in Z_q^{1 \times n}$ , message  $m$  is encoded into binary 0 and 1

**Encryption algorithm:**  $\text{Enc}(s,m) = \{a_i, c_i = g_{ai}(s, e_i) + E(m)\}$  where  $E(m) = q/2 * m$

**Decryption algorithm:**  $\text{Dec}(s, \{a_i, c_i\}) = c_i - a_i * s = e_i + E(m)$  (subtract  $e_i = q/4$  to obtain the message)

Security parameter:  $n$

If the algorithm was to be used in the BAC protocol of the travel documents, then the secret key would be derived in the following manner

$$K_{\text{seed}} = 128\text{msb}(\text{SHA} - 1(\text{DocNo} \parallel \text{DOB} \parallel \text{DOE} \parallel \text{C}))$$

$$K_{\text{ENC}} = 128\text{msb}(\text{SHA} - 1(K_{\text{seed}} \parallel 1))$$

Since, BAC must ensure that the Document number, Date of birth, Date of travel document is available to the terminal, it ensures that travel document and terminal is in contact of less than 9.8 meters, as mentioned in the previous chapter. We obtain the security parameter  $n$  as 128 bits ( $K_{\text{ENC}}$ ) and random numbers in the ring of  $q$  could be added to the private key to obtain the secret key and is exchanged. There is no importance for the public key as it is a common secret key encryption scheme leaving the main evaluation factor as the efficiency of the cryptosystem and security of the scheme.

As mentioned above, total sensitive data that needs to be protected consists of finger prints and iris. The total message size needed to exchange this data would be 10 KB-16KB for the finger print as finger print original resolution is  $128*128*8$  as 16 KB and reduced resolution in 200 Dpi ( $96*96*8$ ) comes to 9 KB with 1 Kb as overhead. Iris data uncompressed consists of 20 KB but with Polar, it needs 2 KB. Hence, total message size needed is around 15 KB with 3 Kb as overhead for communication with the terminal after having performed PACCE, Chip and terminal authentication.

Hence, message  $m$  consists of  $15 \text{ KB} * 1024 * 8 = 122880$  bits and  $n$  is 128 bits.

The first step of encryption would need  $g_{ai}(s, e_i)$  computation. i.e.  $m*n$  and  $1*n$  matrix multiplication. This would cost  $m*n*1 = 15728640 \sim 15$  million computations  $\sim 15$  seconds. Adding and computing the  $E(m) = q/2 * m$  would need  $m*2$  computation =  $122880*2 \sim 1$  second. The overhead would be  $m$  as  $a_i$  needs to be transmitted. (i.e. for every bit of message  $m$ , there is one more bit). Hence, the data transmitted would be 30 KB. But every message is multiplied by  $q/2$  adding a multiplicative factor of  $\log_2(q/2)$ . The total message size of  $m$  would be  $2 * \log_2(q/2) * m$ .  $2^8$  factor for  $q$  would lead to  $14 * m$  i.e. 210 KB total for 30 KB

message. Transmitting 210 KB should not take more than 1 second considering the fast communication speeds.

The first step of decryption would need  $a_i \cdot s$  matrix multiplication ( $m \cdot n$  and  $1 \cdot n$  matrix multiplication) costing approximately 15 million computations  $\sim 15$  seconds. Subtracting it from  $c_i$  and then subtracting  $q/4$  would need  $m \cdot 2 \sim 1$  second of computation.

Considering 15 seconds for encryption, 15 seconds for decryption and 1 second for communication leaves around 15 seconds for verifying the finger print and Iris (total of 45 seconds recommended by ICAO). It can still fulfil the requirement.

#### 4.1.3 Learning with errors asymmetric encryption

Asymmetric communication is the most important cryptosystem with respect to travel document protocols. It can be used in the secured channel between terminal and chip. It can be used to replace the ephemeral Diffie Hellman protocol in PACE and BAC.

**Private key:** Terminal chooses a private key  $s \in \mathbb{Z}_q^{1 \times n}$  and generates a public key and chip performs the same. The public key can be exchanged directly without making use of certificates if terminal and chip authentication is completed successfully.

**Public key:** Encrypt  $p_i = (s, 0)$  to make  $p_1 \dots p_n$  (each generating  $m$  rows hence the public key would be  $m \cdot n$  matrix).  $p_i = ((s \cdot a_i + e_i \bmod q) + (0 \cdot q/2))$

**Encryption algorithm:** Pick a random  $r_i$  and encrypt the message with the public key and add the message part, which would be as follows:

$\sum_{i=0..n} \{p_i \cdot r_i\} + q/2 \cdot m$  (using homomorphic property, it is the same as encrypting the message as shown in the previous chapter)

**Decryption algorithm:**  $\text{Dec}(s, \{a_i, c_i\}) = c_i - a_i \cdot s = e_i + E(m)$  (i.e. same as the symmetric scheme algorithm as the secret key is known but the secret key is known only to the message receiver)

The public key size is an important evaluation factor. The public key size is clearly  $n \cdot m$  matrix. Hence, for a matrix with  $m$  as 122880 and security parameter  $n$  as 128 bits. It will provide maximum security as the length of the key is same as the message size. It would mean a public key size of  $1920 \text{ KB} \cdot \log_2 q$  but by using a modest row size  $m$  as 1024 bits and  $n$  as 128 bits, the public key size would become 128 KB for  $q$  as  $2^8$  ring group. Currently, when passports hold a chip size of 32 KB-144 KB, it might not be possible to hold such a public key size currently. But, when both the terminal and chip have been authenticated, then the chip can compute the public key using terminal resources. For this, the algorithm structure would work in the following manner;

- 1)  $a_i$  can be send by the terminal to the chip
- 2) The chip can generate the private key and compute the public key.
- 3) The terminal can generate the private key and compute the public key

- 4) The public keys can be exchanged for encrypting the messages

Considering the public key computation time, it would take  $m \cdot n \cdot 1$  computations as it involves the matrix multiplication. This would mean around 131K computations and considering million computations per second. The public key can be generated in one second.

The encryption process provides the same complexity as LWE symmetric scheme as the only difference in terms of computation, here, is choosing the random  $r_i$  instead of  $a_i$  and then using public key instead of secret key. Hence,  $m \cdot n \cdot 1$  for encryption process with  $m$  as 1024 bits and  $n$  as 128 bits would need 131K computations and encryption for 1024 bits of message would need less than 0.13 seconds and considering 15 KB of data, it would need 15.7284 million computations needing 15.72 seconds. The decryption process for LWE-symmetric and asymmetric scheme is exactly the same and so computation necessity is also the same. It would need 15 seconds for decryption.

In other words, it can be mentioned that given the constraints of passport, this algorithm provides efficiency of 1.04 seconds for encryption and decryption of 1 KB of data with 1 second over head of computing public key and 1 second overhead for message communication (assuming 128 KB/sec as it uses near field communication, as mentioned in the previous chapter)

#### 4.1.4 GGH/HNF public key cryptosystem- Hermite Normal form

**Private key:** lattice basis  $B$  (short and orthogonal vectors)

**Public key:** Hermite Normal form  $H$  of the private key (Gaussian Elimination)

**Security:** Subset sum problem and special closest vector problem. (Cipher text reveals information about plain text. Hence, closest vector problem is atleast as hard as this special closest vector)

**Encryption process:**  $C = \text{Message encoded } r + v \text{ (lattice point) mod } H$  (random  $(r) + v$  must be reduced diagonally by mod  $H$  and  $v \in H$ )

#### Decryption process

$C \cdot B \text{ mod } H \sim r$ , which is the encoded message vector. The basis inverse is calculated to estimate  $v$  (solving the closest vector problem) which is then subtracted

#### Evaluation

The private key needs  $n \cdot n \cdot \log_2 q$  where  $n$  is the dimensions of the lattice and  $q$  is a large integer to make use of groups to reduce the speed of encryption. For a  $q$  of  $2^8$  and  $n$  as 500 (obtained from practical analysis as mentioned in the second chapter), the private key size would be 244 KB. The passport needs to store this private key in the dedicated file. But, if the terminal and the chip have authenticated each other, then we can let the chip use the computational resources of the terminal.

There is an issue with the same. The chip cannot compute the private key like in the case of Learning with errors. To use this algorithm, the chip needs increased memory storage to store the private key.

The public key is the hermite normal form, which would mean half of the private key size. This would need 122 KB. Transmitting a size of public key of 122 KB cannot be supported even by the PKI of the travel documents as it currently supports 4096 bits of public key.

Encryption process would take  $n+n$  computations for adding the vectors. Mod H would mean an overhead of  $n$  multiplication factor as it needs to be moved across the diagonal. This sums upto 500K computation. It should take less than half a second. Hence, encryption process is quite efficient compared to the previously mentioned scheme

Decryption process makes use of matrix multiplication costing  $n^2.376$  (using coppersmith-winnograd algorithm). The cost would be 2538900 computations. Considering million computations per second, it should cost 2.5 seconds.

#### 4.1.5 GGH/HNF public key cryptosystem- Optimized scheme

Through my research, we could check that there are many possibilities for cost reduction during encryption and decryption processes when using specially for the travel documents. There are four variables; public key, private key, plain text and cipher text with two processes of encryption-decryption with key generation. Continuing my research, we found that we were on the right path and number of papers confirmed the same. [RM '11] covers much details in this area and this section is the summary of the findings in the same paper, which are important for us to understand before progressing further.

The first optimization can be considered as computing  $B^{-1}$  and storing that as the private key instead of computing the private key at each step. The private key size of this key would also remain of the same size. This removes the vector-matrix multiplication needed for decryption dropping the time complexity during decryption.

Also, the private key basis is computed by using the nearest-plan method. Nearest plane algorithm has two steps. First, it applies the LLL reduction to the input lattice. Integer combination of the basis vectors, close to the target vector  $t$  is found. In second step, we identify one translate of the lattice  $L(b_1, \dots, b_{n-1})$  where we suspect that the closest vector to  $t$  resides. It is on this translate that then recurse to compute the closest vector to the target  $t$ . This is the method used in the GGH mentioned previously. This requires a major computational period, which can be reduced by using the rounding off method.

Technically, the round-off step requires the multiplication of integer vector with a rational matrix (all coefficients having a denominator the determinant), then rounded off, we can simplify this by the following method;

- 1) Adding half the determinant to each coefficient

- 2) Dividing by the determinant
- 3) Taking the floor of the result.

Since the determinant (and indeed half the determinant) can be precomputed, depending upon low-level implementation specifics, this method can be implemented in the key computation phase leading to an increased decryption throughput.

Babai had proposed the rounding off method for solving CVP within some specified bound. A significant speed can be realized by calculating and storing the private basis matrix inverse as a matrix embedded in a residue number system (RNS). RNS represents a large integer using a set of smaller integers. It relies on the Chinese remainder theorem of modular arithmetic for its operation. In this way, we are able to perform finite field multiplication inside the rounding step. The idea is that by specifically limiting the primes involved to be strictly smaller than the maximum sized integer representable in a machine word, we are able to perform matrix inverse multiplication using only integer arithmetic. The resulting quotient using the Chinese Remainder Theorem. The comparison between the run-time of both the cases is shown below in figure 9

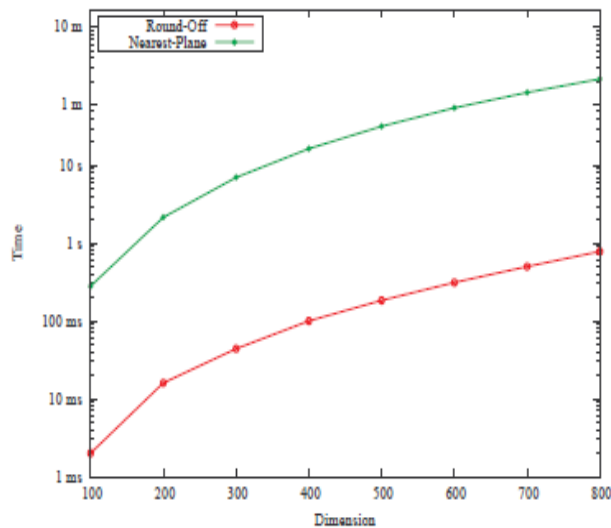


Figure 9: Efficiency comparison between Babai round off and Nearest Plane [RM '11]

Using the calculation mentioned in the figure, for the dimension being 500 for the travel documents, basis inverse computation can be improved from 10 seconds to 100 milliseconds.

Significant drop in decryption speeds over the general GGH cryptosystem is due to the faster multiplication of cipher text and private key inverse inside the RNS using the babai's round off algorithm and Chinese remainder theorem (CRT) though; we need to store each matrix in the Residue Number System, the private key size increases slightly. Figure 10 gives the detailed description of the decryption algorithm time complexity when compared to the normal GGH. It can be seen how the decryption time for a dimension of 500 drops from around 2.4 seconds to less than 200 milliseconds.

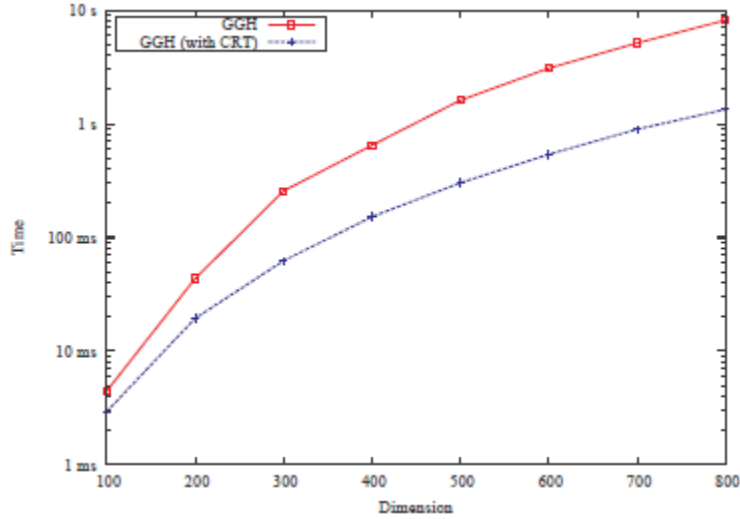


Figure 10: Comparison between GGH-CRT and normal GGH [RM '11]

During my research, I found that Gentry et al uses the method, where by, public key size is reduced by a considerable size by using a vector instead of the matrix.

When the public basis is of a form whereby only one column is populated, the ciphertext is reduced to a form whereby only one coefficient is nonzero. This allows the ciphertext to be represented as a scalar rather than a vector with no loss of information. This optimizes the public key size to a good extend. This method is also used in the signature scheme developed by Plantard et. al.

Since the ciphertext vector is only populated in one position, we are only required to store the corresponding row in the private basis inverse, again without any loss of information. This gives a storage requirement decrease linear in the dimension. In addition, since the time of vector-matrix multiplications is approximately linear in size, we also see a dramatic reduction in decryption times.

Reducing the private key size is of paramount importance as that needs to be stored in the travel document chip, itself. Hence, optimizing the space complexity, here, is of quite some importance.

Both Goldreich et al. (the original paper of GGH scheme) and Micciancio (new scheme focusing on using the hermite normal form for the public key) focus on appropriate lattice basis, which have orthogonality defect, sufficient enough to support polynomial time CVP solutions. Traditionally, such a basis was required to be small and nearly-orthogonal, while being sufficiently randomized to prevent basis recovery through CVP-like and LLL attacks. Through implementing such cryptosystems, however, a new requirement previously overlooked is that such a basis must be computed with implementation efficiency a priority, in order to be more competitive with traditional cryptosystems.

While the LLL-reduced private bases proposed by Micciancio can be seen to exhibit good orthogonality, the LLL-reduction step is computationally expensive. While work is constantly

being done to improve the computational speed of this reduction, orthogonal computations need to be performed faster and we could not find a reason for using the LLL step. The only requirement for CVP is orthogonality.

Rotating the lattice probabilistically through a number of planes in the Hilbert Space in which the lattice is embedded, would create orthogonality. Hilbert space allows the calculus of the two dimension plane to be applied to any dimension. Just to refresh, Orthogonality ensure the inverse of the matrix is same as the transpose of the matrix with all rows and columns being orthogonal unit vectors. Through my research, I found that Alan Glend has proposed a scheme for generating orthogonal matrix in much faster method by using butterfly orthogonal matrix generation(BOM).

The butterfly orthogonal matrices used in Fast Fourier Transforms provide computational efficiency and are uniformly distributed. Firstly, butterfly matrices are computationally inexpensive to generate in fixed precision as they are a product of  $n/2$  matrices of complexity  $O(1)$ . Secondly, the product of a set of butterfly orthogonal matrices is uniformly distributed. Generation of multiple sets of butterfly orthogonal matrices combined with intermediary permutation matrices can be used to ensure the right base, which can act as the private key.

Using this algorithm, there would be significant reductions as is clearly visible after the explanation mentioned in the few above paragraphs. The reductions would clearly be in the public key, private key size and encryption and decryption space.

We are concluding the optimizations from the GGH using hermite normal form in the following paragraphs [RM '11]

- 1) **Computing basis inverse  $B^{-1}$  and storing that as the private key** instead of computing the inverse at each decryption step. (reducing decryption time)
- 2) Computing private key **using Babai round off algorithm and Chinese remainder theorem** (reducing private key computation time)
- 3) **Public basis is of a form whereby only one column is populated** (Reducing public key size)
- 4) Step 3 leading to storing a single row as the private key (Reducing private key size) and using **butter fly orthogonality** (reducing private key computation time even further)

The public key space for 500 dimensions would reduce from 122 KB to 22 KB as single column is populated. The same applies for the private key size as it goes down to 79 KB from 244 KB using scalar vector instead of the matrix.

The encryption speed increases by many folds as the public key size is reduced considerably leading to reduced time for taking the modulo over the public key. The decreased size of the coefficients in the private basis using rotated nearly-orthogonal basis, instead of a LLL-reduced random small basis, leading to reduction in each arithmetic operation in the encryption phase. The encryption speed for this scheme for the dimension of 500 is reduced to 0.014 seconds from 0.13 seconds.

The decryption time (by shifting the inverse computation and reduced private key) is reduced to 0.15 seconds from 2.5 seconds.

To show, that this does not change when higher dimensions or lower dimensions are taken, we show the variance of complexity by altering the dimensions in the following charts.

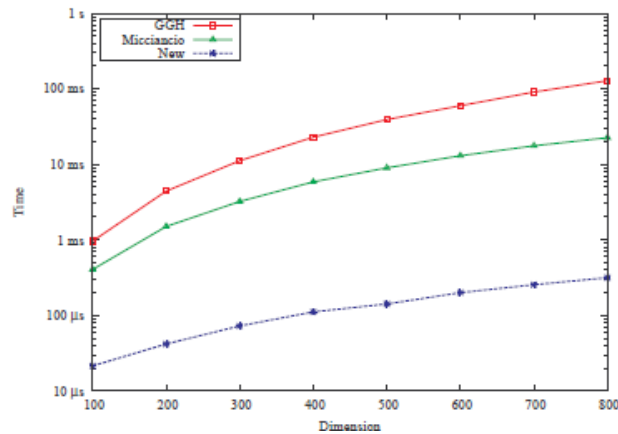


Figure 11: Comparisons between GGH, Micciancio and the new scheme's encryption speed

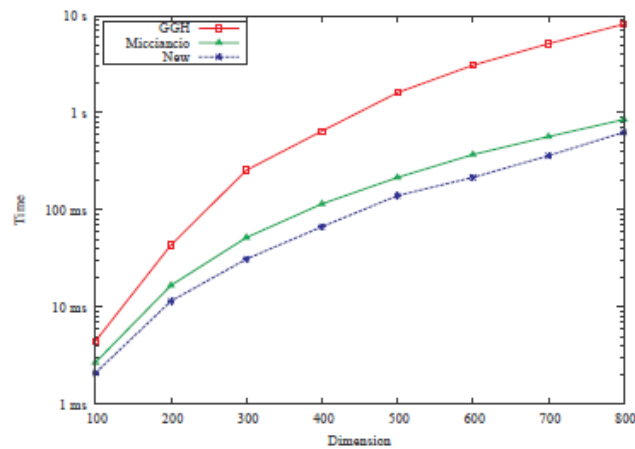


Figure 12: Comparisons between GGH, Micciancio and the new scheme's decryption speed

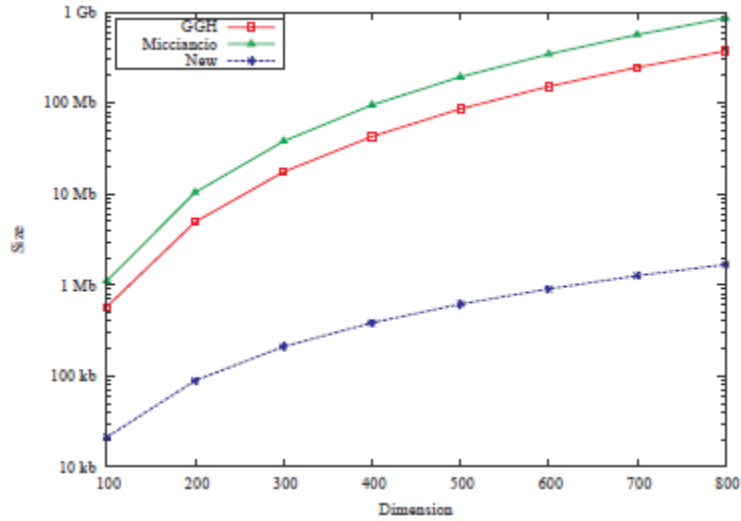


Figure 13: Comparisons between GGH, Micciancio and the new scheme's private key sizes

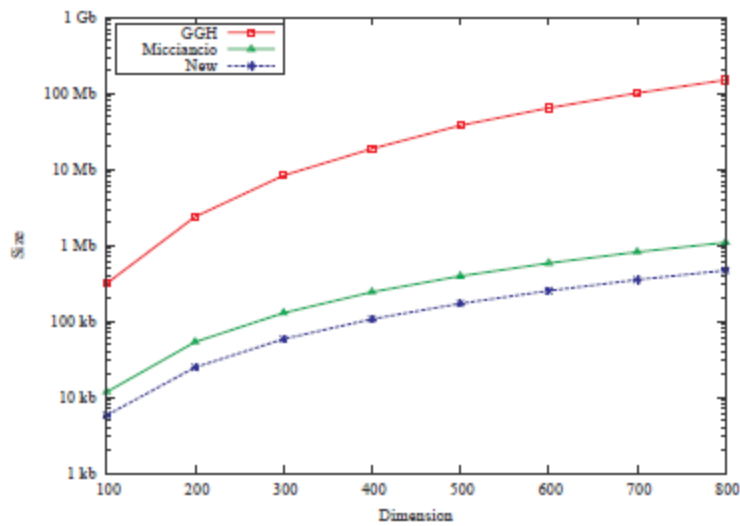


Figure 14: Comparisons between GGH, Micciancio and the new scheme's public key sizes

#### 4.1.6 NTRU public key cryptosystem

**Private key:** short vector  $(f, g)$  which can be seen as the smallest convolutional modular lattice.  $T$  is the inverse element for the private key yielding identity element over multiplication group

**Public key:** HNF form  $H$  of the private key basis.

**Encryption process:** cipher text  $c$  is concatenation of Randomly chosen vector  $r$  and encoded message  $m$  with  $r, m \in \{1, 0, -1\}^n$  and reduced by Modulo public key  $H$

**Decryption process**

The inverse element, private key and the cipher text modulo  $q$  is the decryption step for obtaining the message

$$[T*f]c \bmod q = [T*f]m + [T*f][T*H]r \bmod q = [T*f]m + [T*g]r \bmod q$$

Before reduction modulo  $q$ , the decryption process would allow recovering the exact values of  $[T*f]m$  and  $[T*g]r$  and when reduced over  $P$ , the exact message vector would be produced as  $P$  is the encoder for obtaining the private key such that  $f$  and  $g \in \{P, 0, -P\}^n$ . Hence the message vector would be produced as shown below

$$[T*f]m + [T*g]r \bmod P = I*m + 0*r = m \text{ i.e. the message vector}$$

### **Evaluation process for NTRU**

Assuming the modest security, the dimension of the lattice would be  $n=500$  dimension with  $q$  as  $2^8$  and public key size would clearly be  $n*\log_2 q$  (dimensions are basically over modulo  $q$ )

Public key size:  $\log(2^8)*500=4000$  bits (0.49 KB)

The private key needs to be calculated in a different manner as there are three components making up the private key namely  $f, g$  and  $P$ . With the modest value of  $P$  as 3, we can compute the private key as follows:

$f, g$  would each take  $n$  dimension values over modulo  $P$  leading to  $n*\log_2 P$  for  $f$  and  $n*\log_2 P$  for  $g$ . This would give the private key size as  $2*n*\log_2 P$ . Replacing  $n$  as 5000 and  $P$  as 3, the private key size would be 1585 bits. This would mean private key size of 0.194 KB.

The encrypted text block would obviously be the same of the public key (0.49 KB) and the plain text key block would be  $(n\text{-security bits})*\log_2 P$ . This would mean a cipher text expansion of  $(n*\log_2 q)/(n\text{-security bits})*\log_2 P$ . Using our example, with security bits as 80,  $n$  as 500 and  $q$  as  $2^8$ , the plain text block would be 666 bits. The expansion of cipher text would clearly be 6.06 times.

The encryption and decryption time would be as follows:

Both the encryption and decryption involve matrix multiplication of matrices of  $n$  dimensional size. This would mean time complexity of  $n^2$  operations. Considering the dimension of 500, it would mean 250K operation. Considering million operations per second, it would take 0.25 seconds for encryption and decryption.

Considering that NTRU was used for transmitting the retina and fingerprint data, it would mean 15 KB~122880 bits of data. As obtained in the previous paragraph, 666 bits of plain text offer time complexity of 0.25 seconds, 122880 bits of data would take 46.12 seconds for encryption and the same time for decryption with additional time complexity for message transmission. The total data transmitted would be  $6.06*15$  KB would be 90KB.

From my research, I found that currently, no proof of security supporting NTRU is known, and confidence in the security of the scheme is gained primarily from the best currently known attack as mentioned in the previous chapter.

The core benefit of this scheme is lower private key and public key sizes but there is a significant loss in speed of the encryption and decryption for the scheme, when compared to previously mentioned scheme.

#### 4.1.7 Ajtai-Dwork Cryptosystem

Ajtai presented an efficient cryptosystem whose public key scales in the proportion of  $n^4$  and the encrypted bits remain proportional to  $n$  bits. If a session key in the scale of  $n^2$  can be agreed upon then the public key remains in the proportion of  $n$  bits., which is still around 25K bits for the session key.

Ajtai has also constructed a variable on Ajtai-Dwork Cryptosystem, which reduces the public key size proportional to  $n \log n$  but unfortunately, the security of this cryptosystem is not known to be as strong as that of other lattice-based cryptosystems. Moreover, the primitives are not of using the lattice based cryptography. Hence, it makes no sense of any further evaluation.

#### 4.1.8 Ideal lattices public cryptosystem

**Public key and private key size:** proportional to  $n$

**Bit by bit encryption-decryption.** Provable secured scheme based on SVP

LWE mapping scheme  $(s,e) = G*s + e \bmod q$  works as an injective trap door function.  $G$  is the set of full dimensional vectors and  $e$  is based on the error as discussed in LWE, above

Key is generated by using identity trap algorithm to get  $h$  (as the public key) and the secret  $S$ . The public key and secret key is the subset of  $Z_2^{n \log q}$ . Along with it,  $r$  belonging to  $Z_2^{n \log q}$  is also generated as the public key along with defining the Toeplitz matrix  $M_{GL}$  for allowing faster multiplication.

Public key:  $(h,r)$

Private key:  $S$

##### Encryption process

Sample  $(s,e)$  is computed where  $e$  is computed similarly as in the LWE and  $s$  belongs to  $Z_q^n$ . Cipher text  $c1$  is now computed by substituting in the mapping scheme.  $c1 = h*s + e \bmod q$  is computed. Cipher text  $c2 = \text{Message } M \text{ ex-or}((M_{GL} * s) \bmod 2)$

##### Decryption process

Using the LWE, the receiver can retrieve the  $s$  by  $c1 - S * r \bmod q = s + e$  and removing the error term can retrieve  $s$ , the next step is to retrieve the message by the following computation:

$$\text{Message } m = c2 \text{ ex-or}((M_{GL} * s) \bmod 2)$$

The identity trap door function for computing the public key and secret key:

There exists a probabilistic polynomial time algorithm with the following properties. It takes as inputs  $n$ ,  $\sigma$ ,  $r$ , an odd prime  $q$ , and integers  $m_1, m_2$ . It also takes as input a degree  $n$  polynomial  $f \in \mathbb{Z}[x]$  and random polynomials  $a_1 \in (\mathbb{Z}_q[x]/f)^{m_1}$ . We let  $f = \text{product}_{i \leq t} f_i$  be the factorization of  $f$  over  $\mathbb{Z}_q$ . We let  $\kappa = d_1 + \log q$  and  $m = m_1 + m_2$ . It returns the  $H$  (public key) =  $\text{Transpose}(a_1 \ a_2) \in (\mathbb{Z}_q[x]/f)^\wedge m$  and a basis  $S$  (Secret key) of the lattice [AP'09]

## Evaluation

As the encryption and decryption is clearly ex-or bit by bit, the cost of encryption and decryption for each bit is  $O(1)$

The public key is modulo over 2 but has  $n \log q$  elements. The secret key similarly has  $n \log q$  elements. Considering the value of  $n$  as 500 and  $q$  as  $2^8$ , it would mean the public and private key size of 4000 bits

## 4.2 Efficiencies of the key exchange schemes

### 4.2.1 Our Proposed key exchange scheme

From this crypto system, we discovered that the identity trap door function can also be used in the key exchange algorithm in the following manner:

- 1) The receiver runs the trap door function to compute the private key  $S$  and the public key  $h$
- 2) The sender computes the  $c1 = h * s + e \bmod q$  and transmits to the receiver as mentioned in the encryption process above
- 3) The receiver can compute the  $s$  by  $c1 - S * r \bmod q = s + e$  and removing the error term can retrieve  $s$ , which is the common secret key

This scheme would have almost no overhead and the public and private key size would be 4000 bits and it would be protected by SVP. Solving SVP would mean that  $S$  could be recovered

### 4.2.2 Efficient Key exchange mechanism

Chip authentication and PACE make use of Diffie Hellman algorithms, which need to be replaced by the efficient key exchange mechanism. The Efficient key exchange mechanism is discussed above. It is evaluated in this sub-section.

### Protocol mechanism:

The first level mechanism is the **key exchange mechanism** with a reconciliation mechanism (for additive overhead instead of multiplicative). RSA-KEM (RFC 5990) efficiency can be considered as that is currently used in the internet key exchange (RFC 5996). Considering 128 bits for security is important as computation above  $2^{100}$  is difficult considering the current computational resources.

### Underlying primitive mechanism

Mutual authentication, session completion with common key exchange are the key points

Mutual authentication: SK security with post specified peer model (CK' 02a)

Key exchange: Sign and Mac (Sigma) protocol (Kra'03) (LWE replacing DH key exchange)

### Key encapsulation mechanism:

It would detail the working of key exchange protocol in Sigma

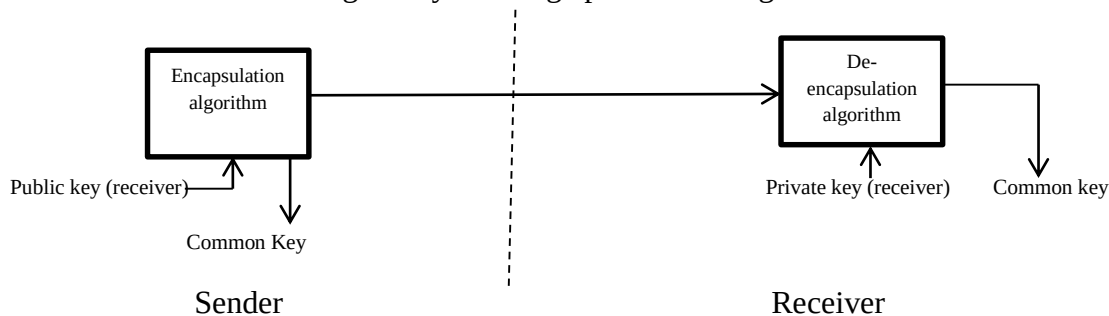


Figure 15: Key Encapsulation mechanism

Offers Passive security: Adversary with public key and encapsulated message, learns nothing about the common key

Needs: Active security: Adversary with encapsulation and decapsulation oracle, public key, encapsulated message learns nothing about the common key.

As explained in the previous chapter, Fujisaki-Okamoto transformation (FO99b) with component of LWE is the only possibility as it maintains the plain text length but Random oracle/coins are needed for the encryption and the Gaussian sampling, which does effect the practical efficiency but offer active security.

### LWE component

Sender encodes a message bit  $u \in (0,1) (\mathbb{Z}_2)$  and  $v = u * \text{floor}(q/2) \in \mathbb{Z}_q$

Receiver recovers  $w \sim v = u * \text{floor}(q/2)$  (Learning from error)

$u = \text{floor function} (2/q * w) \in (0,1) (\mathbb{Z}_2)$  ( $u$  is the secret key)

This can be made even more efficient by using a cross rounding function  $\langle v \rangle_2 = \text{floor}(\text{function}(4/q * v) \bmod 2)$

if  $\langle v \rangle_2$  is known and any  $w \sim v$  is available to receiver,  $v \bmod 2$  can be recovered. Using this concept, the Ring-LWE (ex: cyclotomic rings using principles of associativity) KEM can be built.

The scheme's technical working:

Let  $R = \mathbb{Z}[X]/(X^n + 1)$  and  $R_q = R/q * R$  such that Errors in  $R$  are polynomials with small Gaussian Integer coefficients. Operations in  $R_q$  can be made efficient by using FFT algorithms

Public key is  $(a, b \sim a * s) \in R_q, R_q$  for the private key secret error  $s \in R$

### Encapsulation process

Choose error  $r \in R$ , let  $u = r * a$ ,  $v = r * b \in R$ ,

Output  $C = (u, \langle v \rangle_2) \in (R_q, R_2)$ , key  $k = v \bmod 2 \in R_2$

### DE-capsulation process

As it can be observed that  $u, \langle v \rangle_2$  are known along with  $u, w$ . Hence, obtaining the key is possible and ring-LWE (LPR'10) proves that it is passively secured scheme and when the scheme is passively secured. Fujisaki-Okamoto transformation (FO99b) transforms it into a actively secured scheme. The process of DE-capsulation can simply be calculated as following

$w = u * s \sim r * a * s \sim r * b \sim v$  ( $\sim$  denotes the error)

key  $k = v \bmod 2$

### Evaluation

The whole scheme can be described by the figure mentioned below

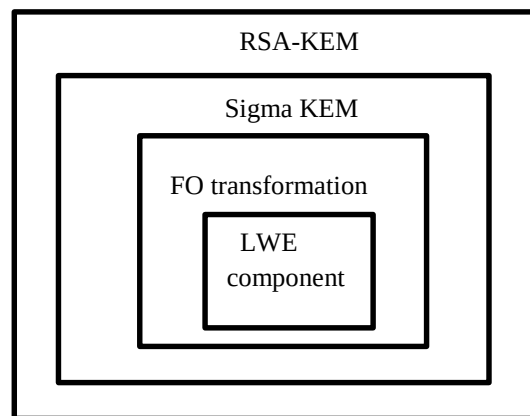


Figure 16: Complete key exchange scheme

It becomes very evident from the diagram that computing the efficiency can be computed for the LWE component and the overhead of each step can be added to obtain the efficiency for the whole scheme.

The complete scheme [PC'14] would work as follows:

The protocol is a digital signature scheme SIG, a key-encapsulation mechanism KEM with key space  $K$ , a pseudorandom function  $F : K \times \{0, 1\} \rightarrow K1$ , and a message authentication code MAC with key space  $K1$  and message space  $\{0, 1\}^*$ . A successful execution of the protocol outputs a secret key in  $K1$ . As in [CK02a],

Trusted public parameters  $pp$  for KEM generated by a trusted party using KEM, and are available to all parties by the certificate authorities.

**Start message** (Initiator  $\rightarrow$  Reciever): (sid, pkI ).

- 1) The protocol is activated by the initiator IDI with a session identifier sid, which must be distinct from all those of prior sessions initiated by IDI .
- 2) The initiator generates a fresh keypair  $(pkI, skI) \leftarrow$  public parameters, stores it as the state of the session (IDI, sid), and sends the above message to the responder.

**Response message** (Receiver  $\rightarrow$  Initiator): (sid, c, IDR, Sign(using Sig)(1, sid, pkI, c), MAC.Tagk1 (1, sid, IDR)).

- 1) A party IDR (receiver) receives a start message (sid, pkI ),
- 2) It generates an encapsulation and key  $(c, k) \leftarrow$  KEM.Encapsulation( $pp, pkI$ ), derives key  $k0$  and  $k1$  from the key, and erases the values  $pkI$  and  $k$  from its memory, storing  $(k0, k1)$  as the state of the session.
- 3) It generates and sends the above response message, where SIG.SignR is computed using its long-term signing private key, and MAC.Tag is computed using key  $k1$

**Finish message** (I  $\rightarrow$  R): (sid, IDI, SIG.SignI (0, sid, c, pkI), MAC.Tagk1 (0, sid, IDI)).

- 1) When party IDI receives the (first) response message (sid, c, IDR,  $\sigma_R$ ,  $\tau_R$ ) having session identifier sid
- 2) It looks up the state  $(pkI, skI)$  associated with session sid and computes  $k =$  KEM.Decaps( $skI, c$ ) and computed  $k0$  and  $k1$  from them. It then retrieves the signature verification key of IDR and uses that key to verify the signature  $\sigma_R$  on the message tuple  $(1, sid, pkI, c)$ , and also verifies the MAC tag  $\tau_R$  on the message tuple  $(1, sid, IDR)$  under key  $k1$ .
- 3) It then produces public session output (IDI, sid, IDR) and session secret output  $k0$ .

### Responder completion

- 1) Party IDR receives the (first) finish message (sid, IDI,  $\sigma_I$ ,  $\tau_I$ ) having session identifier sid,

- 2) It looks up the state  $(k_0, k_1)$  associated with session  $sid$ . It then retrieves the signature verification key of IDI and uses that key to verify the signature  $\sigma_I$  on the message tuple  $(0, sid, c, pk_I)$ , and also verifies the MAC tag  $\tau_I$  on the message tuple  $(0, sid, IDI)$  under key  $k_1$ .
- 3) It then completes the session with public session output  $(IDR, sid, IDI)$  and secret session output  $k_0$

It can clearly be seen that it is computationally quite fast as MAC, Sign and Matrix multiplication (KEM encapsulation and decapsulation) are performed

Rest of the steps use bit to bit computation allowing the speed to scale to  $O(1)$  for each bit. Hence, this is a scheme that is quite efficient for use in the PACE, chip authentication and other protocols.

Hence it is a little more efficient than our proposed digital signature, above.

### 4.3 Efficiencies of other cryptosystems

All the other cryptosystem mentioned are highly inefficient due to high public key sizes as mentioned in the table below:

| Cryptosystem                         | Public key       | Expansion   | Worst-case problem                                                                  | Approximation        |
|--------------------------------------|------------------|-------------|-------------------------------------------------------------------------------------|----------------------|
| LWE, $q = 2^{O(n)}$                  | $O(n^4)$         | $O(\log n)$ | GapSVP                                                                              | $\tilde{O}(n^2)$     |
| Ajtai-Dwork [AD97, AD07]             | $O(n^4)$         | $O(n)$      | unique-SVP*                                                                         | $\tilde{O}(n^2)$     |
| Regev [Reg04]                        | $O(n^4)$         | $O(n^2)$    | unique-SVP*                                                                         | $\tilde{O}(n^{1.5})$ |
| LWE, $q = \text{poly}(n)$            | $\tilde{O}(n^2)$ | $O(\log n)$ | $\frac{\text{poly}(n)\text{-to-}\gamma\text{-GapSVP}}{\text{GapSVP/SVP (quantum)}}$ | $\tilde{O}(n^{1.5})$ |
| old CCA: $q = \text{poly}(n)$ [PW08] | $n^{2+\delta}$   | $O(\log n)$ | same two choices $\uparrow$                                                         | $n^{5+\delta}$       |
| new CCA: $q = \text{poly}(n)$        | $n^{2+\delta}$   | $O(\log n)$ | same two choices $\uparrow$                                                         | $\tilde{O}(n^2)$     |
| new CCA: $q = 2^{O(n)}$              | $n^{4+\delta}$   | $O(\log n)$ | GapSVP                                                                              | $\tilde{O}(n^3)$     |

Figure 17: Tabular details of algorithms [PC'09]

It can clearly be seen that when public key size is of the order  $n^4$ , it becomes of the order of 670 MB, which is clearly impossible to store in a chip or to transmit via the certificate authorities. Hence, further analysis of these schemes is not of much order.

# Chapter 5

## Proposed Protocols for travel documents

---

We had described all the candidates for cryptosystem and key exchange schemes of lattice based cryptography in the second chapter. We had described all the protocols, properties of protocols and constraints of the travel documents in the third chapter. Fourth chapter was focused on our work of evaluation of these cryptosystems.

Hence, in this chapter, we would focus on the proposed changes in the protocols making use of these algorithms (cryptosystems and key exchanges) after discussing the elected candidates for cryptosystem.

### 5.1 Elected cryptosystems-public schemes

Some cryptosystems candidates have already been checked out in the previous chapter. This section considers the four shortlisted schemes (LWE-asymmetric, proposed GGH, NTRU, Ideal lattice cryptosystem)

#### 5.1.1 Learning with Errors asymmetric scheme

Learning with errors asymmetric scheme has a public key size of  $m \cdot n \cdot \log_2 q$  (  $m$  as 1024 bits, security parameter  $n$  as 128 and  $q$  as  $2^8$  ) would lead to a public key of 128 KB, which cannot be transferred by the certificates. Hence, our proposal is to establish a mutual authentication

and then exchange the public keys after generating for each session and it takes just 1.04 seconds for 1 KB of data encryption.

The simplest method could be to use a digital signature to establish mutual authentication and sign the generated public key.

When the Small Integer solution digital signature is used, the public key size for signing 128 KB would be large to maintain maximum security. When the public key of LWE is hashed by SHA-128, the public key to ensure maximum security would be  $128 \times 128 \times \log(2^8)$  ( $m \times \text{security parameter} \times \log q$ ) and second public key would be  $(m \times \log q)$  giving 3.5 KB and 1024 bits for the second public key. Transmitting a public key of 3.5 KB is still quite hard, as mentioned above, the single row can be stored as public key and left circulars could be performed at appropriate location.

For example: Binaryencodedof(1X2X4) would mean left circular of 1 in the first row, 2 left circular in second row and 4 left circulars in fourth row and similarly public key can be generated and X could be the check sum or the padding to differentiate the rows

This would bring down the public key to  $(m \times \log q)$  leading to 1024 bits for first public key and 1024 bits for the second public key, which can be transmitted by the certificate authorities.

**Our proposed LWE asymmetric scheme** would work as follows:

- 1) Chip computes the LWE public key and private key
- 2) Terminal computes the LWE public key and private key
- 3)  $1024 \times 2$  bits for 2 public keys (chip1, chip2) can be obtained by terminal from Certificate
- 4)  $1024 \times 2$  bits for 2 public keys (terminal1, terminal2) can be obtained by chip from Certificate
- 5) Left circulars can be performed as mentioned above and in the previous chapter to generate complete public keys (chipfull from chip1, chip2, terminalfull from terminal1, terminal2)
- 6) The chip can send LWE chip public keys with hashed chip public key signed by terminalfull and terminal2 public key using SIS-Digital signature
- 7) The terminal can send LWE terminal public keys with hashed terminal public key signed by chipfull and chip2 public key using SIS-Digital signature
- 8) Both public keys hashes can be verified with DG and terminal and chip public keys for LWE can be exchanged
- 9) Asymmetric encryption using LWE can be used

The parameters and technical function has been explained in the previous chapter

### 5.1.2 GGH scheme

Private key size is 244 KB for modest security measure of 500 bits and  $q$  as  $2^8(n^2 \times \log q)$  and private key is half of it (HNF form) as 122 KB. Encryption speed is  $n^2 + n^2$  (leading to half second) and decryption speed as  $n^{2.377}$  (2.5 seconds)

This algorithm offers less through put for decryption as well as large public and private key. Hence, we have proposed some changes in the scheme to improve through put and reduce public and private key (in previous chapter), which are as follows

- 1) Computing basis inverse  $B^{-1}$  and storing that as the private key instead of computing the inverse at each decryption step. (reducing decryption time)
- 2) Computing private key using Babai round off algorithm and Chinese reminder theorem (reducing private key computation time)
- 3) Public basis is of a form whereby only one column is populated (Reducing public key size)
- 4) Step 3 leading to storing a single row as the private key (Reducing private key size) and using butter fly orthogonality (reducing private key computation time even further)

The public key space for 500 dimensions would reduce from 122 KB to 22 KB as single column is populated. The same applies for the private key size as it goes down to 79 KB from 244 KB using scalar vector instead of the matrix. The encryption speed for this scheme for the dimension of 500 is reduced to 0.014 seconds from 0.13 seconds. The decryption time (by shifting the inverse computation and reduced private key) is reduced to 0.15 seconds from 2.5 seconds. Yet, the public key sizes are quite large, hence the method using Digital signature still needs to be used to transmit the public key after mutual authentication (achievable by SIS-digital signature)

### 5.1.3 NTRU

There is no security proof of NTRU . It offers public key size of  $n \log q$  leading to 4000 bits for ( $n=500$ ) and 1585 bits for private key ( $n \cdot \log P$  where  $p=3$  is standard as mentioned with quotation above)

Encryption-decryption throughput is quite bad with  $n^2$  leading to 0.25 seconds for each

### 5.1.4 Ideal lattice Cryptosystem

Public key and private key costs  $n \log q$  (4000 bits for  $n=500$ ) and it is a bit by bit encryption with security proof of SVP. The cryptosystem has the underlying primitive of LWE and SVP.

The following table summarizes the four cryptosystems for  $n=128$  for LWE and  $n=500$  for all other schemes,  $m$  as 128 bits,  $q=2^8$  and  $P=3$

| Scheme                     | Number of public keys | Public key size                                            | Private key size                                                  | Enc time                                             | Dec time                                            |
|----------------------------|-----------------------|------------------------------------------------------------|-------------------------------------------------------------------|------------------------------------------------------|-----------------------------------------------------|
| LWE+Digital signature(DS)  | 1(LWE) and 2 for DS   | $m*n*\log_2 q$ (128 KB)+ $2*(m*\log q)$ (1024 bits for DS) | $n\log q$ (4000 bits)+ private keys for DS $m*m\log q+m*\log q$   | $m*n$ +(1.04 sec)+ throughput for DS                 | $m*n + m^2$ +throughput for DS                      |
| Our GGH + DS               | 1(GGH) and 2 for DS   | 22KB+ $2*(m*\log q)$ (1024 bits for DS)                    | $n^2*\log q/2$ (122KB) + private keys for DS $m*m\log q+m*\log q$ | $n(n+n)$ +throughput for DS-reduced to 0.013 seconds | $n^2.3678$ (reduced to 0.5 sec) + throughput for DS |
| NTRU                       | 1                     | $n\log q$ (4000 bits)                                      | $N*\log P$ (1585 bits)                                            | $n^2$ (2.5 sec)                                      | $n^2$ (2.5 sec)                                     |
| Ideal lattice cryptosystem | 2                     | $n\log q$ (4000 bits)                                      | $n\log q$ (4000 bits)                                             | $O(1)$                                               | $O(1)$                                              |

Table 1: Tabular details of candidates

From the table, it becomes clear that the **ideal lattice cryptosystem and our proposed GGH (using SIS-DS)** is the optimal candidate for using it in the electronic travel documents.

With regards to the key exchange scheme, we have already established above that our proposed key exchange uses 4000 bits for public and private key whereas the KEM scheme, mentioned in the previous chapter makes sue of 128 bits for public key and private key and offers  $O(1)$  computations

The symmetric scheme using LWE makes use of  $n\log q$ (4000 bits) for the encryption and offers a through put of  $m*n + m^2$  for encryption and decryption as mentioned in the previous chapter leading to a good throughput. Hence, it can be used as the symmetric scheme algorithm in BAC, PACE

## 5.2 Proposed Protocols in the travel documents

### 5.2.1 Proposed Public key Infrastructure

As shown above, the digital signature (with our proposed changes) needs 2 public keys of 1024 bits. The scheme can then ensure transmission of large public keys between chip and terminal by signing the key. To ensure certain level of mutual authentication, the public keys need to be transmitted by the PKI of the travel documents, which would work as follows:

- 1) Country Signing Certificate Authorities (CSCA) generates and stores a key ( $K_{Pu_{CSCA}}$ ,  $K_{Pr_{CSCA}}$ )
- 2) It signs the Document Singer (DS) in the country using its private key
- 3) DS signs the security data element of the passport issued by itself. Security data element is the group of individual hashes collectively signed. Along with it, passport stores the dedicated file (encryption, MAC and private key of the e-passport bearer) and Elementary file (data groups and DG)
- 4) This would need the 2 DS public key to verify the travel documents obtainable from ICAO's public key directory

The CSCA would sign the DS in the following manner:

**Inputs:**  $X \in \{0,1\}^{128 \times 128}$  and  $x \in \{0,1\}^{1 \times 128}$ .

$$f_A(x) = Ax \bmod q, \quad A \in Z_q^{128 \times 128}$$

**Public key:**  $f_A(X) = |f_A(X_1), \dots, f_A(X_m)| \in Z_q^{1 \times 128}$  (each of the  $X_i$  is a matrix of  $1 \times 128$  elements) and  $f_A(x) \in Z_q^{1 \times 128}$ . (As mentioned above, our proposal is to store the first row and left circular digits with check sum would be stored to reduce the size and ensure that random left circulars are used to strengthen the public key)

**Message**  $M \in \{0,1\}^{1 \times 128}$  (The certificate of DS)

**Signature algorithm:**  $f_A(y) = X * M + x * X * M \bmod q$  (output of 128 columns)  $\in \{0,1\}^{1 \times 128}$ .

One time digital Signature for message m:  $f_A(y)$

Verifying nation can obtain the two public keys  $f_A(X)$ ,  $f_A(x)$  and compute  $f_A(X) * m + f_A(x)$  and check if it matches the signature of the CSCA, to ensure that DS is valid. The CVCA would also work down the hierarchy in the same manner upto the inspection level.

Similarly, Digital signature would also be used by document signer as it would also issue its 2 public keys via ICAO and sign the passport, it issues. The Message would be the collective hashes of individual data groups of the mandatory data of passport (i.e. excluding iris and finger print DG as mentioned in the chapter 3).

### 5.2.2 Proposal for Passive authentication

The inspection system retrieves the certificate of the issuing document singer, which would give inspection system access to both the public keys;  $f_{DS}(X)$ ,  $f_{DS}(x)$  as mentioned above

- 1) The inspection system receives the individual hashes (message M). It computes  $f_{DS}(X) * M + f_{DS}(x)$ , if it matches the signature of the document signer, then it goes to the next step
- 2) It has been established that the signature is correct but now, inspection system needs to check if the hashes were also not modified. The terminal/inspection device computes the hashes of the mandatory data and compares with data stored in the SOD (which is signed by Document signer)

When validated, it can be established that the data in the RFID was not manipulated

### 5.2.3 Proposed Active authentication (optional protocol)

**$KP_{uAA1}(f_{pass}(X))$ ,  $KP_{uAA2}(f_{pass}(x))$  are the two public keys** stored in the Data group 15 of the LDS (elementary file) and its hash representation in the SOD. The corresponding private key  **$KP_{rAA1}(X)$ ,  $KP_{rAA2}(x)$**  is stored in the secure section of memory (dedicated file). X is again stored in the same way. The first row of the matrix then for each row, corresponding left circulars are known. X and x are as explained above. In this way, both the private keys would be of the size 1024 bits

- 1) The terminal Reader sends a randomly generated 64 bit string (R) to the chip
- 2) The Tag signs this string by  $X * M + x * X * M \bmod q$  (q being  $2^8$ , x and X as private keys, M as the randomly generated key) and sends this signature to the terminal reader
- 3) The terminal reader obtains the public key  $KP_{uAA1}$ ,  $KP_{uAA2}$  stored in Data Group 15
- 4) The terminal reader verifies the correctness of the signed string using its knowledge of R by computing  $KP_{uAA1} * m + KP_{uAA2}$

### 5.2.4 Proposed Basic Access Control

The focus is not on increasing the entropy (an issue as mentioned above in chapter 3) but to make the protocol quantum resistant.

Hence, the manner in which the access keys are generated would still remain the same but the LWE-symmetric scheme as discussed in chapter 4 would be used.

LWE symmetric scheme would work in the following manner.

Parameters:  $m, n, q \in \mathbb{Z}$  ( $m=n=128$ ,  $q=2^8$  as established in the previous chapter)

Key:  $A \in \mathbb{Z}_q^{m \times n}$  ( $n$  is the number of columns for the matrix)

Input:  $s \in \mathbb{Z}_q^{1 \times n}$ ,  $e \in E^{m \times 1}$  where  $E$  is the error term with  $|E|$  terms

Output  $g_A(x) = (As+e) \bmod q$

Secret key  $s \in \mathbb{Z}_q^{1 \times n}$ , message  $m$  is encoded into binary  $o$  and  $1$

Encryption algorithm:  $Enc(s,m) = \{a_i, c_i = g_{a_i}(s, e_i) + E(m)\}$  where  $E(m) = q/2 * m$

Decryption algorithm:  $Dec(s, \{a_i, c_i\}) = c_i - a_i * s = e_i + E(m)$  (subtract  $e_i = q/4$  to obtain the message)

The core problem being that the secret key  $s$  needs to be exchanged. This can be achieved by the key exchange algorithm or can be generated. BAC would allow both the parties (terminal and chip) to generate the key

The Access Keys ( $K_{ENC}$ ,  $K_{MAC}$ ) are derived from the following data available on the MRZ: The Passport Number (Doc No), Date of Birth of the Passport Holder (DOB), Valid Until Date of the Passport (DOE), 3 Check Digits (C). Terminal when within 9.8 cm (derived in chapter 3) would have access to this data proving that it is in contact to the passport.

$K_{seed} = 128\text{msb}(\text{SHA} - 1(\text{DocNo} \parallel \text{DOB} \parallel \text{DOE} \parallel \text{C}))$

$K_{ENC} = 128\text{msb}(\text{SHA} - 1(K_{seed} \parallel 1))$

$K_{MAC} = 128\text{msb}(\text{SHA} - 1(K_{seed} \parallel 2))$

The keys  $K_{ENC}$ ,  $K_{MAC}$  obtained would be only of 128 bits. Hence, **a pseudo random number in the group of  $q$  ( $2^8$ )** would need to be added to it and then exchanged.

The terminal reader will now enter a Challenge-Response mechanism (described below) to prove possession of the access keys and to derive a session key. It replaces the module of DES3 with the LWE symmetric scheme, whose technical description is discussed in the previous paragraph in this section

- 1) The chip generates and sends the terminal reader a 64 bit string ( $R_T$ ) with the MAC using the key  $K_{MAC}$ .
- 2) The terminal reader receives  $R_T$ , verifies the MAC (the key  $K_{MAC}$ ) and **generates the 128 bits random string  $R$  from  $R_T$**  (as shown above, there are number of ways to do so. One instance could be SHA-128 hash of  $R_T$ ) and generates two random 64 bit strings ( $R_R$ ,  $K_R$ ).
- 3)  **$K = K_{ENC} * R \bmod q$  is obtained as the LWE secret encryption key**
- 4) The terminal reader encrypts  $R_R \parallel R_T \parallel K_R$  using symmetric encryption algorithm and the key  $K$ .
- 5) The terminal reader computes the MAC of the cipher with the key  $K_{MAC}$ .
- 6) The terminal reader sends the cipher and the MAC to the chip.
- 7) **The reader generates the 128 bits random string  $R$  from  $R_T$  and computes  $K = K_{ENC} * (R) \bmod q$  as the symmetric encryption key**

- 8) The reader checks the MAC, decrypts the cipher (which proves that terminal reader has access to epassport and has also received the string  $R_T$  else  $K$  and  $K_{MAC}$  wouldn't have been calculated). It verifies the correctness of  $R_T$  and then extracts  $K_R$ .
- 9) The chip generates another 64 bit random string  $K_T$ .
- 10) The chip encrypts  $R_T || R_R || K_T$  using the LWE-symmetric algorithm and  $K$ , where  $R_R$  has remained secret and is randomly send by the terminal reader
- 11) The chip computes the MAC of the cipher with the key  $K_{MAC}$
- 12) The chip sends the cipher and the MAC to the terminal reader.
- 13) The terminal reader checks the MAC, decrypts the cipher. It verifies the correctness of  $R_R$  and then extracts  $K_T$ , where by checking  $R_R$ , (chip is authenticated)
- 14) Both the terminal reader and the chip compute the session key seed ( $K_{seed}$ ) as  $K_R * K_T \bmod q$ , which can be used as the session key for the LWE symmetric scheme. Now both parties generate a session MAC key  $K_M$  as shown below, and since, it is based on  $K_R$  and  $K_T$ , the key will always be session keys

$$K_M = 128\text{msb}(\text{SHA} - 1(K_{seed}))$$

Communication is secured using the above LWE encryption and MAC key where the strength lies on the Passport Number (Doc No), Date of Birth of the Passport Holder (DOB), Valid Until Date of the Passport (DOE), 3 Check Digits (C).

### 5.2.5 Proposed Chip Authentication (mandatory EAC specification)

The Chip Authentication protocol is a mandatory protocol in the EAC specifications. It aims to replace Active Authentication as a mechanism to detect cloned ePassports. The original chip authentication makes use of Diffie Hellman (DH) algorithm, which can be replaced by the key exchange algorithm (mentioned in the previous chapter) Chip Authentication establishes a new pair of encryption and MAC keys to replace weak BAC derived session keys.

ePassport chip will have two Chip Authentication public key (in Data Group 14) and two private key (in secure memory) as mentioned in the authentication proposal phase. The process of Chip Authentication is described below:

- 1) The chip generates a **fresh keypair** ( $pkI, skI$ )  $\leftarrow$  **public parameters (derived from two public keys in DG14)**, stores it as the state of the session (chipid, sesssionid), and sends the keypair to the Terminal
- 2) **The terminal reader generates an encapsulation and key** ( $c, k$ )  $\leftarrow$   $\text{KEM.Encapsulation}(pp, pkI)$ , derives key  $k_0$  and  $k_1$  from the key, and erases the values  $pkI$  and  $k$  from its memory, storing ( $k_0, k_1$ ) as the state of the session.
- 3) The terminal also performs the passive authentication protocol to ensure that the public parameters are valid.  $\text{SIG.SignR}$  is computed using the terminal's two private keys (whose public key is signed by the document verifier certificate) and  $\text{MAC.Tag}$  is computed using key  $k_1$ . The terminal sends the encapsulation, Signature and Mac to the chip

- 4) The **chip** looks up the state (chipid,sessionid) associated with session 'sessionid' and **computes**  $k = \text{KEM.Decaps}(skI, c)$  (two secret key  $s$  as the input stored in the secured memory of the chip) and computed  $k_0$  and  $k_1$  from them. It then retrieves the signature verification key of the terminal from the document verifier certificate and uses that key to verify the signature and also verifies the MAC using  $k_1$ .
- 5) The Reader and Tag can now generate a **new seed key (Kseed) using this shared key K**.

The technical working of Sig.SingR, Mac.Tag, KEM.encapsulation, KEM.Decaps are already expanded in the key encapsulation mechanism section in the previous chapter.

The step 4 in the authentication proves that the terminal authentication as without knowing the secret keys, it would have not derived the key  $k$  and  $k_{seed}$  in the following step.

### 5.2.6 Proposed Terminal Authentication

The Terminal Authentication protocol is a protocol necessary, if access to IRIS and Fingerprint is required.

Using the proposed infrastructure, we propose the process of Terminal Authentication as described below;

- 1) The terminal reader sends an **Inspection System certificate** to the chip (which was received from the local DV) along with the **DV's** certificate (which was received from the CVCA, which will validate that the inspection system certificate is valid).
- 2) The chip inspects the certificates and extracts both the public keys of the terminal reader from the Inspection System certificate.
- 3) The chip generates a random string ( $R$ ) and sends it to the terminal reader.
- 4) The terminal reader computes the hash of both the public keys  $KP_{uAA1}$ ,  $KP_{uAA2}$
- 5) The terminal reader **signs the message**  $(R || \text{SHA-1}(KP_{uAA1}) || KP_{uAA2})$  with both its private key by  $\text{TerminalP}_{r1} * (R || \text{SHA-1}(KP_{uAA1}) || KP_{uAA2}) + \text{TerminalP}_{r2} * \text{TerminalP}_{r1} * (R || \text{SHA-1}(KP_{uAA1}) || KP_{uAA2}) \bmod q$  (**SIS-Digital Signature**)
- 6) The chip verifies the correctness of  $R$  and  $KP_{uAA1}$ ,  $KP_{uAA2}$  by computing  $\text{TerminalP}_{u1} * (R || \text{SHA-1}(KP_{uAA1}) || KP_{uAA2}) + \text{TerminalP}_{u2}$  (obtained from the certificate from step 1) and matches with the received signature and grants access to sensitive data.

### 5.2.7 Proposed Password Authenticated Connection Establishment(PACE)

PACE replaces the Basic Access Control protocol as a mechanism allowing chip to verify that the terminal reader has authorized access to the electronic passport. The chip and the terminal reader share a common password ( $\pi$ ) which is obtained by the Diffie-Hellman key agreement protocol to provide a stronger session key. The key exchange algorithm can be replaced with

the key encapsulation scheme as mentioned above. The entire proposed process would be as follows based on KEM as discussed above;

- 1) The chip encrypts random nonce using the key  $K\pi$ . Here,  $K\pi$  is  $\text{SHA-1}(\pi||3)$ .
- 2) The chip sends the encrypted nonce. The chip generates a **fresh keypair** ( $pkI, skI$ )  $\leftarrow$  public parameters (derived from two public keys in DG14), stores it as the state of the session (chipid, sessionid), and sends the keypair to the Terminal
- 3) The terminal reader decrypts the random nonce using the key  $K\pi$  derived from  $\pi$
- 4) The terminal reader generates an **encapsulation and key** ( $c, k$ )  $\leftarrow \text{KEM.Encapsulation}(pp, pkI)$ , derives key  $k0$  and  $k1$  from the key, and erases the values  $pkI$  and  $k$  from its memory, storing ( $k0, k1$ ) as the state of the session.
- 5) The terminal also performs the passive authentication protocol to ensure that the public parameters are valid.  $\text{SIG.SignR}$  is computed using the terminal's two private keys (whose public key is signed by the document verifier certificate) and  $\text{MAC.Tag}$  is computed using key  $k1$ . The terminal sends the encapsulation, Signature and Mac to the chip
- 6) The **chip** looks up the state (chipid, sessionid) associated with session 'sessionid' and **computes**  $k = \text{KEM.Decaps}(skI, c)$  (two secret key  $s$  as the input stored in the secured memory of the chip) and computed  $k0$  and  $k1$  from them. It then retrieves the signature verification key of the terminal from the document verifier certificate and uses that key to verify the signature and also verifies the MAC using  $k1$ .
- 7) The Reader and Tag can now generate a new seed key (**Kseed**) using this shared key  $K$  and MAC using  $k1$
- 8) The terminal reader computes an authentication token: **Token-TerminalReader** =  $\text{MAC}(K1, (\text{TerminalP}_{u1} || \text{TerminalP}_{u2}))$  where both the keys are public keys of the terminal
- 9) The chip computes an **authentication token**:  $\text{Token-chip} = \text{MAC}(K1, (KP_{uAA1} || KP_{uAA2}))$

When chip received token-terminal reader, it can authenticate by obtaining the **public keys from the DV and Inspection device certificate**. Similarly, when terminal reader receives the token-chip, it can verify by checking **with the public keys in the DG15**. When common key  $k1$  is used for MAC, a secret communication channel has been established and PACE has been successful

$\Pi$  password used in the PACE could be printed on passport/dynamic password (Card Access Number (CAN)) or Machine readable zone (MRZ) passwords.

### 5.3 Proposed Usage of Asymmetric cryptosystem

After a secret communication is established for transmitting sensitive information (including IRIS and Fingerprint data), asymmetric cryptosystem becomes necessary. In the first step of PACE, the chip encrypts the random nonce and terminal decrypts it. This is also possible if

chip encrypts using public key of the terminal reader. The terminal reader decrypts using its own private key

As derived earlier in this chapter, Ideal lattice cryptosystem and our proposed GGH scheme embedded with Digital signature is the most efficient lattice based cryptosystem.

It is quite evident how the cryptosystem would be used with regards to the travel documents. To illustrate the usage of the cryptosystem, we are figuring out the steps of our proposed GGH scheme with Digital signature (Similarly, Ideal lattice based cryptosystem as explained in the 4<sup>th</sup> chapter could also be used)

- 1) The chip computes its private key basis and computes the inverse  $B^{-1}$  and holds that as the private key using Babai round off algorithm, butterfly orthogonality and Chinese remainder time
  - 2) The chip computes its Public key by computing Hermite normal form of the basis
  - 3) The chip stores a single row of  $B^{-1}$  for decryption as private key Chip  $B^{-1}$
  - 4) The chip takes the first column of the public key as ChipH and sends it to the Terminal reader after signing it with the two private keys  $KP_{rAA1}(X)$ ,  $KP_{rAA2}(x)$  is stored in the secure section of memory (dedicated file) by  $KP_{rAA1}(X)*ChipH + KP_{rAA2}(x)* KP_{rAA1}(X)*ChipH \text{ mod } q$ . Hashes of chipH can be used in signature
- 
- 1) The terminal similarly computes its private key basis and computes the inverse  $LB^{-1}$  and holds that as the private key using Babai round off algorithm, butterfly orthogonality and Chinese remainder time
  - 2) The terminal computes its Public key by computing Hermite normal form of the basis
  - 3) The terminal stores a single row of  $B^{-1}$  for decryption as private key terminal  $B^{-1}$
  - 5) The terminal takes the first column of the public key as terminalH and sends it to the Terminal reader after signing it with the two private keys  $TerminalP_{r1}$ ,  $TerminalP_{r2}$  by  $TerminalP_{r1}*terminalH + TerminalP_{r2}* TerminalP_{r1}*terminalH \text{ mod } q$ . Hashes of terminalH can be used in signature

The terminal can validate the received GGH chip public key (~22 KB) for the GGH scheme by obtaining the  $KP_{uAA1}$ ,  $KP_{uAA2}$  stored in the DG14 of the Logical object structure. The terminal has to just compute  $KP_{uAA1}*GGHChippublickey$  (~22KB hashes)+  $KP_{uAA2}$  and match with the signature received

The chip can validate the received GGH terminal public key (~22 KB) for the GGH scheme by obtaining the  $TerminalP_{u2}$ ,  $TerminalP_{ru1}$  (obtainable from the DV certificate and Inspection certificate). The terminal has to just compute  $TerminalP_{ru1}*terminalpublickey$  (~22KB hashes)+  $TerminalP_{ru2}$  (Obtained from the DV and IS certificate) and match with the signature received

After completing the above mentioned process, encryption and decryption can take place in the following manner as explained in the previous chapter as well

**Encryption process:**  $C = \text{Message encoded } r + v \text{ (lattice point) mod } H$  (random  $(r) + v$  must be reduced diagonally by mod  $H$  and  $v \in H$ )

**Decryption process**

$C * B^{-1} \text{ mod } H \sim r$ , which is the encoded message vector.

## Chapter 6

# Conclusion

---

We started with the motivation part to explain the reasons for using quantum resistant cryptosystems and derived a justification for using lattice cryptosystem among other classes of algorithms. We also explained the motivation for replacing protocols in the travel documents. We also derived the logic as to why lattice based problems cannot be solved using the quantum computers and classical computers.

In the following chapter, we showed the concept behind the different classes of quantum attack resistant algorithms. We also explained the Shor's algorithm, which would break the different classical crypt algorithm, currently in use in travel documents. We did our research to find the right candidates for the lattice based algorithms. We also discussed the primitives behind these algorithms and the problems, these algorithms face. We also focused on the working of these algorithms. We also described the key exchange mechanism based on lattice based cryptography. The important candidates among all were one time digital signature based on Small integer solution problem, LWE based symmetric scheme and LWE based asymmetric scheme, original and a new GGH scheme, NTRU, Ajtai-Dwork cryptosystem, Stehle Stienfeld Tanaka ideal lattice based cryptosystem and Regev cryptosystem.

In the third chapter, we discussed the operations of the travel documents, need for the travel documents, and the security goals of the travel documents. We also derived all the protocols of the travel documents and showed how each of them could be broken if a quantum attack was performed on these protocols. We also showed the constraints of the travel documents with the memory and time constraints of the protocol and travel documents. The core goal was to ensure that our proposed crypto algorithm, key exchange scheme and digital signature must still hold these constraints (memory, time constraints true) we also discussed the public key infrastructure and the constraints of the same.

In the fourth chapter, we evaluated all the candidates discussed in the second chapter with the constraints derived in the third chapter. We also focused on how each of the candidates would be implemented in the scenario of travel documents. We also showed all the components of the key exchange scheme. We also gave our proposals as an addition to the list of candidates for optimizing these candidates with respect to functions and components that would reduce their

public key size, private key size and improve throughput for encryption and decryption. We also discovered a key exchange algorithm, which we discussed in the same chapter derived from the LEW-injection scheme but the key exchange mechanism provides a better efficiency with respect to public key and private key size. We described the Key exchange scheme recently discovered Peikert, which uses components derived from many schemes previously known. We showed how it provides efficiency and how it can be used in the travel documents. We also shortlisted the cryptosystems based on the parameters discussed in the same chapter.

In the Final chapter, we zoomed in our evaluation process into four important schemes namely LWE-asymmetric scheme, NTRU, Ideal lattice cryptosystem and proposed GGH scheme with many alterations to the original scheme. We showed that the new GGH and LWE are efficient in encryption and decryption but the public key sizes are quite large for implementation in travel documents. We gave our proposal of using a Digital signature based on Small Integer solution problem to communicate the large keys. This can be done by using 2 -1024 bits of public keys, which can be supported by the current public key infrastructure of the travel documents. We also showed how the Ideal lattice cryptosystem meets all the constraints of the travel documents with NTRU being slow in encryption and decryption process. We then through our research in travel documents and these algorithms showed how the LWE-symmetric scheme, efficient Key exchange algorithm and asymmetric schemes can be implemented in the travel documents. We also drew the outline for each of the protocols used in the travel documents.

There are still some important works to be done, which are serious changes in the public key infrastructure of the travel documents. Our proposal and research ensures that efficiency and memory constraints are met but it needs two public keys and two private keys for our proposed protocols to meet. We need to understand how it can be reduced to one so the current PKI itself can be used. After that, the implementation of the protocols in the travel documents would be the major challenge

## Index of Figures

|                                                                                    |    |
|------------------------------------------------------------------------------------|----|
| 1 Probabilistic SVP oracles in lower dimension .....                               | 13 |
| 2 Comparision of effeciency of algorithms.....                                     | 13 |
| 3 A lattice in $R^2$ and two of its bases .....                                    | 21 |
| 4 Shortest vector problem .....                                                    | 23 |
| 5 Closest vector problem .....                                                     | 24 |
| 6 Relationship between lattice problems .....                                      | 24 |
| 7 Comparision of GGH algorithms .....                                              | 36 |
| 8 Tabular details of Algorithms .....                                              | 36 |
| 9 Efficiency comparision between Babai round off and nearest plane algorithm ..... | 59 |
| 10 Comparision between GGH-CRT and normal GGH .....                                | 59 |
| 11 Comparision between GGH, Micciancio and new scheme's encryption scheme.....     | 61 |
| 12 Comparision between GGH, Micciancio and new scheme's decryption scheme .....    | 62 |
| 13 Comparision between GGH, Micciancio and new scheme's private key size .....     | 62 |
| 14 Comparision between GGH, Micciancio and new scheme's public key size .....      | 63 |
| 15 KEM .....                                                                       | 67 |
| 16 Complete key exchange mechanism .....                                           | 68 |
| 17 Tabular details of the algorithms .....                                         | 70 |

## Index of Tables

|                                      |    |
|--------------------------------------|----|
| 1 Tabular details of candidates..... | 73 |
|--------------------------------------|----|

# Glossary and Abbreviation

---

|                          |                                                                                                                                                                                              |
|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| GCD                      | Greatest Common Divisor                                                                                                                                                                      |
| Qubits                   | Quantum bits; Bits used in quantum computer allowing non-binary state unlike classical computers                                                                                             |
| DH                       | Diffie Hellman algorithm; Classical key exchange algorithm                                                                                                                                   |
| Triple DES               | Symmetric encryption algorithm used in classical computer                                                                                                                                    |
| NP-Complete              | Group of exponential time needed to break algorithms; when one of the algos is solved, all others would be solved                                                                            |
| SVP                      | Shortest vector problem; obtaining the shortest non zero vector in it                                                                                                                        |
| CVP                      | closest vector problem; the goal is to find lattice point closest to t with in the radius r.                                                                                                 |
| SIS                      | Short integer solution; The concept used in building digital signature due to the properties of homomorphism and compression                                                                 |
| Public key               | Public key used in asymmetric scheme allows to encrypt the data using reciever's and verify the signature using sender's public key. Public key is known to all and is distributed by th PKI |
| PKI                      | Public key infrastructure; It issues certificates allowing the sender and reciever to obtain the public key                                                                                  |
| Private key              | Private key allows decrypting the data and signing the signature. Private key is not known to anyone else                                                                                    |
| Digital signature scheme | Digital signature allows authentication, non-repudiation and integrity by signing data by its private key                                                                                    |
| Z                        | Integer                                                                                                                                                                                      |

|                      |                                                                                                                                               |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| homomorphic property | Addition and multiplication on encrypted data will equally modify the data on same operations                                                 |
| MAC                  | Mac offers authentication and integrity by using a common key over data                                                                       |
| CSCA                 | Country Signing Certificate Authorities                                                                                                       |
| DS                   | Document Signer                                                                                                                               |
| CVCA                 | Country Verifying Certificate Authority                                                                                                       |
| DV                   | Document Verifier                                                                                                                             |
| ICAO                 | International civil aviation organization                                                                                                     |
| RFID                 | Radio frequency ID (chip in the epassport)                                                                                                    |
| EF                   | elementary file stores the security object descriptors and DG                                                                                 |
| DG                   | data groups for storing the data of the passport                                                                                              |
| SOD                  | Security Object descriptors is the hashing of DG individually but collectively signed by the issuing state                                    |
| DF                   | The dedicated file (DF) in chip file system stores the encryption, MAC and private key of the e-passport bearer used in active authentication |
| KP <sub>uAA</sub>    | Public key stored in DG15 for active authentication                                                                                           |

# References

---

[AM '05] Ajtai, M.: Representing hard lattices with  $O(n \log n)$  bits. In Proc. 37th Annual ACM Symp. on Theory of Computing (STOC) (2005).

[AM '96] Ajtai, M.: Generating hard instances of lattice problems. In: Proc. 28th ACM Symp. on Theory of Computing. (1996) 99–108 Available from ECCC at <http://www.uni-trier.de/eccc/>.

[AM '98] [8] Ajtai, M.: The shortest vector problem in  $\mathbb{Z}^2$  is NP-hard for randomized reductions (extended abstract) 10-19. In Proc. 30th ACM Symp. on Theory of Computing (STOC), pages 10–19, ACM (1998).

[AP'09] J. Alwen and C. Peikert,: Generating shorter bases for hard random lattices. In Proceedings of STACS 2009, LNCS. Springer, (2009)

[BBBV '97][2] Bennett C.H., Bernstein E., Brassard G., Vazirani U.: The strengths and weaknesses of quantum computation. SIAM Journal on Computing 26(5), page 1510–1523 (1997)

[BCDP'96] Beckman D., Chari A. P., Devabhaktuni S., Preskill J.: Efficient Networks for Quantum Factoring. Cornell Symposium, pages 1-56 (1996)

[BFK09] Jens Bender, Marc Fischlin, and Dennis Kugler. Security Analysis of the PACE Key- " Agreement Protocol. In Information Security, volume 5735 of Lecture Notes in Computer Science, pages 33–48. Springer, 2009.

[BFK13] Jens Bender, Marc Fischlin, and Dennis Kugler. The PACE—CA Protocol for Machine " Readable Travel Documents. In Trusted Systems, volume 8292 of Lecture Notes in Computer Science, pages 17–35. Springer, 2013.

[Blundo et al] Blundo,: Attacking Cryptographic Schemes Based on "Perturbation Polynomials", US Army Research Laboratory and the UK Ministry of Defence under agreement number W911NF-06-3-0001, 2011

[Bogari, Zavarisky, Lindskog, Ruhl, 2005] Bogari, Zavarisky, Lindskog, Ruhl: An Analysis of Security Weaknesses in the Evolution of RFID Enabled Passport, ISBN: 978-1-908320-04/9/2012 IEEE (2005)

[BOM] Alan Genz,: Methods for Generating Random Orthogonal Matrices, Proceedings of the MCQMC98 Meeting., 1998

[BPBP13] Nicolas Buchmann, Roel Peeters, Harald Baier, and Andreas Pashalidis. Security Considerations on Extending PACE to a Biometric-Based Connection Establishment. In

Proceedings of the Special Interest Group on Biometrics and Electronic Signatures, Lecture Notes in Informatics (LNI), pages 15–26, Darmstadt, DE, 2013. Bonner K’ollen Verlag.

[BSI10] KLUGLER, D., “Advance security mechanisms for machine readable travel documents, Technical report”, Federal Office for Information Security (BSI), Germany, 2005.

[BTA’09] Barral and A. tria, ‘Fake finger prints in fingerprint recognition’, Springer,( 2009)

[CK01] R. Canetti and H. Krawczyk,: Analysis of key-exchange protocols and their use for building secure channels. In EUROCRYPT, pages 453–474. 2001. Full version at <http://eprint.iacr.org/2001/040>.

[CK02a] R. Canetti and H. Krawczyk,: Security analysis of IKE’s signature-based key-exchange protocol. In CRYPTO, pages 143–161. 2002. Full version at <http://eprint.iacr.org/2002/120>

[coppersmith winograd algorithm] Virginia Vassilevska Williams,: Multiplying matrices in  $O(n^{2.373})$  time, Stanford university, July, 2014

[DJ’04] Daugman,J.: ‘How IRIS recognition works’ . IEEE,( 2004)

[FA’11] Frontex Agency; Operational and Technical security of Electronic Passports Warsaw, July 2011

[FCR’06] FAKHRY H. and CARDOZO Benedict; RESEARCH AND DEVELOPMENT OF AN IRIS BASED RECOGNITION SYSTEM FOR IDENTIFICATION AND SECURE AUTHENTICATION, An International Journal, Vol.19, 2006, 39-57, 2006

[FO99a,b] E. Fujisaki and T. Okamoto,: Secure integration of asymmetric and symmetric encryption schemes. In CRYPTO, pages 537–554. 1999.

[GMSS’99] Goldreich, Micciancio, Safra, Seifert,: Approximating shortest lattice vectors is not harder than approximating closest lattice vectors (Inf. Proc. Letters, 1999)

[GN ‘08] Gama, N. and Nguyen, P.Q.: Finding short lattice vectors within Mordell’s inequality. In Proc. 40th ACM Symp. on Theory of Computing (STOC), pages 207–216 (2008).

[GO’01] O. Goldreich. Foundations of Cryptography., Volume 2- basic applications. Cambridge University press, 2001

[GPV’08][22] Gentry C., Peikert C., and Vaikuntanathan V.: Trapdoors for hard lattices and new cryptographic constructions. In Proceedings of STOC, pages 197206 (2008)

[HEG’13][30] C Hasher, G. Erlebcher, ‘A novel technique for face recognition using range images’, seventh international symposium, (2003)

[HHPS ‘07] Hoffstein, J., Howgrave-Graham, N., Pipher, J., and Silverman, J.H.: Hybrid lattice reduction and meet in the middle resistant parameter selection for NTRUEncrypt.

Submission/contribution to ieeep1363.1, NTRU Cryptosystems, Inc., URL <http://grouper.ieee.org/groups/1363/lattPK/submissions.html#2007-02> (2007).

[ICAO6] ICAO, : Machine readable travel documents, Technical report, ICAO 2006

[ICAO13] ICAO,: Machine Readable Travel Documents, Part 1 Machine Readable Passports. ICAO, Fifth Edition, 2013

[Juels et al] D. Monar, A. Juels, and D. Wagner., Security and privacy issues in e-passports, Cryptology, ePrint Archive, Report 2005/095, 2005.

[KC and Karger, 2005] Gaurav S. Kc and Paul A. Karger. Security and privacy issues in machine readable travel documents (MRTDs). IBM Technical Report (RC 23575), IBM T. J. Watson Research Labs, April 2005.

[Kra03] H. Krawczyk. SIGMA: The ‘SIGn-and-MAC’ approach to authenticated Diffie-Hellman and its use in the IKE-protocols. In CRYPTO, pages 400–425. 2003.

[Kra96] H. Krawczyk. SKEME: a versatile secure key exchange mechanism for Internet. In NDSS, pages 114–127. 1996.

[KS’13] [Kumar, Shrinivasan, 2013] V.K. Narendira Kumar & B. Srinivasan,: Evolution of Electronic Passport Scheme using Cryptographic Protocol along with Biometrics Authentication System, (2013)

[LC ‘03] Ludwig, C.: A faster lattice reduction method using quantum search. In ISAAC, pages 199–208 (2003).

[LLS’91] Lagarias, Lenstra, Schnorr.: Some new results in simultaneous Diophantine approximation , (Combinatorica 1990)

[LM08] V. Lyubashevsky and D. Micciancio. Asymptotically efficient lattice-based digital signatures. In TCC, pages 37–54. 2008.

[LPR10] V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. Journal of the ACM, 60(6):43:1–43:35, November 2013. Preliminary version in EUROCRYPT ‘10.

[L’87] L Lovasz,: Normal hypergraphs and the perfect graph conjecture, (SIAM 1986)

[MC ‘08] [4] Meyer, Carl D.: Matrix Analysis and Applied Linear Algebra, SIAM, ISBN 978-0-89871454-8 (2008)

[MG’01] Micciancio, Goldwasser,: The shortest vector problem is NP-hard to approximate to within some constant, (Springer 2001)

[MJW’05] D. Monar, A. Juels, D. wagner: Security and privacy issues report,( 2005/095)

[MR ‘08] D. Micciancio and O. Regev.: Post-quantum Cryprography, chapter Lattice-based Cryptography. pages 147–191. Springer, 1 edition, (2008)

[M'08] M Micciancio,:The hardness of the closest vector problem with preprocessing (SODA 2008) (FOCS 2002/Comp. Compl. 2007)

[NBS'12] V.K. Narendra Kumar, B. Shrinivasan :Evolution of electronic passport scheme using cryptographic protocol along with biometric authentication system, Computer Network and information security, (2012)

[NR'09] Rishab Nithyanand; A Survey on the Evolution of Cryptographic Protocols in ePassports, University of California

[NV '11] Nguyen P.Q. and Vidick Thomas: Sieve Algorithms for the Shortest Vector Problem are Practical

[PC'09] C. Peikert.: Public-key cryptosystems from the worst-case shortest vector problem. In Proceedings of STOC 2009, pages 333-342. ACM, (2009)

[PC '14] Peikert C.: Lattice Cryptography for the Internet. School of Computer Science, College of Computing, Georgia Institute of Technology, (July 2014)

[PHF13] Roel Peeters, Jens Hermans, and Junfeng Fan. IBIHOP: Proper Privacy Preserving Mutual RFID Authentication. In Workshop on RFID and IoT Security - RFIDSec Asia 2013, Cryptology and Information Security, pages 45–56, Guangzhou, China, 2013.

[PRS'09] [36] T. Plantard, M. Rose, and W. Susilo.: Improvement of lattice-based cryptography using CRT. In A. Sergienko, S. Pascazio, and P. Villoresi, editors, Quantum

[PVB '08] C. Peikert, V. Vaikuntanathan, and B. Waters: A framework for efficient and composable oblivious transfer. In Proceedings of Crypto 2008, volume 5157 of LNCS, pages 554-571. Springer,(2008)

[RKBT10] J. Randall, B. Kaliski, J. Brainard, and S. Turner. Use of the RSA-KEM Key Transport Algorithm in the Cryptographic Message Syntax (CMS). RFC 5990 (Proposed Standard), September 2010

[RM '11] Rose M.: lattice based cryptography- a practical implementation' (2011)

[RO'00] Regev O.: On lattices, learning with errors, random linear codes and cryptography

[RO '05] Regev O.: On lattices, learning with errors, random linear codes, and cryptography. In Proceedings of STOC 2005, pages 849-3. ACM, (2005)

[RO '14] Regev O.: Lattice-based Cryptography. CSE Department, University of California, San Diego, La Jolla, CA 92093, US (2014)

[Sarma, Weis,Engels, 2002] Sanjay E. Sarma, Stephen A. Weis, Daniel W. Engels,: RFID Systems, Security & Privacy Implications, MIT-AUTOID-WH-014, 2002

[SA '13] V. Shoup. NTL: Number theory library and The GNU multiple precision arithmetic library, (2013)

[SSTX'00] Stehle, Stienfeld, Tanaka and Xagawa.: An efficient public key encryption based on Ideal Lattices.

[VP '11] Voulgaris P.: Algorithms for the closest and shortest vector problems on general lattices. UC San Diego Electronic Theses and Dissertations, (2011)

[ZA'11][3] Zornitsa B.: Rankin Lattice Reduction for Programmers, University of Technology Darmstadt (2011)

(BR'94, Shoup'01) [BR94] M. Bellare and P. Rogaway. Optimal asymmetric encryption. In EUROCRYPT, pages 92–111. 1994.

(BR'94, Shoup'01) [Sho01] V. Shoup. OAEP reconsidered. J. Cryptology, 15(4):223–249, 2002. Preliminary version in CRYPTO 2001.

(BR'97, CS'98,ABR'01) [ABR01] M. Abdalla, M. Bellare, and P. Rogaway. The oracle Diffie-Hellman assumptions and an analysis of DHIES. In CT-RSA, pages 143–158. 2001.

(BR'97, CS'98,ABR'01) [BR97] M. Bellare and P. Rogaway. Minimizing the use of random oracles in authenticated encryption schemes. In ICICS, pages 1–16. 1997.

(BR'97, CS'98,ABR'01) [CS98] R. Cramer and V. Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. SIAM J. Comput., 33(1):167226, 2003. Preliminary version in CRYPTO 1998.

(Johnson, 2006; Knight, 2006; Reid, 2006) JOHNSON, B. (2006): Hackers crack new biometric passports, The Guardian

(Johnson, 2006; Knight, 2006; Reid, 2006) KNIGHT, W. (2006): Hackers clone radio-chip passports, NewScientist

(Johnson, 2006; Knight, 2006; Reid, 2006) REID, D. (2006): E-passports at risk from cloning, BBC.

(Kalach, Quisqater, 2007) Gildas Avoine, Kassem Kalach, and Jean-Jacques Quisquater. ePassport: Securing International Contacts with Contactless Chips. In Financial Cryptography and Data Security, volume 5143 of Lecture Notes in Computer Science, pages 141–155. Springer, 2008.

(OP'01) [OP01] T. Okamoto and D. Pointcheval. REACT: Rapid enhanced-security asymmetric cryptosystem transform. In CT-RSA, pages 159–175. 2001.

(Peeters, Hermans, Mennink; 2013) Peeters R., Hermans J., Menmink: Speedup for European ePassport Authentication

(Regev' 05) Oded Regev, “On lattices, learning with errors, random linear codes, and cryptography”, 2005

(RFC 5990) Randall J., Kaliski B., Brainard J., Turner S., “ Use of the RSA-KEM Key transport Algorithm in the Cryptographic Message Syntax”

(STOC’08) 40th ACM International Symposium on theory of computing, Association for Computing machinery, 2008

(Tom A. F. Kinneging for ICAONTWG, 2004) TOM A. F. KINNEGING for ICAO-NTWG, P. T. F. (2004): PKI for machine readable travel documents offering ICC readonly access, Technical report. Version 1.1.

Home affairs justice,: EU standard specifications for security features and biometrics in passports and travel documents, Technical report, European Union, 2006.

A. K. Lenstra, H. W. Lenstra, and L. Lovasz. Factoring polynomials with rational coefficients. *Mathematische Annalen*, Springer-Verlag, 1982

A. Kawachi, K. Tanaka, and K. Xagawa. Multi-bit cryptosystems based on lattice problems. In *Public Key Cryptography*, 2007.

Ajtai, M., Kumar, R., and Sivakumar, D.: A sieve algorithm for the shortest lattice vector problem. In *Proc. 33rd ACM Symp. on Theory of Computing*, pages 601–610 (2001).

Ajtai, M.: Generating hard instances of lattice problems. In *Complexity of computations and proofs*, volume 13 of *Quad. Mat.*, pages 1–32. Dept. Math., Seconda Univ. Napoli, Caserta (2004).

Babai, L.: On Lovász lattice reduction and the nearest lattice point problem. *Combinatorica*, 6:1–13 (1986).

C. H. Bennett and G. Brassard. Quantum cryptography: Public-key distribution and coin tossing. In *Proceedings of IEEE International Conference on Computers, Systems and Signal Processing*, Bangalore, India,. IEEE Press,1984.

Communication and Quantum Networking - QUANTUMCOMM 2009, volume 36,, 2009

D. Micciancio. Improving lattice based cryptosystems using the Hermite normal form. In *Cryptography and Lattices*, 2001

D. Stehle and R. Steinfeld. Making NTRU as secure as worst-case problem over ideal lattice. In *Proceedings of EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*. Springer, 2011.

David Galindo, Sebastia Martín, Paz Morillo and Jorge L. Villar,: Fujisaki-Okamoto IND-CCA hybrid encryption revisited, 1999

Gentry C.: Fully Homomorphic Encryption Using Ideal Lattices. In the 41st ACM Symposium on Theory of Computing (STOC), (2009).

In D. J. Bernstein, J. Buchmann, and E. Dahmen, editors, Post-quantum Cryptography. Springer, 2008.

J.-Y. Cai and T. W. Cusick. A lattice-based public-key cryptosystem. In Selected Areas in Cryptography, 1998.

J. Hofstein, J. Pipher, and J. H. Silverman. Ntru: A ring-based public key cryptosystem. In Algorithmic Number Theory (ANTS 1998), Lecture Notes in Computer Science 1423, Springer-Verlag, 1998

Kannan, R.: Improved algorithms for integer programming and related lattice problems. In Proc. 15th ACM Symp. on Theory of Computing (STOC), pages 193–206. ACM (1983).

L. Babai. On Lovasz' lattice reduction and the nearest lattice point problem. Combinatorica, 1986.

Lagarias, J.C., Lenstra, Jr., H.W., and Schnorr, C.P.: Korkin-Zolotarev bases and successive minima of a lattice and its reciprocal lattice. Combinatorica, (2008).

Lyubashevsky, V., Regev p.: On Ideal Lattices and Learning with Errors Over Rings. IACR. pages 333-348(2011).

M. Ajtai and C. Dwork. A public-key cryptosystem with worst-case/averagecase equivalence. In Twenty-Ninth Annual ACM Symposium on the Theory of Computing (STOC 1997), pages 284, 1997

M. Rose, T. Plantard, and W. Susilo. Improving BDD cryptosystems in general lattices. In to appear in Proceedings of ISPEC 2011, 2011.

O. Goldreich, S. Goldwasser, and S. Halevi. Eliminating decryption errors in the ajtai-dwork cryptosystem. In CRYPTO, 1997

Pashupathinathan V., Pieprzyk J.: Wang H., Security Analysis of Australian and E.U. E. passport Implementation. Journal of Research and Practice in Information Technology, vol 40. No.3, August 2008

R. Fischlin and J.-P. Seifert. Tensor-based trapdoors for cvp and their application to public key cryptography. In IMA Int. Conf., 1999

Regev, O.: Quantum computation and lattice problems. SIAM J. on Computing, 33(3),pages 738–760 (2004).

Ruckert M.: Estimating the Security of Lattice-based Cryptosystems. TU Darmstadt, CASED (2008)

Yuriy Arbitman, Gil Dogon, Vadim Lyubashevsky, Daniele Micciancio, Chris Peikert, and Alon Rosen. SWIFFTX: a proposal for the SHA-3 standard, 2008.