MOR CRYPTOSYSTEM WITH ORTHOGONAL GROUPS

A thesis

submitted in partial fulfillment of the requirements

of the degree of

Doctor of Philosophy

by

Pralhad Mohan Shinde

Roll Number: 20123168

INDIAN INSTITUTE OF SCIENCE EDUCATION AND RESEARCH PUNE

April, 2017

Dedicated to

My beloved Parents

Certificate

Certified that the work incorporated in the thesis entitled "*MOR cryptosystem with orthogonal groups*", submitted by *Pralhad Mohan Shinde* was carried out by the candidate, under our supervision. The work presented here or any part of it has not been included in any other thesis submitted previously for the award of any degree or diploma from any other university or institution.

Date: April 13, 2017 *Dr. Ayan Mahalanobis and Dr. Anupam Singh*

Thesis Supervisors

Declaration

I declare that this written submission represents my ideas in my own words and where others ideas have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that violation of the above will be cause for disciplinary action by the institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Date: April 13, 2017 **Pralhad Mohan Shinde**

Roll Number: 20123168

Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisors Dr. Ayan Mahalanobis and Dr. Anupam Singh for their excellent guidance, constant encouragement, patience and care during the entire course of Ph.D. Their advice helped me in all the time of research and writing of this thesis. I must thank them for going through the thesis for several times and suggesting many corrections. I could not have imagined having better advisors and mentors for my Ph.D. study.

I would like to thank Dr. Manoj Kumar Yadav (HRI Allahabad), the member of my Research Advisory Committee, for his valuable comments and useful suggestions regarding my research. I would like to express my sincere thanks to Dr. Anisa Chorwadwala who was the mentor for my minor thesis. I would also like to thank all the faculty members of the department of mathematics at IISER Pune for their assistance and support.

I gratefully acknowledge financial support from CSIR for the past five years that made my Ph.D. work possible. I am also thankful to the staff at the administration, IT and library of the institute. Special thanks are due to Mrs. Suvarna Bharadwaj and Mr. Tushar Kurulkar who always provided their generous help in technical paperwork and procedural requirements.
It is a pleasure to thank my friends and colleagues at IISER Pune, for the wonderful times we shared at the campus. In addition, special thanks to my batchmates Prabhat, Sushil, Manidipa for all your support these past five

years to get me here.

Last, but not least, I would like to dedicate this thesis to my family, my mother Janaki Shinde, my father Mohan Shinde and brother Bholenath and his wife Rohini, for their love, patience, and understanding- they allowed me to spend most of the time on this thesis.

Pralhad Mohan Shinde

Abstract

In 2001, Paeng, Ha, Kim, Chee, and Park presented a new public key encryption scheme based on the difficulty of the discrete logarithm problem in an inner automorphism group of a non-abelian group G called MOR cryptosystem. It is an elementary generalization of the Classical ElGamal cryptosystem. In this case, the discrete logarithm problem works in an automorphism group of the group G, rather than G itself.

Automorphisms used for MOR cryptosystem are presented as an action on generators of G, so in a practical implementation of the MOR cryptosystem, there are two things that matter the most.

- Any automorphism of the group can be determined by its action on the group's generators. Hence, the size of public-key depends on the number of generators.

- If one has to compute $\phi(g)$, then the word problem must be solved. Thus, we need an efficient algorithm to solve the word problem in G.

Hence the natural question arises: what are the right groups for the MOR cryptosystem?

Since its inception, several people have tried to provide different candidates for G and have analyzed the security of corresponding MOR cryptosystem. Paeng, et al. proposed the MOR cryptosystem using group $\mathrm{SL}(2, \mathbb{Z}_p) \rtimes \mathbb{Z}_p$ and later Tobias showed that MOR using $\mathrm{SL}(2, \mathbb{Z}_p) \rtimes \mathbb{Z}_p$

is not harder than MOR using $\mathrm{SL}(2, \mathbb{Z}_p)$. Mahalanobis [27] used the group of unitriangular matrices for the MOR cryptosystem and showed that the MOR cryptosystem with the group of unitriangular matrices over $\mathbb{F}_q$ is as secure as the ElGamal cryptosystem over the finite field $\mathbb{F}_q$. Further, Mahalanobis [28] studied the MOR cryptosystem over $\mathrm{SL}(d, q)$ and showed that if one assumes that the only way to break the proposed MOR cryptosystem is to solve the discrete logarithm problem in the automorphism group, then MOR cryptosystem over $\mathrm{SL}(d, q)$ is as secure as the ElGamal cryptosystem over $\mathbb{F}_{q^d}$. This d is called the security advantage, or the **embedding degree**.

In this thesis, we have studied the MOR cryptosystem with finite orthogonal groups and analyzed its security. We have addressed both of the above-mentioned questions about the key size and word problem. We present an algorithm similar to Gaussian elimination algorithm for twisted orthogonal group $\mathrm{O}^-(d, K)$ and orthogonal groups over a field of characteristics 2 to solve the world problem. Our algorithms works in any orthogonal group $\mathrm{O}(d, K)$ for particular non-degenerate symmetric bilinear form and any field K. Further, we have shown that security advantage (embedding degree) of the MOR cryptosystem using $\mathrm{O}(d, q)$ with q even and d odd is $d - 1$ and we have conjectured that the security advantage of the MOR cryptosystem using other orthogonal groups is $d^2 - 1$. Further, as an application of algorithm developed to solve the word problem we computes the spinor norm of elements of $\mathrm{O}^-(d, K)$.

Contents

List of Tables

Chapter 1

Introduction

Cryptography has a significant impact on our day-to-day life. Every time one makes a mobile phone call, send an email, buy something with a credit card in a shop or on the web, or even get cash from an ATM, cryptography ensures the confidentiality of the transaction and provides the security to make it possible. There are two types of cryptography: symmetric key cryptography and public-key cryptography. Symmetric key cryptosystems provide the perfect security, but there is a problem that receiver and sender need to agree on the shared secret key ahead of time. This requires either both of them to meet in person or agree on the key through an insecure channel. However, none of these is acceptable if one wants to encrypt his credit card information for online shopping from a distant vendor. This problem was solved in 1976 by Diffie-Hellman [15] using public-key cryptography. The security of most of the public-key cryptosystems that are currently used in practice is based, either directly or indirectly, on either the difficulty of factoring large numbers or the difficulty of finding discrete logarithms in a finite group.

One-way functions form the basis of public-key cryptography. Although we have computationally hard problems that are believed to be one-way, none has been proven to be so. Therefore the security of the corresponding cryp-

tographic schemes depends on the intractability assumptions of these problems. Two major species of such problems, factoring and discrete logarithm, are widely believed to be intractable and serve as the basis of many popular schemes. However, once a general purpose quantum computer is built, then people will be able to run Shor's algorithm [52] and solve factoring, DLP in any finite field $\mathbb{F}_{p^s}$ in polynomial time. This is the most worrisome long-term threat to current public-key cryptosystems. So the biggest challenge for cryptography community is to build more secure and alternative cryptographic primitives.

At present, RSA and discrete logarithm problem over elliptic curves [24, 38] are believed to be more secure. One can raise the question: Why do we care about elliptic curves if RSA works well? The obvious answer is the key size because ECC with a key size of 160 bits can achieve the same level of security as RSA with a key size 1024 bits. This shows that not only ECC uses less memory, but also that key generation and signing (signature scheme) are considerably faster. However, in light of MOV attack [34], one needs to be more careful about the choices of elliptic curves in practice. The MOV attack uses a bilinear pairing, which is a bilinear map from $E(\mathbb{F}_q) \times E(\mathbb{F}_q)$ to the finite field $\mathbb{F}_{q^k}$, where k is the embedding degree associated with the curve. The bilinearity means that $e(rP, sQ) = e(P, Q)^{rs}$ for points P, Q. By computing $u = e(P, Q)$ and $v = e(rP, Q)$ for any Q and using the bilinearity, we have $v = e(P, Q)^r = u^r$. Now one can solve the discrete logarithm in $\mathbb{F}_{q^k}$ in order to solve the discrete logarithm in the elliptic curve. So in order to avoid the MOV attack, one needs to choose those elliptic curves whose embedding degree is high. Though the security in elliptic curves is considered much better than that of finite fields because of the non-existence of sub-exponential algorithms in most cases of elliptic curves [5, 54]; However, there are only a few practically used elliptic curves recommended by NIST and

the recent article issued by NSA [1] gives emphasis on to find alternative cryptosystems which are more secure.

Cryptographers have begun to pay more attention towards non-Abelian cryptography based on non-Abelian structures, and several attempts were made in that direction. To name a few, Maze et.al [31] proposed a semigroup action problem (SAP), Shilrain and Zapata developed CAKE [53], both works with non-abelian structures. There is an interesting cryptosystem in the work of Climent et.al [12].

Another line of research in cryptography is to generalize the well-established cryptosystems with the hope that something practical and useful will come out of the generalization. Keeping this fact in mind, we have studied MOR cryptosystem which is a natural generalization of ElGamal cryptosystem which uses discrete logarithm as the cryptographic primitive. It is well known that the security of such cryptosystem depends on the difficulty of finding discrete logarithms in a platform group. In this case, the discrete logarithm problem works in the automorphism group of a group G, rather than G itself. This shows that one can use almost any group for the MOR cryptosystem. Thus, the natural question arises what are groups for which we get a secure MOR cryptosystem? In light of Joux's attack on the discrete logarithm problem in finite fields of small characteristics [6, 20], one must look for a non-Abelian group(s) G in the hope that discrete logarithm problem in G is hard. Several attempts were made to propose MOR cryptosystem with different non-Abelian groups [27–29, 45]. In this thesis, we propose a MOR cryptosystem using orthogonal groups over finite fields and analyze its security.

1.1 Our contribution and overview

This thesis is organized as follows: In chapter 2, we give a description of MOR cryptosystem and discuss the motivation for our work. In chapter 3, we briefly describe the theory of Chevalley groups which will be used in later chapters.

Chapters 4, 5 and Appendix A are our contributions to this thesis. In chapter 4, we develop algorithms to solve the word problem in twisted orthogonal groups and orthogonal groups over field of even characteristics. Also, we describe the algorithm developed in [8] for the split orthogonal groups over finite fields of odd characteristics. At the end of this chapter we discuss some applications of our algorithms.

In chapter 5, we have analyzed the security of MOR cryptosystem with orthogonal groups and addressed the issues with the public-key size. We also discuss how to choose right automorphisms for the MOR cryptosystem with orthogonal groups and showed that the security of MOR cryptosystem with $O(d, q)$, where d-odd and q-even is same as security of ElGamal cryptosystem over $\mathbb{F}_{q^d}$. Further, we have given enough evidence and conjectured that security of the MOR cryptosystem with other orthogonal groups is same as the security of ElGamal cryptosystem over $\mathbb{F}_{q^{d^2-1}}$. At the end of this chapter, we discuss the implementation of MOR cryptosystem with orthogonal groups.

In appendix A, as an application to our algorithms developed in chapter 4, we compute the spinor norm of elements in twisted orthogonal groups, which is of great importance in computational group theory.

Chapter 2

MOR cryptosystem

This chapter contains some basic definitions and known results that serve as a prerequisite material for this thesis and motivates to our work. In Section 2.1, we describe the discrete logarithm problem which is a basis for the MOR cryptosystem. In Section 2.2, we describe the Diffie-Hellman problem which is used as a security measure in ElGamal cryptosystem [Section 2.3]. In Section 2.4, we describe MOR cryptosystem and give some examples of MOR cryptosystems.

2.1 Discrete logarithm problem

The discrete logarithm problem is a frequently used cryptographic primitive which works in any cyclic group but need not be secure over it. If G is a multiplicative cyclic group and g is a generator of G, then from the definition of cyclic groups, we know every element h in G can be written as g^x for some integer x. The discrete logarithm of h to the base g in the group G is defined to be x. For example, if the group is $\mathbb{Z}_{11}^{\times}$, and the generator is 2, then the discrete logarithm of 9 to the base 2 is 6 because $2^6 \equiv 9 \bmod 11$.

Definition 2.1.1. The discrete logarithm problem is defined as: given

a cyclic group G, a generator g of the group and an element g^x of G, the problem is to compute x.

Discrete logarithm problem is not always difficult to solve. The hardness of the discrete logarithm problem depends on the structure of the groups. For example, an obvious and popular choice of groups for discrete logarithm based cryptosystems is $\mathbb{Z}_p^\times$, where p is a prime number. However, one can use Chinese remainder theorem to reduce the DLP in smaller groups. If $p-1$ is a product of small primes, then the Pohlig-Hellman algorithm [47] can solve the discrete logarithm problem in this group very efficiently. This is the reason we always want p to be a large and safe prime when using $\mathbb{Z}_p^\times$ as an underlined group for discrete logarithm based crypto-systems. A safe prime is a prime number which equals $2q+1$, where q is a large prime number. This guarantees that $p-1=2q$ has a large prime factor so that the Pohlig-Hellman algorithm cannot solve the discrete logarithm problem easily. The only condition that p is a safe prime is not sufficient because there is a sub-exponential algorithm which is called the index calculus [51]. Thus, p must be very large (usually at least 1024-bit) to make the cryptosystems secure.

Consider another example, let N be a positive integer and consider the case when $G = \mathbb{Z}_N$ the additive group of integers modulo N. Here the generators of the group are precisely the $x \in G$ such that $\gcd(x, N) = 1$ and the equation $dx \equiv y \pmod{N}$ can be solved by finding the multiplicative inverse of x modulo N with the extended Euclidean algorithm. Thus for this group, the DLP can be solved in polynomial time $O(\log^2 N)$. Note that if G is a finite cyclic group of order N then G is isomorphic to $\mathbb{Z}_N$ in which the DLP is easy. It is interesting to note that though any two finite cyclic groups of the same order are isomorphic, computing the image of an element g^x in G under this isomorphism entails solving the DLP. Thus, it is not the structure

of a group, but its representation, which can make its DLP difficult.

2.2 Diffie-Hellman problem

A more important cryptographic primitive, related to the discrete logarithm problem is the **Diffie-Hellman problem**, which states that if g, g^{x_1} and g^{x_2} are given then find $g^{x_1 x_2}$. It's easy to see that if one solves discrete logarithm problem, then Diffie-Hellman problem can be solved, the other way is not known to be true for general groups.

The very important and very first cryptosystem based on the discrete logarithm problem is the ElGamal cryptosystem. It works in any cyclic subgroup of a group G, however, it need not be secure. Here we describe the ElGamal cryptosystem. Suppose Alice wishes to send a message m to Bob over an insecure channel. We assume that Alice's message, m, is encoded as an element in the cyclic group $G =< g >$.

2.3 Description of the ElGamal cryptosystem

Let the integer x be Bob's private key and g and $h = g^x$ are of public knowledge.

- Alice generates a random integer $r \in [1, |G|]$ and computes $c_1 = g^r$, $c_2 = h^r m$.
- Alice sends the cipher text (c_1, c_2) to Bob.
- Bob can recover the message by computing $c_1^{-x} c_2$.

Note that the hardness of the ElGamal cryptosystem is equivalent to the Diffie-Hellman problem [19, Proposition 2.10]. Most commonly groups used for the ElGamal cryptosystem are: $\mathbb{Z}_p^\times$, $\mathbb{F}_{p^n}^\times$, the group of rational points on elliptic curves over a finite field, etc.

2.4 Description of MOR cryptosystem

The concept of MOR cryptosystem was first proposed by Paeng, Ha, Kim, Chee and Park [45]. It is a public key encryption scheme which is based on the difficulty of the discrete logarithm problem in the inner automorphism group of a non-abelian group G. The MOR cryptosystem is a natural generalization of the ElGamal cryptosystem. In this case, the discrete logarithm problem works in the automorphism group of a group G, rather than G itself. This shows that one can use almost any group for the MOR cryptosystem. The description of MOR cryptosystem is as follows: Let $G =< g_1, g_2, ..., g_k >$ be a finite group, and ϕ be a non-identity automorphism of G. Suppose Alice wishes to send a message $m \in G$ to Bob. Let the integer x be a Bob's private key and $\{\phi(g_i)\}_{i=1}^k$, $\{\phi^x(g_i)\}_{i=1}^k$ are public.

- Alice generates a random integer $r \in [1, |G|]$ and computes $c_1 = \phi^r$, $c_2 = \phi^{rx}(m)$.
- Alice sends the ciphertext (c_1, c_2) to Bob.
- Bob can recover the message by computing $c_1^{-x} \circ c_2$.

Alice knows the order of the automorphism ϕ; she can use the identity $\phi^{t-1} = \phi^{-1}$ whenever $\phi^t = 1$ to compute ϕ^{-xr}. Note that if one can solve the Diffie-Hellman problem in $\langle\phi\rangle$, he can break the MOR cryptosystem. This follows from the fact that ϕ^r and ϕ^x are public. If one can solve the Diffie-Hellman problem, one can compute ϕ^{rx} and get the plaintext. The following theorem proves the converse, thus the hardness to break MOR cryptosystem is equivalent to the Diffie-Hellman problem in the group $< \phi >$.

Theorem 2.4.1. [29, Theorem 3.1]. If there is an oracle that can decrypt arbitrary ciphertext for the MOR cryptosystem, one can solve the Diffie-Hellman problem in $\langle\phi\rangle$.

In a practical implementation of MOR cryptosystem, there are two things that matter the most.

- Note that the public key used in MOR cryptosystem is presented as an action of automorphism on generators of group G. So the size of public-key depends on the number of generators and, also depends on the size of their representations.

- Efficient algorithm to solve the word problem. This means, given $G =< g_1, g_2, ..., g_k >$ and $g \in G$, is there an efficient algorithm to write g as a word in $g_1, g_2, ..., g_k$? The reason of this importance is immediate - the automorphisms are presented as action on generators and if one has to compute $\phi(g)$, then the word problem must be solved.

The obvious question is: what are the right groups for the MOR cryptosystem? Since its inception several people have tried to provide different candidate for G and have analyzed the security of corresponding MOR cryptosystem. To get the flavor, one can use a d-dimensional vector space $(V, +)$ over $\mathbb{F}_q$ as an underlying group for MOR cryptosystem.

2.4.1 MOR cryptosystem using $(V, +)$

To build the MOR cryptosystem using $(V, +)$ we need to understand its automorphisms. Note that the automorphism group of V is $GL(d, q)$, hence the MOR cryptosystem on V is equivalent to the ElGamal cryptosystem over $GL(d, q)$. Therefore the security of MOR cryptosystem depends on the hardness of DLP in $GL(d, q)$. Now, we can ask ourself how hard is DLP in $GL(d, q)$? Menezes and Wu [35] studied this problem and proved that DLP in $GL(d, q)$ is no more difficult than the DLP in $\mathbb{F}_{q^d}$. Their approach is as follows: Let A and $B = A^x$ are the elements of $GL(d, q)$, we can find the value of x as follows: Factorize the characteristic polynomial of A into its

irreducible factors over $\mathbb{F}_q$ and go to the splitting field of each irreducible factor. Then the discrete logarithm problem in $GL(d, q)$ can be reduced to the discrete logarithm problem in the splitting fields $\mathbb{F}_{q^t}$, where $1 \leq t \leq d$. If the characteristic polynomial is irreducible, then the discrete logarithm problem in A reduces to the discrete logarithm problem in $\mathbb{F}_{q^d}$. Thus the DLP in $GL(d, q)$ is no harder than the DLP in $\mathbb{F}_{q^d}$. This d is called the security advantage, or the **embedding degree**.

Now, the natural question arises: can we do better in terms of embedding degree than the case of MOR with V? Paeng, et al. proposed the MOR cryptosystem using group $SL(2, \mathbb{Z}_p) \rtimes \mathbb{Z}_p$. Later, Tobias [60], Paeng [44] showed that MOR using $SL(2, \mathbb{Z}_p) \rtimes \mathbb{Z}_p$ is not harder than MOR using $SL(2, \mathbb{Z}_p)$ and as a consequence of Mahalanobis work [28] we can see that this has embedding degree 2. In an attempt to study the MOR cryptosystem using the finite p-groups Mahalanobis [29] used the p'-automorphisms (an automorphism ϕ of a p-group G is a p-automorphism if its order is power of p and p'-automorphism if its order is co-prime to p.), and showed that there are secure MOR cryptosystems on a p-group. However, they offer no advantage than working with matrices over the finite field $\mathbb{F}_p$. This fact motivated Mahalanobis to use the matrix groups for the study of MOR cryptosystem in the hope of achieving higher embedding degree. So, in that direction Mahalanobis [27] used the group of unitriangular matrices for the MOR cryptosystem and showed that the MOR cryptosystem over the group of unitriangular matrices over $\mathbb{F}_q$ is as secure as the ElGamal cryptosystem over the finite field $\mathbb{F}_q$. Further, Mahalanobis [28] studied the MOR cryptosystem over $SL(d, q)$ and showed that if one assumes that the only way to break the proposed MOR cryptosystem is to solve the discrete logarithm problem in the automorphism group, then MOR cryptosystem over $SL(d, q)$ is as secure as the ElGamal cryptosystem over $\mathbb{F}_{q^d}$. Thus, has the embedding degree equal to d. Here we describe the

approach used by Mahalanobis for the study of MOR cryptosystem using $SL(d, q)$.

2.4.2 MOR cryptosystem using SL(d, q)

To build a MOR cryptosystem using any group G one needs to understand the automorphism group of G. Let ϕ be an automorphism of $SL(d, q)$, it is well-known [11, 14, 56] that $\phi = c_\chi \iota \delta \gamma \theta$, where c_χ is a central automorphism, ι is an inner automorphism, δ is a diagonal automorphism, γ is a graph automorphism and θ is a field automorphism. The group of central automorphisms are too small and DLP in field automorphisms reduces to a discrete logarithm in the field $\mathbb{F}_q$. So there is no benefit of using these in a MOR cryptosystem. The graph automorphism is just a transpose inverse map. Thus, the only automorphisms which show a potential to be a right candidate for MOR cryptosystem with $SL(d, q)$ are conjugation automorphisms. Note that, our automorphisms are presented as action on generators. An inner automorphism can be thought as a linear operator on a d^2-dimensional algebra of matrices. Thus for a fixed basis, DLP in $< \phi >$ can be reduced to DLP in $GL(d^2, q)$. Thus, if we can recover the conjugating matrix from the action on generators, then the security is $\mathbb{F}_{q^d}$, if not then the security is $\mathbb{F}_{q^{d^2-1}}$ because characteristics polynomial of ϕ can have an irreducible factor of degree at most $d^2 - 1$ ($\phi(1) = 1$ implies 1 is always an eigenvalue). So from these, we conclude that for a secure MOR cryptosystem we must look at automorphisms that act by conjugation, as the inner automorphisms. Inner automorphisms form a normal subgroup of $Aut(G)$ and usually constitute the bulk of automorphisms.

If ϕ is an inner automorphism, say $\iota_g \colon X \mapsto gXg^{-1}$, we are interested in determinng the conjugating element g. Note that the group homomorphism $G \to \mathrm{Inn}(G)$ given by $g \mapsto \iota_g$ is surjective. Thus if G is generated by

$g_1, g_2, \ldots, g_s$ then $\mathrm{Inn}(G)$ is generated by $\iota_{g_1}, \ldots, \iota_{g_s}$. Let ϕ be an inner automorphism and if we can find $g_j, j = 1, 2, \ldots, r$, generators, such that $\phi = \prod_{j=1}^{r} \iota_{g_j}$ then $\phi = \iota_g$, where $g = \prod_{j=1}^{r} g_j$. This implies that our problem is equivalent to solving the word problem in $\mathrm{Inn}(G)$. Note that solving word problem depends on how the group is represented and it is not invariant under group homomorphisms. Thus the Gaussian elimination algorithm used to solve the word problem in the special linear group does not help us in the above situation.

Suppose we have given ϕ and ϕ^x, want to find x. Without loss of generality, we can assume that ϕ is an inner automorphism. To find the value of x, the standard trick used is to recover the conjugating matrix up to scalar multiple and then solved DLP problem in matrices using the Menezes and Wu's idea [35]. Here we briefly describe the procedure of recovering the conjugating matrix.

Recovering the conjugating matrix

As we discussed earlier inner automorphisms $\phi : X \mapsto gXg^{-1}$, for some matrix g, are the best candidates for MOR cryptosystem with $SL(d, q)$. Recall that the automorphisms used for MOR cryptosystem are presented as an action on generators. So, the strategy to recover the conjugating matrix g is to use the action of automorphisms on elementary matrices $x_{i,j} = I + te_{i,j}$ for $i \neq j$. Let $g = [C_1, C_2, \ldots, C_d]$, where C_i for $i = 1, \ldots, d$ are the columns of the matrix g. Consider the action of ϕ on the elementary matrices: $\phi(x_{i,j}(t)) = g(I + te_{i,j})g^{-1} = I + tge_{i,j}g^{-1}$. Observe that the effect of multiplying $e_{i,j}$ on left with g^{-1} gives us a matrix whose all rows are zero except the i^{th} row, which is nothing but the j^{th} row of g^{-1}. Since g is non-singular implies that at least one of the entry in each row has to be non-zero. Now if we compute $ge_{i,j}g^{-1}$ then we can observe that each of the column is

a constant multiple of the i^{th} column of g and one of these columns must be non-zero. Next, we compute $g(I + e_{i,i+1})g^{-1} - I$ for each $1 \leq i \leq d-1$ and choose a non-zero column C'_i. From these C'_i construct a $d \times d$ matrix B whose first $d-1$ columns are C'_i, $1 \leq i \leq d-1$ and the d^{th} column is a non-zero column of $g(I + e_{d,1})g^{-1} - I$. As each column of this matrix is a constant multiple of i^{th} column of g, we can decompose B as $B = gD$, where $D = \text{diag}(a_1, \ldots, a_d)$ is a diagonal matrix with each diagonal entry is equal to the constant multiple in the respective column of B. Thus, we have recovered the matrix g up to a diagonal matrix. Further, we can see that $B^{-1}\phi(X)B = D^{-1}XD$ and hence $B^{-1}(I + e_{i,j})B - I = a_i^{-1}a_j e_{i,j}$. In particular, compute $B^{-1}(I + e_{i,j})B - I$ for $j = 1$, $i = 1, 2, \ldots, d$ and form a matrix $D' = \text{diag}(1, a_1a_2^{-1}, a_1a_3^{-1}, \ldots, a_1a_d^{-1})$. Finally, we observe that $BD' = a_1 g$ and hence we have found g up to a scalar multiple. That means from ϕ we can recover g up to scalar multiple.

Similarly, we can recover g^x up to scalar multiple from ϕ^x say bg^x. Thus from $a_1 g$ and bg^x we compute $(ag)^{q-1} = g^{q-1}$ and $(bg^x)^{q-1} = g^{x(q-1)}$. Once we get rid of the scalar multiples, we recover the value of x by solving DLP in a matrix group $< g^{q-1} >$. Thus the DLP in $\langle\phi\rangle$, where ϕ is conjugation automorphism reduces to the DLP in $\langle g^{q-1}\rangle$. Observe that if we choose g nicely such that $g^{q-1} = 1$ one can avoid the above attack.

However, one could use the eigenvalues of g to bypass that argument and recover the value of x by solving the DLP in a finite field $\mathbb{F}_{q^d}$ as follows: Suppose $\lambda_1, \lambda_2, \cdots, \lambda_d$ are the eigenvalues of g. Then $\lambda_1^x, \lambda_2^x, \ldots, \lambda_d^x$ becomes the eigenvalues of g^x and thus ag has eigenvalues $a\lambda_1, a\lambda_2, \ldots, a\lambda_d$; bg^x has eigenvalues $b\lambda_1^x, b\lambda_2^x, \ldots, b\lambda_d^x$. Next, compute $\frac{\lambda_i}{\lambda_j}$ and $(\frac{\lambda_i}{\lambda_j})^x$, by taking quotients. Notice that these quotients belong to $\mathbb{F}_{q^d}$. As there is no unique way to order the eigenvalues, one might not be able to match a quotient with its power. Then we might have to deal with several quotients. If we choose d

moderately large then one need to do a significant number of computations to get a match. However, in most of practical applications we choose d small and so this search is not going to cost much; moreover one can do this in parallel. Once we get a match, we can solve the DLP in $\mathbb{F}_{q^d}$ and recover the value of x. Thus, the DLP in $\langle\phi\rangle$ reduces in DLP in $\mathbb{F}_{q^d}$.

For the implementation of MOR cryptosystem, it is very important to have an efficient algorithm to solve the world problem; one can use the Gaussian elimination algorithm to take care of that part. The public key depends on the number of generators of the group. Thus, the less number of generators of the group, the better the cryptosystem. To settle this problem one can use two generators due to Albert and Thompson [3], $C = 1 + e_{d-1,2} + e_{d,1}$ and $D = (-1)^d(e_{1,2} - e_{2,3} + \sum_{i=1}^{d} e_{i,i+1})$. One can recover the elementary matrices from these two generators constructively [29, Theorem 8.1] thus one can use them effectively to publish the public key.

In this thesis, we have studied the MOR cryptosystem using finite orthogonal groups. We will address both of the above-mentioned questions about the key size and word problem.

Chapter 3

Chevalley groups

This chapter contains some basic definitions, terminology, notation and known theory that serve as a prerequisite for this thesis. We will describe the elementary matrices which we have used later in our algorithms. In section 3.1, we briefly describe the construction of Chevalley groups, we also describe several subgroups of Chevalley groups which we need in the construction of twisted Chevalley groups [section 3.2]. In section 3.3, we give the definitions of various automorphisms of Chevalley groups which will be used in later chapters, and we finally state Theorem 3.4.1 of Dieudonne which helps us to choose the right automorphisms of finite orthogonal groups to build a secure MOR cryptosystem.

To perform the row-column operations, our algorithm in chapter 4 uses the set of elementary matrices which are nothing but the Chevalley generators which comes from the theory of Chevalley groups. For the sake of completeness of this thesis, we briefly describe the theory of Chevalley groups. Throughout this chapter, we follow the notations from [11] and [17]. The reason we are taking this route, is to handle all the orthogonal groups of our interest uniformly.

3.1 Chevalley groups

A Lie algebra is called simple if it is non-Abelian and has no non-zero proper ideals. To define the Chevalley groups, we first describe the construction of Chevalley basis.

Chevalley basis

Let L be a simple Lie algebra over $\mathbb{C}$. It is well known that a simple Lie algebra can be decomposed as $L = H \oplus \sum_{r\in\Phi} L_r$, where H is a Cartan subalgebra of L, Φ is a root system of L and for $r \in \Phi \subseteq H^*$, H^* is a dual of H and $L_r = \{x \in L \mid [x,h] = r(h)x,\ \forall h \in H\}$. For $r \in \Phi$, $h_r := \frac{2r}{(r,r)}$ is called the co-root of r. Here the scalar product $(\ ,\ ) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is the usual inner product on the Euclidean space containing the roots. Thus we have:

$$(r,s) = ||r||\,||s||\cos(\theta),$$

where θ is the angle between the roots r and s. Define the reflection in the hyperplane orthogonal to r,

$$w_r(x) = x - 2\frac{(r,x)}{(r,r)}r.$$

The group generated by the reflections w_r , $r \in \Phi$, is called the Weyl group of Φ. Let L be a simple Lie algebra over $\mathbb{C}$. We are interested in constructing the Chevalley group over an arbitrary field K, so in order to do that one needs to pass by a basis of L, called Chevalley basis. By fixing an order in the Euclidean space, we get a system of positive roots Φ^+ and negative roots Φ^- so that $\Phi = \Phi^+ \bigcup \Phi^-$. Let $e_r \in L_r$ for $r \in \Phi^+$ be a non-zero element of L_r then there exists unique elements $e_{-r} \in L_{-r}$ for $r \in \Phi^+$, such that $[e_r, e_{-r}] = h_r$. Let $\Pi = \{p_1, p_2, \ldots, p_l\}$ be a system of simple roots, i.e., any root is either non-positive or non-negative integer linear combination of simple roots. We state the Chevalley basis theorem which helps in the construction of Chevalley groups over an arbitrary field.

Theorem 3.1.1. [11, Theorem 4.2.1] Let L be a simple Lie algebra over $\mathbb{C}$ and $L = H \oplus \sum_{r\in\Phi} L_r$ be a Cartan decomposition of L. Let h_r be a co-root corresponding to the root r. Then, for each root $r \in \Phi$, an element e_r can be chosen in L_r such that

$$[e_r, e_{-r}] = h_r,$$
$$[e_r, e_s] = \pm(p+1)e_{r+s},$$

where p is the greatest integer for which $s - pr \in \Phi$.

The elements $\{h_r, r \in \Pi; e_r, r \in \Phi\}$ form a basis for L, called a Chevalley basis. The elements multiply together as follows:

$$[h_r, h_s] = 0,$$
$$[h_r, e_s] = A_{rs}e_s,$$
$$[e_r, e_{-r}] = h_r,$$
$$[e_r, e_s] = 0 \quad if \quad r + s \notin \Phi,$$
$$[e_r, e_s] = N_{r,s}e_{r+s} \quad if \quad r + s \in \Phi,$$

where $A_{rs} := \frac{2(r,s)}{(r,r)}$, $N_{r,s} = \pm(p+1)$ and $-pr+s, \cdots, -r+s, s, r+s, \cdots, qr+s$ is a r-chain passing through s. The multiplication constants of the algebra with respect to Chevalley basis are all integers.

Chevalley group is a subgroup of the automorphism group of the simple Lie algebra L. Chevalley groups over an arbitrary field can be constructed by generalizing the construction of the Chevalley group of a simple Lie algebra defined over the complex numbers. The generators of the Chevalley group over a given field K are constructed with the help of a Chevalley basis of the Lie algebra, the ad-functor and the exponential map.

The Exponential Map

The generators for Chevalley group are obtained by applying the exponential map on the functions $ad(e_r)$, $r \in \Phi$, where ad is the Lie algebra homomorphism $ad : L \to End(L)$ given by $ad(x).y = [x, y]$.

Definition 3.1.1. Let L be a Lie algebra defined over a field of characteristic 0. Then define the exponential map by, $\exp : Der(L) \to Aut(L)$,

$$\delta \to \exp\delta := 1 + \delta + \frac{\delta^2}{2!} + \cdots + \frac{\delta^{n-1}}{(n-1)!}, \quad if \quad \delta^n = 0.$$

Note that the map $\exp\delta$ is, in fact, an automorphism of L [11, section 4.3].

Since $ad(e_r)$ are nilpotent derivations, the elements $x_r(\xi) := \exp(\xi ad(e_r))$ are therefore automorphisms of L for $\xi \in \mathbb{C}$, $r \in \Phi$. Action of these automorphisms on the Chevalley basis is given as follows, if r and s are linearly independent:

$$\begin{aligned} x_r(\xi)e_r &= e_r, \\ x_r(\xi)e_{-r} &= e_{-r} + \xi h_r - \xi^2 e_r, \\ x_r(\xi)h_r &= h_r - 2\xi e_r, \\ x_r(\xi)h_s &= h_s - A_{sr}\xi e_r, \\ x_r(\xi)e_s &= \Sigma_{i=0}^{q} M_{r,s,i}\xi^i e_{ir+s}, \end{aligned}$$

with $M_{r,s,i} := \frac{1}{i!}N_{r,s}N_{r,r+s}\cdots N_{r,(i-1)r+s} = \pm\binom{p+i}{i}$, where p is defined by the r-chain through s. Observe that the automorphisms $x_r(\xi)$, $r \in \Phi$, $\xi \in \mathbb{C}$ transform the basis elements into linear combinations of basis elements. The coefficients are integral multiples of positive powers of ξ [11, section 4.3].

Now, we are ready to define the Chevalley group of type L over $\mathbb{C}$. The group $L(\mathbb{C}) := \langle x_r(\xi) \mid r \in \Phi,\ \xi \in \mathbb{C}\rangle$ is called Chevalley group of type L over $\mathbb{C}$. The property of the automorphisms $x_r(\xi)$ transforming the basis elements into a linear combinations of basis elements with the coefficients which are integral multiples of positive powers of ξ enables us to define automorphisms of this type over an arbitrary field.

Let L be a simple Lie algebra defined over $\mathbb{C}$ and let $\{h_r,\ r \in \Pi;\ e_r,\ r \in \Phi\}$ be a Chevalley basis of L. Furthermore, let $L_{\mathbb{Z}}$ be the subset of L consisting of all integer linear combinations of the Chevalley basis elements. Note that $L_{\mathbb{Z}}$ becomes a Lie algebra over $\mathbb{Z}$. Let K be an arbitrary field and define $L_K := K \bigotimes L_{\mathbb{Z}}$. Since $L_{\mathbb{Z}}$ and K are additive abelian groups, L_K, is also an additive abelian group. Every element a of L_K can be written as follows:

$$a = \sum_{r\in\Pi} \lambda_r(1_K \otimes h_r) + \sum_{r\in\Phi} \mu_r(1_K \otimes e_r),$$

where 1_K is the unit element of K and $\lambda_r, \mu_r \in K$. Define scalar multiplica-

tion as

$$ka = \sum_{r\in\Pi} k\lambda_r(1_K \otimes h_r) + \sum_{r\in\Phi} k\mu_r(1_K \otimes e_r),$$

with this scalar multiplication, L_K becomes a K-vector space. If we write $h_r^{'} := 1_K \otimes h_r$ and $e_r^{'} := 1_K \otimes e_r$ then $\{h_r^{'}, r \in \Pi;\ e_r^{'}, r \in \Phi\}$ is a basis of the vector space L_K, called a Chevalley basis of L_K. Then one can define a Lie algebra structure on L_K as follows:

$$[1 \otimes x, 1 \otimes y] : = 1 \otimes [x, y],$$

for basis elements x, y and extended by linearity.

Thus L_K is a Lie algebra over K. Moreover, the multiplication constants of L_K with respect to $\{h_r^{'}, r \in \Pi;\ e_r^{'},\ r \in \Phi\}$ are the same as the multiplication constants of L with respect to the basis $\{h_r, r \in \Pi;\ e_r, r \in \Phi\}$ [11, page 63] reduced to the field K.

Let $A_r(\xi)$ denote the matrix corresponding to $x_r(\xi)$ with respect to the Chevalley basis $\{h_r, r \in \Pi;\ e_r,\ r \in \Phi\}$ of L. The entries of $A_r(\xi)$ have the form $a\xi^i$, where a and i are integers, $i \geq 0$. Obtain the matrix $A_r^{'}(t)$ from $A_r(\xi)$ by replacing each coefficient $a\xi^i$ by $a^{'}t^i$, where $a^{'}$ is the element of the prime field of K corresponding to $a \in \mathbb{Z}$ and $t \in K$. Without loss of anything we can suppress dash from the notation.

Let $x_r(t) : L_K \to L_K$ be the linear map corresponding to the matrix $A_r(t)$ with respect to the basis $\{h_r,\ r \in \Pi;\ e_r,\ r \in \Phi\}$. In fact, $x_r(t)$ is an automorphism of L_K, $\forall\ r \in \Phi,\ t \in K$ [11, proposition 4.4.2]. The elements $x_r(t)$ act on the Chevalley basis of L_K in the same way as the elements $x_r(\xi)$ on the Chevalley basis of L.

Chevalley group

The **Chevalley group** of type L over the field K is the following group of Lie algebra automorphisms of L_K :

$$L(K) : = \langle x_r(t) \mid r \in \Phi,\ t \in K \rangle.$$

The Chevalley group $L(K)$ is determined up to isomorphism by the simple Lie algebra L over $\mathbb{C}$ and the field K [11, proposition 4.4.3].

It is well known that there are four infinite families of simple lie algebras (of classical type) over $\mathbb{C}$ and denoted as $A_l(l \geq 1)$, $B_l(l \geq 2)$, $C_l(l \geq 3)$ and $D_l(l \geq 4)$. The main tool used in the classification of these algebras is Dynkin diagram. Let $A_l(K)$, $B_l(K), C_l(K)$ and $D_l(K)$ be the Chevalley groups corresponding to the simple Lie algebras A_l, B_l, C_l and D_l respectively. In the following section, we show that the Chevalley groups $A_l(K)$, $B_l(K)$, $C_l(K)$ and $D_l(K)$ are isomorphic to certain classical groups [11, section 11.2].

3.2 Identification of Chevalley groups with classical groups

Let $G = L(K)$ be the Chevalley group of type L over K and denote $\bar{G}$ be the group of matrices generated by the elements $\exp(te_r)$ for all $r \in \Phi$ and all $t \in K$. By [11, theorem 4.5.1] we have

$$\exp(t\ ad\, e_r) \cdot x = \exp(te_r) \cdot x \cdot \exp(te_r)^{-1}, \quad \text{for all}\, x \in L_K.$$

Thus there is a homomorphism σ of $\bar{G}$ onto $G = L(K)$ such that kernel of σ is the center Z of $\bar{G}$ [11, Lemma 11.3.1]. The group $\bar{G}$ is close to groups of our interest. In the later section, we will abuse the notation slightly and denote the generators of $\bar{G}$ as $x_r(t)$. To understand the diagonal automorphisms of Chevalley groups $G = L(K)$ let's define some subgroups of it.

3.2.1 Subgroups of Chevalley groups

Definition 3.2.1. For $r \in \Phi$, the subgroups $X_r := \langle x_r(t) \mid t \in K \rangle$ are called root subgroups of G.

Now for a fixed simple root system Π of the root system Φ define U to be a subgroup of G generated by the root subgroups X_r, $r \in \Phi^+$ and define V to be the subgroup of G generated by the root subgroups X_r, $r \in \Phi^-$.

For every $r \in \Phi$ there is a surjective homomorphism [11, Theorem 6.3.1] $\phi_r : SL(2,K) \to \langle X_r, X_{-r} \rangle$, which maps $\begin{pmatrix} 1 & t \\ 0 & 1 \end{pmatrix}$ to $x_r(t)$ and $\begin{pmatrix} 1 & 0 \\ t & 1 \end{pmatrix}$ to $x_{-r}(t)$. Let us define $h_r(\lambda)$ as $\phi_r(\begin{pmatrix} \lambda & 0 \\ 0 & \lambda^{-1} \end{pmatrix})$ and $n_r(t)$ as $\phi_r(\begin{pmatrix} 0 & t \\ t^{-1} & 0 \end{pmatrix})$. Set $n_r = n_r(1)$ for notational convenience. We now define some important subgroups, $H := \langle h_r(t) \mid r \in \Phi,\ t \in K^* \rangle$ and $N := \langle H,\ n_r \mid r \in \Phi \rangle$.

In order to identify the Chevalley groups with classical groups, we will first describe a matrix representation of each of the simple Lie algebras A_l, B_l, C_l and D_l.

3.2.2 Type A_l

The A_l type complex Lie algebra is $\mathrm{sl}_{l+1}(\mathbb{C})$ consisting of trace 0 matrices of size $l+1$. The set of all diagonal matrices in $sl_{l+1}(\mathbb{C})$ is a Cartan subalgebra, and the Cartan decomposition with respect to this gives a Chevalley basis. The roots (eigenvectors for non-zero eigenvalues) which are part of Chevalley basis is given by $\Phi = \{e_{i,j} \mid 1 \leq i \neq j \leq l\}$, where $e_{i,j}$ is the elementary matrix with (i, j)-coefficient 1 and other coefficient 0. A Chevalley basis is obtained from taking union of Φ with the set $\{[e_{i,i+1},\ e_{i+1,i}] \mid 1 \leq i \leq l\}$ where the bracket operation is given by $[X,Y] = XY - YX$. Thus the generators for $\bar{G}$ of type A_l over the field K are $x_{i,j}(t) = I + te_{i,j}$, where $i \neq j$ and $t \in K$. Hence $\bar{G} = \mathrm{SL}(l+1,K)$ and $A_l(K) \cong \mathrm{PSL}(l+1,K)$.

It can be checked that the set of $n \times n$ matrices X satisfying ${}^TXA + AX = 0$, where A is an $n \times n$ matrix over $\mathbb{C}$ form a Lie algebra [11, Lemma 11.2.2]. Moreover, if X is nilpotent matrix satisfying this condition then

$$ {}^T(\exp X)A(\exp X) = A. $$

3.2.3 Type B_l

The B_l type Lie algebra is $\mathrm{o}_{2l+1}(\mathbb{C}) = \{X \in M(2l+1, \mathbb{C}) \mid {}^TX\beta + \beta X = 0\}$, where β is given by $\begin{pmatrix} 2 & 0 & 0 \\ 0 & 0 & I_l \\ 0 & I_l & 0 \end{pmatrix}$.

Following the theory of Lie algebra, we index rows by $0, 1, \cdots, l, -1, \cdots, -l$. The set of diagonal matrices gives a Cartan subalgebra and the Cartan decomposition gives us a Chevalley basis. Thus the roots in this case are $\Phi = \{e_{i,j} - e_{-j,-i}, -e_{-i,-j} + e_{j,i},\ e_{i,-j} - e_{j,-i},\ -e_{-i,j} + e_{-j,i},\ 2e_{i,0} - e_{0,-i},\ -2e_{-i,0} + e_{0,i} \mid 1 \leq i < j \leq l\}$. The simple roots are $\Pi = \{e_{i,i+1} - e_{-(i+1),-i},\ 2e_{l,0} + e_{0,-l} \mid 1 \leq i \leq l-1\}$.

Let $G = B_l(K)$, then $\bar{G}$ is a group of matrices generated by elements X satisfying ${}^TX\beta X = \beta$. Such matrices represent isometries of the quadratic form $Q(x) = x_0^2 + x_1x_{-1} + x_2x_{-2} + \cdots + x_lx_{-l}$ and $\bar{G}$ is a subgroup of $\mathrm{O}_{2l+1}(K, Q)$ generated by the matrices

$$\begin{aligned}
x_{i,j}(t) &= I + t(e_{i,j} - e_{-j,-i}) && \text{for } i \neq j,\\
x_{i,-j}(t) &= I + t(e_{i,-j} - e_{j,-i}) && \text{for } i < j,\\
x_{-i,j}(t) &= I + t(e_{-i,j} - e_{-j,i}) && \text{for } i < j,\\
x_{i,0}(t) &= I + t(2e_{i,0} - e_{0,-i}) - t^2e_{i,-i},\\
x_{0,i}(t) &= I + t(-2e_{-i,0} + e_{0,i}) - t^2e_{-i,i},
\end{aligned}$$

where $1 \leq i, j \leq l$ and $t \in K$. Thus, in this case $\bar{G} = \Omega_{2l+1}(K)$ and the Chevalley group $B_l(K)$ is isomorphic to $\mathrm{P}\Omega_{2l+1}(K)$. With the above mentioned generators, the elements $d(\xi) = \mathrm{diag}(1, \underbrace{1, \ldots, 1, \xi}_{l}, \underbrace{1, \ldots, 1, \xi^{-1}}_{l})$ and $w_l = I - e_{l,l} - e_{-l,-l} - e_{l,-l} - e_{-l,l}$ generate the orthogonal group $\mathrm{O}(2l+1, K)$.

3.2.4 Type C_l

The C_l type Lie algebra is $\mathrm{sp}_{2l}(\mathbb{C}) = \{X \in \mathrm{M}(2l, \mathbb{C}) \mid {}^TX\beta + \beta X = 0\}$, where β is given by $\begin{pmatrix} 0 & I_l \\ -I_l & 0 \end{pmatrix}$. Following the theory of Lie algebra, we index rows by $1, \ldots, l, -1, \ldots, -l$. The set of diagonal matrices gives a Cartan subalgebra and the Cartan decomposition gives us a Chevalley basis. Thus the roots in this case are $\Phi = \{e_{i,j} - e_{-j,-i}, -e_{-i,-j} + e_{j,i},\ e_{i,-j} + e_{j,-i},\ e_{-i,j} + e_{-j,i},\ e_{i,-i},\ e_{-i,i} \mid 1 \leq i < j \leq l\}$. The simple roots are $\Pi = \{e_{i,i+1} - e_{-(i+1),-i},\ e_{l,-l} \mid 1 \leq i \leq l-1\}$.

Thus the generators for $\bar{G}$ of type C_l over the field K are: For $1 \leq i, j \leq l$,

$$\begin{aligned} x_{i,j}(t) &= I + t(e_{i,j} - e_{-j,-i}) \quad \text{for } i \neq j, \\ x_{i,-j}(t) &= I + t(e_{i,-j} + e_{j,-i}) \quad \text{for } i < j, \\ x_{-i,j}(t) &= I + t(e_{-i,j} + e_{-j,i}) \quad \text{for } i < j, \\ x_{i,-i}(t) &= I + te_{i,-i}, \\ x_{-i,i}(t) &= I + te_{-i,i}, \end{aligned}$$

where $t \in K$. Thus, in this case $\bar{G} = \mathrm{Sp}(2l, K)$ and the Chevalley group $C_l(K)$ is isomorphic to $\mathrm{PSp}(2l, K)$.

3.2.5 Type D_l

The D_l type Lie algebra is $\mathrm{o}_{2l}(\mathbb{C}) = \{X \in \mathrm{M}(2l, \mathbb{C}) \mid {}^TX\beta + \beta X = 0\}$, where β is given by $\begin{pmatrix} 0 & I_l \\ I_l & 0 \end{pmatrix}$. Following the theory of Lie algebra, we index rows by $1, \ldots, l, -1, \ldots, -l$. The set of diagonal matrices gives a Cartan subalgebra and the Cartan decomposition gives us a Chevalley basis. Thus the roots in this case are $\Phi = \{e_{i,j} - e_{-j,-i}, -e_{-i,-j} + e_{j,i},\ e_{i,-j} - e_{j,-i},\ e_{-i,j} + e_{-j,i},\ \mid 1 \leq i < j \leq l\}$. The simple roots are $\Pi = \{e_{i,i+1} - e_{-(i+1),-i}, e_{(l-1),-l} - e_{l,-(l-1)} \mid 1 \leq i \leq l-1\}$.

Let $G = D_l(K)$, then $\bar{G}$ is a group of matrices generated by elements X satisfying ${}^TX\beta X = \beta$. Such matrices represent isometries of the quadratic form $Q(x) = x_1x_{-1} + x_2x_{-2} + \cdots + x_lx_{-l}$ and $\bar{G}$ is a subgroup of $\mathrm{O}_{2l}(K, Q)$. Thus the generators for $\bar{G}$ of type D_l over the field K are: For $1 \leq i, j \leq l$,

$$\begin{aligned} x_{i,j}(t) = &\quad I + t(e_{i,j} - e_{-j,-i}) \quad \text{for } i \neq j, \\ x_{i,-j}(t) = &\quad I + t(e_{i,-j} - e_{j,-i}) \quad \text{for } i < j, \\ x_{-i,j}(t) = &\quad I + t(e_{-i,j} - e_{-j,i}) \quad \text{for } i < j. \end{aligned}$$

where $t \in K$. Thus in this case $\bar{G} = \Omega_{2l}(K)$ and the Chevalley group $B_l(K)$ is isomorphic to $P\Omega_{2l}(K)$. With the above mentioned generators, the elements $d(\xi) = \mathrm{diag}(\underbrace{1, \ldots, 1, \xi}_{l}, \underbrace{1, \ldots, 1, \xi^{-1}}_{l})$ and $w_l = I - e_{l,l} - e_{-l,-l} - e_{l,-l} - e_{-l,l}$ generate the orthogonal group $\mathrm{O}(2l, K)$.

Let us summarize the above groups in the following table:

Table 3.1: Identification of Chevalley groups

Type	Group of our interest	$\bar{G}$	$G = L(K)$
A_l	$\mathrm{SL}(l+1, K)$	$\mathrm{SL}(l+1, K)$	$\mathrm{PSL}(l+1, K)$
B_l	$\mathrm{O}(2l+1, K)$	$\Omega_{2l+1}(K)$	$\mathrm{P}\Omega_{2l+1}(K)$
C_l	$\mathrm{Sp}(2l, K)$	$\mathrm{Sp}(2l, K)$	$\mathrm{PSp}(2l, K)$
D_l	$\mathrm{O}(2l, K)$	$\Omega_{2l}(K)$	$\mathrm{P}\Omega_{2l}(K)$

3.3 Twisted Chevalley groups

We have seen that the Chevalley groups of type A_l, B_l, C_l, D_l can be identified with certain classical groups, but only some of the classical groups can be interpreted as Chevalley groups as described in the previous section. There are classical groups which are not Chevalley groups, for example the unitary groups and the second class of orthogonal groups in even dimension. Steinberg [55] generalized the idea of Chevalley and introduced twisted Chevalley

groups to produce even more new classical groups. These groups are now called Steinberg groups. These groups can be constructed in those cases where the Dynkin diagram has symmetry. We are interested in the group of the type ${}^2D_l(K)$. The exposition here follows [11, Chapters 13 &14] and serves as a motivation for choosing the set of generators described below. Let L be a simple Lie algebra of classical type, and $G = L(K)$ be a Chevalley group of type L over K. Suppose the Dynkin diagram of L has a non-trivial symmetry ρ. Then there is a graph automorphism γ of $L(K)$ corresponding to ρ. We can choose a field automorphism θ such that $\sigma = \gamma\theta$ satisfies $\sigma^n = 1$, where n is the order of ρ. Then by [11, Proposition 13.4.1], we have $\sigma(U) = U$, $\sigma(V) = V$, $\sigma(H) = H$ and $\sigma(N) = N$ where U, V, H and N are subgroups of $L(K)$ described as before. Denote $U_1 = \{x \in U \mid \sigma(x) = x\}$ and $V_1 = \{x \in V \mid \sigma(x) = x\}$. Consider the group G_1 generated by U_1 and V_1. These are called the twisted groups. We describe the twisted orthogonal groups in the following section.

3.3.1 Twisted orthogonal group

For the purpose of computations in chapter 4, we choose a bilinear form β as $\begin{pmatrix} \beta_0 & 0 & 0 \\ 0 & 0 & I_{l-1} \\ 0 & I_{l-1} & 0 \end{pmatrix}$, where β_0 is $\begin{pmatrix} 1 & 0 \\ 0 & \epsilon \end{pmatrix}$, ϵ is a fixed non-square element of K and I_{l-1} is the identity matrix of size $l-1$ over K. Note that the twisted orthogonal group obtained using Steinberg's construction is with respect to a different basis [11, Theorem, 14.5.2]. Suppose the characteristics of the field is odd and ϵ is a non-square element then the twisted orthogonal group is the set of matrices X satisfying ${}^TX\beta X = \beta$. As one of the consequences of our algorithm, we can prove that the following elementary matrices generate the group G^1.

The generators for the group G^1 are:

$$\begin{aligned}
x_{i,j}(t) &= I + t(e_{i,j} - e_{-j,-i}) && \text{for } i \neq j,\\
x_{i,-j}(t) &= I + t(e_{i,-j} - e_{j,-i}) && \text{for } i < j,\\
x_{-i,j}(t) &= I + t(e_{-i,j} - e_{-j,i}) && \text{for } i < j,\\
x_{i,1}(t) &= I + t(e_{1,i} - 2e_{-i,1}) - t^2 e_{-i,i},\\
x_{1,i}(t) &= I + t(-e_{1,-i} + 2e_{i,1}) - t^2 e_{i,-i},\\
x_{i,-1}(t) &= I + t(e_{-1,i} - 2\epsilon e_{-i,-1}) - \epsilon t^2 e_{-i,i},\\
x_{-1,i}(t) &= I + t(-e_{-1,-i} + 2\epsilon e_{i,-1}) - \epsilon t^2 e_{i,-i},\\
x_1(t,s) &= I + t(e_{1,1} - e_{-1,-1}) - (e_{1,1} + e_{-1,-1}) + s(e_{-1,1} + \epsilon e_{1,-1}),\\
x_2 &= I - 2e_{-1,-1}, \qquad \text{where} \quad s^2 + \epsilon t^2 = 1.
\end{aligned}$$

If the characteristics of field is 2 then the twisted orthogonal group is a group of matrices which represents the isometries of the quadratic form $Q(x) = \alpha(x_1^2 + x_{-1}^2) + x_1x_{-1} + x_2x_{-2} + \cdots + x_lx_{-l}$, where $\alpha t^2 + t + \alpha$ is irreducible. In this case the generators for the group G^1 are:

$$\begin{aligned}
x_{i,j}(t) &= I + t(e_{i,j} - e_{-j,-i}) && \text{for } i \neq j,\\
x_{i,-j}(t) &= I + t(e_{i,-j} - e_{j,-i}) && \text{for } i < j,\\
x_{-i,j}(t) &= I + t(e_{-i,j} - e_{-j,i}) && \text{for } i < j,\\
x_{1,-i}(t) &= I + te_{1,-i} + te_{i,-1} + \alpha t^2 e_{i,-i},\\
x_{-1,-i}(t) &= I + te_{-1,-i} + te_{i,1} + \alpha t^2 e_{i,-i},\\
x_0 &= I + (t-1)e_{1,1} + (s-1)e_{-1,-1} + pe_{1,-1} + re_{-1,1},
\end{aligned}$$

where $ts + pr = 1$.

3.4 Description of automorphisms of classical groups

For a discussion of diagonal automorphisms of Chevalley groups we need the diagonal subgroups of the similitude groups.

Definition 3.4.1. (Diagonal group). The diagonal groups are defined to be the group of non-singular diagonal matrices in the corresponding similitude group and are as follows: in the case of $\mathrm{GO}(2l+1, K)$ it is

$$\{\text{diag } (\alpha, \lambda_1, \ldots, \lambda_l, \mu\lambda_1^{-1}, \ldots, \mu\lambda_l^{-1}) \,|\, \lambda_1, \ldots, \lambda_l, \alpha^2 = \mu \in K^\times\},$$

and in the case of $\mathrm{GO}(2l, K)$ and $\mathrm{GSp}(2l, K)$ it is

$$\{\text{diag } (\lambda_1, \ldots, \lambda_l, \mu\lambda_1^{-1}, \ldots, \mu\lambda_l^{-1}) \,|\, \lambda_1, \ldots, \lambda_l, \mu \in K^\times\}.$$

Conjugation by these diagonal elements produce diagonal automorphisms in the respective Chevalley groups.

Definition 3.4.2. (Orthogonal similitude groups). The orthogonal similitude group is defined as the set of matrices X of size d as follows:

$$\mathrm{GO}(d, K) = \{X \in \mathrm{GL}(d, K) \,|\, {}^TX\beta X = \mu\beta,\ \mu \in K^\times\},$$

where $d = 2l + 1$ or $2l$ and β is a non-degenerate symmetric bilinear form.

Definition 3.4.3. (Symplectic similitude group). The symplectic similitude group is denoted by $\mathrm{GSp}(2l, K) = \{X \in \mathrm{GL}(2l, K) \,|\, {}^TX\beta X = \mu\beta,\ \mu \in K^\times\}$, where β is a non-degenerate symplectic form.

Here μ is called similitude factor which depends on the matrix X. It is easy to see that the similitude factor μ defines a group homomorphism from the similitude group to $K^\times$ and the kernel is the orthogonal group $\mathrm{O}(d, K)$,when β is symmetric and symplectic group $Sp(2l, K)$, when β is skew-symmetric respectively [23, Section 12]. Note that scalar matrices λI for $\lambda \in K^\times$ belong to the center of similitude groups. The similitude groups are thought of analog of what $\mathrm{GL}(d, K)$ is for $\mathrm{SL}(d, K)$.

To build a MOR cryptosystem we need to work with the automorphism group of Chevalley groups. In this section, we describe the automorphism group of classical groups following Dieudonne [14].

Conjugation Automorphisms: For $t \in G$ the map given by $g \mapsto tgt^{-1}$ is an automorphism of G, called an inner automorphism. More generally, if G is a normal subgroup of N then the conjugation maps $g \mapsto ngn^{-1}$ for $n \in N$

are called conjugation automorphisms of G.

Central Automorphisms: Let $\chi : G \to Z(G)$ be a group homomorphism to the center of the group. Then the map $g \mapsto \chi(g)g$ is an automorphism of G, known as the central automorphism. There are no non-trivial central automorphisms for perfect groups, for example, the adjoint Chevalley groups $\mathrm{SL}(l+1, K)$ and $\mathrm{Sp}(2l, K)$, $l \geq 2$. In the case of orthogonal groups, the center is of two elements $\{I, -I\}$. Any map χ maps $\Omega_d(K)$ to identity. This implies that there are at most four central automorphisms in this case.

Field Automorphisms: Let $f \in Aut(K)$. Then the map $x_r(t) \mapsto x_r(f(t))$ for all $r \in \phi$ and $t \in K$ extends to an automorphism of G. These are called field automorphisms. In terms of matrices these amount to replacing each term of the matrix by its image under f.

Graph Automorphisms: Symmetry of Dynkin diagram induces such automorphisms. This way we get automorphisms of order 2 for $A_l(K)$, $l \geq 2$ and $D_l(K)$, $l \geq 4$. We also get an automorphism of order 3 for $D_4(K)$. This map is given by $x_r(t) \mapsto x_{\bar{r}}(\gamma_r t)$, where $r \mapsto \bar{r}$ is Dynkin diagram automorphism and $\gamma_r = \pm 1$. In the case of A_l for $l \geq 2$, the map $x \mapsto A^{-1T} x^{-1} A$ where

$$A = \begin{pmatrix} & & & & 1 \\ & & & -1 & \\ & & 1 & & \\ & -1 & & & \\ \iddots & & & & \end{pmatrix}$$

explicitly describes the graph automorphism. In the case of D_l for $l \geq 5$, the graph automorphism is given by $x \mapsto B^{-1} x B$, where B is a permutation matrix obtained from identity matrix of size $2l \times 2l$ by switching the l^{th} row

and $-l^{th}$ row. This automorphism is a conjugation automorphism.

Theorem 3.4.1. [14] Let K be a field of odd characteristic and $l \geq 2$.

1. For the group $\mathrm{SL}(l+1, K)$ any automorphism is of the form $i\gamma\theta$, where i is a conjugation automorphism defined by elements of $\mathrm{GL}(l+1, K)$ and γ is a graph automorphism of A_l type.
2. For the group $\mathrm{O}(d, K)$ any automorphism is of the form $c_\chi i\theta$, where c_χ is a central automorphism, i is a conjugation automorphism by $\mathrm{GO}(d, K)$ elements (this includes the graph automorphism of D_l case).
3. For the group $\mathrm{Sp}(2l, K)$ any automorphism is of the form $i\theta$, where i is a conjugation automorphism by $\mathrm{GSp}(2l, K)$ elements.

In all cases, θ denotes field automorphisms.

In the above theorem, conjugation automorphisms are given by conjugation by elements of a larger group and it includes the group of inner automorphisms. We introduce diagonal automorphisms to make it more precise. The conjugation automorphism i can be written as a product of i_g and δ where i_g is an inner automorphism and δ is a diagonal automorphism.

Diagonal Automorphisms: In the case of A_l the diagonal automorphisms are given by conjugation by diagonal elements of $\mathrm{PGL}(l+1, K)$ on $A_l(q) = \mathrm{PSL}(l+1, K)$. In the case of B_l, C_l and D_l the diagonal automorphisms are given by conjugation by the corresponding diagonal group defined as above.

Let K be a finite field and $G = \mathrm{L}(K)$ be a Chevalley group over K. Steinberg described the automorphisms of these groups. We have the following theorem [11, Theorem 12.5.1] and [55],

Theorem 3.4.2. [55] Let $G = \mathrm{L}(K)$ where L is simple, and $K(= F_q)$ is a finite field. Let $\phi \in Aut(G)$. Then there exist inner, diagonal, graph and field automorphisms, denoted by i, δ, γ and θ respectively, such that $\phi = i\delta\gamma\theta$.

Chapter 4
Algorithms

To build an effective MOR cryptosystem with finite orthogonal group(s) $\mathrm{O}(d, K)$, we need an efficient algorithm to solve the word problem. The reason of this importance is immediate - Recall that in MOR cryptosystem public key is presented as an action of an automorphism ϕ on the generators. In order to encrypt a message $m \in \mathrm{O}(d, K)$, one need to compute $\phi(m)$ for an arbitrary element m, thus the word problem in orthogonal groups must be solved. In this chapter, we present an algorithm to solve the word problem for twisted orthogonal groups over a field of odd characteristics and orthogonal groups over a field of characteristics 2. Our algorithm works for a field of characteristics zero as well. We also describe the Gaussian elimination algorithm developed in [8] for the split orthogonal groups.

The basic idea of the algorithm is to use the fact that multiplying any orthogonal matrix by any one of the elementary generators enables us to perform row or column operations. The algorithm is slightly different for matrices of even and odd sizes. For the shake of simplicity and to keep the presentation uniform we give different algorithms based on the size d.

Gaussian elimination algorithms play very important role in computational group theory. Gaussian elimination algorithms are seen as a subprocess of the constructive matrix group recognition project. Several attempts were made to develop Gaussian elimination algorithms for classical groups.

To name a few, Brooksbank [9, section 5] used the similar idea to that in our algorithm in his constructive group recognition algorithm to solve the word problem for classical groups. Elementary matrices used by Brooksbank for the algorithm are the output of a probabilistic Las Vegas algorithm. So, it's hard to judge, if he has the same elementary matrices as ours. He does not define elementary row-column operations. Moreover, he uses a low-dimensional oracle to solve the word problem. For example, to solve the word problem in twisted orthogonal groups, he assumes the existence of a four-dimensional oracle. Our algorithm is more straightforward and works directly with elementary matrices. It seems that his methods could be modified to produce a Gaussian elimination algorithm in all classical groups in finite fields of all characteristics. However, his treatment depends on the primitive element ρ of $\mathbb{F}_q^\times$ and on expressing $\mathbb{F}_q$ as a finite dimensional vector space over $\mathbb{F}_p$ – the prime subfield. This suggests that his algorithm would only work for finite fields.

Costi [13] develops an algorithm similar to ours using standard generators, these generators are different from elementary matrices. His algorithm assumes the existence of discrete logarithm oracle to solve the discrete logarithm problem in a finite field K. Moreover, Costi's algorithm is recursive and cannot be extended to infinite fields.

Cohen, Murray, and Taylor [41] proposed a generalized algorithm using the row-column operations, using a representation of Chevalley groups. The key idea there was to bring down an element to a maximal parabolic subgroup and repeat the process inductively. Here we use the natural matrix representation of these groups. Thus our algorithm is more direct and works with matrices explicitly and more effectively.

A novelty of our algorithm is that we do not need to assume that the Chevalley generators generate the group under consideration. Thus our al-

gorithm proves independently the fact that these groups are generated by elementary matrices.

This chapter is organized as follows: in section 4.1 we give the description of orthogonal groups in odd characteristics and even characteristics separately. In section 4.2, we present the algorithms for $\mathrm{O}(2l+1, K)$, $\mathrm{O}^+(2l, K)$ and $\mathrm{O}^-(2l, K)$ respectively. The algorithm decsribed for orthogonal groups over a field of odd characteristics works for the orthogonal groups over a field of characteristics 0. Henceforth, we treat the case of orthogonal groups over a field of characteristics 0 as the orthogonal groups over a field of odd characteristics.

Definition 4.0.1. Let V be a vector space over a field K. A **bilinear form** B on V is a map $B : V \times V \longrightarrow K$ which satisfies the following properties:

$$\begin{aligned} B(x_1 + x_2, y) &= B(x_1, y) + B(x_2, y) \\ B(\alpha x, y) &= \alpha B(x, y) \\ B(x, y_1 + y_2) &= B(x, y_1) + B(x, y_2) \\ B(x, \alpha y) &= \alpha B(x, y) \end{aligned}$$

for all $x_1, x_2, x, y_1, y_2, y \in V$, all $\alpha \in K$.

A bilinear form B on a vector space V is called **symmetric** if $B(x, y) = B(y, x)$ for all $x, y \in V$. A bilinear form B is called **alternate** or (**skew-symmetric** if char $K \neq 2$) if $B(x, x) = 0$ for all $x \in V$. By fixing a basis for V we can write $B(x, y) = {}^T x \beta y$, we say B is non-degenerate if $\det(\beta) \neq 0$. The set $V_0 = \{u \in V \mid B(u, v) = 0, \ \forall\ v \in V\}$ is called the kernel of B.

Definition 4.0.2. Let B be a symmetric bilinear form on a vector space V over a field K of odd characteristics. A **quadratic form** on V associated with the symmetric bilinear form B is a map $Q : V \longrightarrow K$ defined as $Q(x) = B(x, x)$ for all $x \in V$.

Definition 4.0.3. Quadratic form Q is said to be **non-singular** if the kernel of the associated bilinear form is zero.

4.1 Orthogonal groups

Let V be a vector space of dimension d over a field K and $Q : V \to K$ be a quadratic form. Consider a symmetric bilinear form $B : V \times V \to K$ and

associate a matrix β to B by fixing a basis. Thus $B(x, y) = {}^T x\beta y$, where x, y are column vectors. Here, we recall the definition of the orthogonal group.

Definition 4.1.1. The orthogonal group associated with Q is defined as:
$$O(d, Q) := \{X \in GL(V) \mid Q(X(v)) = Q(v) \text{ for all } v \in V\}.$$

As the quadratic form is defined in a slightly different way in case of even characteristics, we describe the orthogonal groups for odd and even characteristics separately. By fixing a basis for V, we identify $\mathrm{GL}(V)$ with $\mathrm{GL}(d, K)$ and treat the orthogonal group as a subgroup of the matrix group $\mathrm{GL}(d, K)$.

4.1.1 Orthogonal groups for odd characteristics

In this section, we assume that K is of odd characteristics. Recall that odd characteristics includes characteristics 0 as well. We will work with non-singular (non-degenerate) quadratic form Q, however, when the characteristics of K is odd this corresponds to β being non-degenerate. Note that one can easily recover the bilinear form from the quadratic form Q by the formula
$$B(x, y) = \frac{1}{2}\{Q(x + y) - Q(x) - Q(y)\}$$
and it is easy to see that a matrix X satisfies ${}^T X\beta X = \beta$ if and only if $Q(X(x)) = Q(x)$ for all $x \in V$. We use this relation later in our algorithms.

Let $K(= F_q)$ be a finite field of odd characteristic. We write the dimension of V as $d = 2l + 1$ or $d = 2l$ for $l \geq 1$. If d is odd then there is only one orthogonal group up to conjugation [17, page 79] and thus, we can fix β as below. In this case, the orthogonal group is simply denoted by $\mathrm{O}(2l + 1, q)$. However, up to conjugation, there are two different orthogonal groups [17, page 79] in even dimension $d = 2l$. For the orthogonal group with even dimension, we fix β as below. We denote these orthogonal groups by $\mathrm{O}^+(2l, q)$ and $\mathrm{O}^-(2l, q)$. The later one is known as the twisted orthogonal group. Throughout this chapter, we assume that I_l is the identity matrix of

size l over K.

We consider the non-degenerate symmetric bilinear forms β on a vector space V defined over a field of odd characteristics given by the following matrices:

- Orthogonal group with odd dimension $d = 2l + 1$: We fix a basis of V and index it by $0, 1, 2, \ldots, l, -1, -2, ..., -l$ and $\beta = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 0 & I_l \\ 0 & I_l & 0 \end{pmatrix}$.
- Orthogonal groups with even dimension $d = 2l$: The two non-degenerate symmetric bilinear forms are as follows
  1. We fix a basis of V and index it by $1, 2, ..., l, -1, -2, ..., -l$ and $\beta = \begin{pmatrix} 0 & I_l \\ I_l & 0 \end{pmatrix}$.
  2. For the twisted form, we fix a basis of V and index it by $1, -1, 2, 3, ..., l, -2, -3, ..., -l$ and $\beta = \begin{pmatrix} \beta_0 & 0 & 0 \\ 0 & 0 & I_{l-1} \\ 0 & I_{l-1} & 0 \end{pmatrix}$, where $\beta_0 = \begin{pmatrix} 1 & 0 \\ 0 & \epsilon \end{pmatrix}$ and ϵ is a fixed non-square in K.

4.1.2 Orthogonal groups for even characteristics

Assume that char(K)=2, in this case, the quadratic form is defined in a slightly different way. A quadratic form Q is defined as follows:

$$Q(\lambda x + \mu y) = \lambda^2 Q(x) + \mu^2 Q(y) + \lambda\mu B(x, y)$$

for all $x, y \in V$, $\lambda, \mu \in K$, and $B(x, y)$ a symmetric bilinear form on V which is called the associated bilinear form of Q. Putting $\lambda = \mu$ we can see that $B(x, x) = 0$ and $B(x, y) = B(y, x)$. Thus B is an alternating form and with

the suitable choice of basis for V, we can represent the form B by the rank $2r$ matrix $\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & I_r \\ 0 & I_r & 0 \end{pmatrix}$. Let $V_0 = \{x \in V \mid B(x, y) = 0,\ \forall\ y \in V\}$, a subspace of dimension $d-2r$. Dimension of V_0 is called the defect of Q. Here we recall the definition of non-degenerate quadratic form.

Definition 4.1.2. The quadratic form Q is said to be non-degenerate if no non-zero vector $x \in V_0$ satisfies $Q(x) = 0$.

Hereafter, we will work with non-degenerate quadratic forms. Let $K = F_q$ be a finite field of even characteristics. It is well known that if $\dim(V) = 2l+1$ then there is only one quadratic form up to equivalence [17, Chapter 14]. In this case we fix a basis for V and index it as $0, 1, 2, \ldots, l, -1, -2, \ldots, -l$, thus we can write Q as $Q(x) = x_0^2 + x_1x_{-1} + \ldots + x_lx_{-l}$ with the associated bilinear form $\beta = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & I_l \\ 0 & I_l & 0 \end{pmatrix}$. However, in case of $\dim(V) = 2l$ there are two quadratic forms up to equivalence [17, Chapter 14] given as below

1. We fix a basis of V and index it as $1, 2, \ldots, l, -1, -2, \ldots, -l$ and form is $Q(x) = x_1x_{-1} + \ldots + x_lx_{-l}$ with the associated bilinear form $\beta = \begin{pmatrix} 0 & I_l \\ I_l & 0 \end{pmatrix}$.

2. For the second one (twisted) we rearrange the basis to make the algorithm uniform for both even and odd q and index it by $1, -1, 2, 3, \ldots, l, -2, -3, \ldots, -l$ and the quadratic form is given as $Q(x) = \alpha(x_1^2 + x_{-1}^2) + x_1x_{-1} + \ldots + x_lx_{-l}$ with the associated bilinear form $\beta = \begin{pmatrix} \beta_0 & 0 & 0 \\ 0 & 0 & I_{l-1} \\ 0 & I_{l-1} & 0 \end{pmatrix}$, where $\alpha t^2 + t + \alpha$ is irreducible in $K[x]$

and $\beta_0 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$.

Let Q be a non-degenerate quadratic form as defined above. For fixed basis, we can note that any isometry g satisfying $Q(g(v)) = Q(v)$ for all $v \in V$ also satisfy ${}^T g\beta g = \beta$, however, converse is not true. We denote the orthogonal groups associated with Q by $\mathrm{O}(2l+1, K)$, $\mathrm{O}^+(2l, K)$ and $\mathrm{O}^-(2l, K)$ respectively.

The following algorithms work only for the bilinear forms described above. However, with a proper change of basis, our algorithm works for any equivalent bilinear forms. Our algorithm works well on fields of all characteristics and sizes.

4.2 Gaussian elimination algorithms for orthogonal groups

We first describe the elementary matrices and the row and column operations for the respective groups. These row and column operations are nothing but multiplication by elementary matrices from left and right respectively. Here the elementary matrices used are nothing but the Chevalley generators which follows from the theory of Chevalley groups and are described in the previous chapter.

The basic idea of the algorithm is to use the fact that multiplying any orthogonal matrix by any one of the generators enables us to perform row or column operations. The relation ${}^T g\beta g = \beta$ gives us some compact relations among the blocks of g which can be used to make the algorithm more faster. To make the algorithm simple we will write the algorithm for $\mathrm{O}(2l+1, K)$, $\mathrm{O}^+(2l, K)$ and $\mathrm{O}^-(2l, K)$ separately.

4.2.1 Gaussian elimination algorithm for $\mathbf{O}(2l+1, K)$

Elementary matrices (Chevalley generators) of orthogonal groups of odd size $\mathbf{O}(2l+1, K)$: For $1 \leq i, j \leq l$ and $t \in K$, $\xi \in K^*$

Table 4.1: Elementary matrices for $\mathrm{O}(2l+1, K)$

Char(K)		Elementary matrices	
both	$x_{i,j}(t)$	$I + t(e_{i,j} - e_{-j,-i})$,	for $i \neq j$
	$x_{i,-j}(t)$	$I + t(e_{i,-j} - e_{j,-i})$,	for $i < j$
	$x_{-i,j}(t)$	$I + t(e_{-i,j} - e_{-j,i})$,	for $i < j$
odd	$x_{i,0}(t)$	$I + t(2e_{i,0} - e_{0,-i}) - t^2 e_{i,-i}$,	
	$x_{0,i}(t)$	$I + t(-2e_{-i,0} + e_{0,i}) - t^2 e_{-i,i}$,	
	w_i	$I - e_{i,i} - e_{-i,-i} + e_{i,-i} + e_{-i,i}$,	
even	$x_{i,-i}(t)$	$I + te_{0,-i} + t^2 e_{i,-i}$,	
	$x_{-i,i}(t)$	$I + te_{0,i} + t^2 e_{-i,i}$.	

Elementary matrices for the orthogonal group over a field of even characteristics differs from that of odd characteristics so in table 4.1 we made that distinction and listed them separately in different row according to the characteristics of K.

Let us note the effect of multiplying g by elementary matrices described in table 4.2. We write an element $g \in \mathrm{O}(2l+1, q)$ as $g = \begin{pmatrix} \alpha & X & Y \\ E & A & B \\ F & C & D \end{pmatrix}$, where A, B, C, D are $l \times l$ matrices, X and Y are $1 \times l$ matrices, E and F are $l \times 1$ matrices. Note that any isometry g satisfying the quadratic form Q also satisfy ${}^T g \beta g = \beta$. In case of char(k) even, we can construct the elements w_i, which interchanges the i^{th} row with $-i^{th}$ row, using the relation $w_i = (I + e_{0,i} + e_{-i,i})(I + e_{0,-i} + e_{i,-i})(I + e_{0,i} + e_{-i,i}) = I + e_{i,i} + e_{-i,-i} + e_{i,-i} + e_{-i,i}$. In case of the orthogonal groups $O(2l+1, k)$ over a field of odd characteristics we can construct w_i, which interchanges the i^{th} row with $-i^{th}$ with a sign change in $i^{th}, -i^{th}$ and 0^{th} row, using the relation $w_i = x_{0,i}(-1)x_{i,0}(1)x_{0,i}(-1) = I - 2e_{0,0} - e_{i,i} - e_{-i,-i} - e_{i,-i} - e_{-i,i}$.

Table 4.2: The row and column operations for $O(2l+1, K)$

	Row operations		Column operations
ER1 (both)	$i^{\text{th}} \mapsto i^{\text{th}} + tj^{\text{th}}$ row and $-j^{\text{th}} \mapsto -j^{\text{th}} - t(-i)^{\text{th}}$ row	EC1 (both)	$j^{\text{th}} \mapsto j^{\text{th}} + ti^{\text{th}}$ column and $-i^{\text{th}} \mapsto -i^{\text{th}} - t(-j)^{\text{th}}$ column
ER2 (both)	$i^{\text{th}} \mapsto i^{\text{th}} + t(-j)^{\text{th}}$ row and $j^{\text{th}} \mapsto j^{\text{th}} - t(-i)^{\text{th}}$ row	EC2 (both)	$-i^{\text{th}} \mapsto -i^{\text{th}} - tj^{\text{th}}$ column and $-j^{\text{th}} \mapsto -j^{\text{th}} + ti^{\text{th}}$ column
ER3 (both)	$-i^{\text{th}} \mapsto -i^{\text{th}} - tj^{\text{th}}$ row and $-j^{\text{th}} \mapsto -j^{\text{th}} + ti^{\text{th}}$ row	EC3 (both)	$j^{\text{th}} \mapsto j^{\text{th}} + t(-i)^{\text{th}}$ column and $i^{\text{th}} \mapsto i^{\text{th}} - t(-j)^{\text{th}}$ column
ER4 (odd)	$0^{\text{th}} \mapsto 0^{\text{th}} - t(-i)^{\text{th}}$ row and $i^{\text{th}} \mapsto i^{\text{th}} + 2t0^{\text{th}} - t^2(-i)^{\text{th}}$ row	EC4 (odd)	$0^{\text{th}} \mapsto 0^{\text{th}} + 2ti^{\text{th}}$ column and $(-i)^{\text{th}} \mapsto (-i)^{\text{th}} - t0^{\text{th}} - t^2 i^{\text{th}}$ column
ER5 (odd)	$0^{\text{th}} \mapsto 0^{\text{th}} + ti^{\text{th}}$ row and $(-i)^{\text{th}} \mapsto (-i)^{\text{th}} - 2t0^{\text{th}} - t^2 i^{\text{th}}$ row	EC5 (odd)	$0^{\text{th}} \mapsto 0^{\text{th}} - 2t(-i)^{\text{th}}$ column and $i^{\text{th}} \mapsto i^{\text{th}} + t0^{\text{th}} - t^2(-i)^{\text{th}}$ column
ER6 (even)	$0^{\text{th}} \mapsto 0^{\text{th}} + t(-i)^{\text{th}}$ row and $i^{\text{th}} \mapsto i^{\text{th}} + t^2(-i)^{\text{th}}$ row	EC6 (even)	$(-i)^{\text{th}} \mapsto (-i)^{\text{th}} + t0^{\text{th}} + t^2 i^{\text{th}}$ column
ER7 (even)	$0^{\text{th}} \mapsto 0^{\text{th}} + ti^{\text{th}}$ row and $(-i)^{\text{th}} \mapsto (-i)^{\text{th}} + t^2 i^{\text{th}}$ row	EC7 (even)	$i^{\text{th}} \mapsto i^{\text{th}} + t0^{\text{th}} + t^2(-i)^{\text{th}}$ column
w_i (odd)	Interchange i^{th} and $(-i)^{th}$ rows with a sign change in $i^{th}, -i^{th}$ and 0^{th} rows	w_i (odd)	Interchange i^{th} and $(-i)^{th}$ column with a sign change in $i^{th}, -i^{th}$ and 0^{th} columns
w_i (even)	Interchange i^{th} and $(-i)^{th}$ row	w_i (even)	Interchange i^{th} and $(-i)^{th}$ column

We use these elements in the following algorithm to interchange the rows or columns of matrix g. The main reason the following algorithm works is the closed condition ${}^T g\beta g = \beta$ which gives the following relations:

$$2{}^T XX + {}^T AC + {}^T CA = 0, \tag{4.1}$$

$$2\alpha {}^T X + {}^T AF + {}^T CE = 0, \tag{4.2}$$

$$2\alpha Y + {}^T FB + {}^T ED = 0, \tag{4.3}$$

$$2{}^T XY + {}^T CB + {}^T AD = I_l. \tag{4.4}$$

and the effect of ER1 (both) and EC1 (both) which is the usual Gaussian elimination on A. Using this operation, one can reduce A to a diagonal matrix.

The Gaussian elimination for $\mathbf{O}(2l+1, K)$

- **Step 1**: Use ER1 (both) and EC1 (both) to make A into a diagonal matrix. However, the process changes matrices A, B, C, D, E, F, X, and Y as well. For the sake of notational convenience, we keep calling these changed matrices as A, B, C, D, E, F, X, and Y.

- **Step 2**: Now there will be two cases depending on the rank r of the new matrix A. The rank of A can be easily determined from the number of non-zero diagonal entries.

- **Step 3**: Use ER3 (both) and non-zero diagonal entries of A to make corresponding r rows of C zero.

  (i) If $r = l$ then C becomes zero matrix.

  (ii) If $r < l$ then interchange all zero rows(i^{th}) of A with corresponding rows($-i^{th}$) of C using w_i so that the new C becomes zero matrix.

  Once C becomes zero, note that the relation 4.1 if char(K) is odd or the relation $Q(g(v)) = Q(v)$ if char(K) is even guarantees that X becomes zero. Then relation 4.4 guarantees that A has full rank l which also make D to be diagonal with full rank l. Thus the relation 4.2 shows that F becomes zero as well.

- **Step 4**: Now if char(K) is even then the relation 4.3 guarantees that E becomes zero. If char(K) is odd, then use ER4 (odd) to make E a zero matrix.

- **Step 5**: Use ER2 (both) to make B a zero matrix. For char(K) even the relation $Q(g(v)) = Q(v)$ guarantees that Y is a zero matrix and for char(K) odd the relation 4.3 shows that Y becomes zero.

Thus the matrix g reduces to $\begin{pmatrix} \pm 1 & 0 & 0 \\ 0 & A & 0 \\ 0 & 0 & D \end{pmatrix}$, where $A = \mathrm{diag}(1, \cdots, 1, \lambda)$ and $D = \mathrm{diag}(1, \cdots, 1, \lambda^{-1})$.

Elementary matrices (Chevalley generators) of orthogonal groups of even size $\mathbf{O}^+(2l, K)$:

Elementary matrices for $\mathrm{O}^+(2l, K)$ are independent of $\mathrm{char}(K)$ so we describe them uniformly as follows. We treat Weyl group elements w_i which are used to interchange the rows or columns of a matrix as elementary matrices. These elements can not be constructed using the Chevalley generators.
For $1 \leq i, j \leq l$ and $t \in K$, $\xi \in K^*$,

Table 4.3: Elementary matrices for $\mathrm{O}^+(2l, K)$

Char(K)		Elementary matrices	
both	$x_{i,j}(t)$	$I + t(e_{i,j} - e_{-j,-i})$,	for $i \neq j$
	$x_{i,-j}(t)$	$I + t(e_{i,-j} - e_{j,-i})$,	for $i < j$
	$x_{-i,j}(t)$	$I + t(e_{-i,j} - e_{-j,i})$,	for $i < j$
	w_i	$I - e_{i,i} - e_{-i,-i} + e_{i,-i} + e_{-i,i}$.	

4.2.2 Gaussian elimination algorithm for $\mathbf{O}^+(2l, K)$

The elementary matrices are described in table 4.3. Let us note the effect of multiplying g by elementary matrices. We write $g \in \mathrm{O}^+(2l, K)$ as $g = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$, where A,B,C,D are $l \times l$ matrices.

Note that any isometry g satisfying the quadratic form Q also satisfy ${}^T g \beta g = \beta$. The main reason the following algorithm works is the closed

Table 4.4: The row and column operations for $\mathrm{O}^+(2l, K)$

	Row operations		Column operations
ER1 (both)	$i^{\text{th}} \mapsto i^{\text{th}} + tj^{\text{th}}$ row and $-j^{\text{th}} \mapsto -j^{\text{th}} - t(-i)^{\text{th}}$ row	EC1 (both)	$j^{\text{th}} \mapsto j^{\text{th}} + ti^{\text{th}}$ column and $-i^{\text{th}} \mapsto -i^{\text{th}} - t(-j)^{\text{th}}$ column
ER2 (both)	$i^{\text{th}} \mapsto i^{\text{th}} + t(-j)^{\text{th}}$ row and $j^{\text{th}} \mapsto j^{\text{th}} - t(-i)^{\text{th}}$ row	EC2 (both)	$-i^{\text{th}} \mapsto -i^{\text{th}} - tj^{\text{th}}$ column and $-j^{\text{th}} \mapsto -j^{\text{th}} + ti^{\text{th}}$ column
ER3 (both)	$-i^{\text{th}} \mapsto -i^{\text{th}} - tj^{\text{th}}$ row and $-j^{\text{th}} \mapsto -j^{\text{th}} + ti^{\text{th}}$ row	EC3 (both)	$j^{\text{th}} \mapsto j^{\text{th}} + t(-i)^{\text{th}}$ column and $i^{\text{th}} \mapsto i^{\text{th}} - t(-j)^{\text{th}}$ column
w_i	Interchange i^{th} and $(-i)^{th}$ row		Interchange i^{th} and $(-i)^{th}$ column

condition ${}^T g\beta g = \beta$ which gives the following relations:

$$ {}^T AC + {}^T CA = 0, \tag{4.5}$$

$$ {}^T DB + {}^T BD = 0, \tag{4.6}$$

$$ {}^T DA + {}^T BC = 0. \tag{4.7}$$

The above equation implies among other things, ${}^T CA + {}^T AC = 0$. This implies that ${}^T AC$ is skew-symmetric. The working principle of our algorithm is simple, uses the symmetry of ${}^T AC$. The problem is, for arbitrary A and C, it is not easy to use this symmetry. In our case, we were able to reduce A to a diagonal matrix and then it is relatively straightforward to use this symmetry.

The Gaussian elimination algorithm for $O^+(2l, K)$ is as follows:

- **Step 1**: Use ER1 (both) and EC1 (both) to make A into a diagonal matrix. However, the process changes matrices A, B, C, and D as well. For the sake of notational convenience, we keep calling these changed matrices as A, B, C and D.

- **Step 2**: Now there will be two cases depending on the rank r of matrix A. The rank of A can be easily determined from the number of non-zero diagonal entries.

- **Step 3**: Use ER3 (both) and non-zero diagonal entries of A to make corresponding r rows of C zero.

  (i) If $r = l$ then C becomes zero matrix.

  (ii) If $r < l$ then interchange all zero rows(i^{th}) of A with corresponding rows($-i^{th}$) of C using w_i so that the new C becomes zero matrix.

- **Step 4**: Use ER2 (both) to make B a zero matrix.

Thus the matrix g reduces to $\begin{pmatrix} A & 0 \\ 0 & D \end{pmatrix}$, where $A = \text{diag}(1, \cdots, 1, \lambda)$ and $D = \text{diag}(1, \cdots, 1, \lambda^{-1})$.

4.2.3 Gaussian elimination algorithm for $O^-(2l, K)$

In this section, we describe row-column operations for twisted Chevalley groups. These groups are also known as the Steinberg groups. An element $g \in O^-(2l, K)$ as $g = \begin{pmatrix} A_0 & X & Y \\ E & A & B \\ F & C & D \end{pmatrix}$, where A, B, C, D are $(l-1) \times (l-1)$ matrices, X and Y are $2 \times (l-1)$ matrices, E and F are $(l-1) \times 2$ matrices and A_0 is a 2×2 matrix. In the Gaussian elimination algorithm that we discuss, we reduce X, Y, E & F to zero and A and D to diagonal matrices. However, unlike the previous cases we were unable to reduce A_0 to an identity matrix. However, for odd characteristics we were able to reduce A_0 to a two-parameter subgroup. Furthermore, our algorithm provides for the spinor norm as before. Let us go ahead and talk about the output of the algorithm. In the output we will have a 2×2 block (also called A_0) which will satisfy

${}^TA_0\beta_0A_0 = \beta_0$, where $\beta_0 = \begin{pmatrix} 1 & 0 \\ 0 & \epsilon \end{pmatrix}$ for odd characteristics, as defined earlier. Then A_0 is a orthogonal group given by the bilinear form β_0. Now if we write $A_0 = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$, then we get the following equations:

$$a^2 + c^2\epsilon = 1$$

$$ab + cd\epsilon = 1$$

$$b^2 + d^2\epsilon = \epsilon$$

Considering the fact that $\det(A_0) = \pm 1$, one more equation $ac - bd = \pm 1$ and this leads to two cases either $a = d$ and $b = c\epsilon$ or $a = -d$ and $b = -c\epsilon$. Recall that, since ϵ is not a square, $d \neq 0$. Then if $c = 0$, then there are four choices for A_0 and these are $A_0 = \begin{pmatrix} \pm 1 & 0 \\ 0 & \pm 1 \end{pmatrix}$.

To summarize, in the output of the algorithm A_0 will have either of the six forms

$$\begin{pmatrix} t & -s\epsilon \\ s & t \end{pmatrix} \text{ or } \begin{pmatrix} t & s\epsilon \\ s & -t \end{pmatrix} \text{ where } t^2 + s^2\epsilon = 1$$

$$\text{or } \begin{pmatrix} \pm 1 & 0 \\ 0 & \pm 1 \end{pmatrix}$$

There are now two ways to describe the algorithm, one is to leave A0 as it is in the output of the algorithm and the other is to include these matrices as generators. For the purpose of uniform exposition we choose the later and included the following two generators

$$x_1(t,s) = I + (t-1)e_{1,1} - (t-1)e_{-1,-1} + s(e_{-1,1} + \epsilon e_{1,-1});\ t^2 + \epsilon s^2 = 1$$

$$x_2 = I - 2e_{-1,-1}$$

in the list of elementary generators in Table 4.5. In the case of even characteristics no such reduction is possible and we included the matrix $\begin{pmatrix} t & p \\ r & s \end{pmatrix}$ in the list of generators with the condition that the determinant is 1.

Elementary matrices (Chevalley generators) of twisted orthogonal groups of even size $O^-(2l, K)$: The elementary matrices for $O^-(2l, K)$ depends on the characteristics of K, so we describe them separately in the following table 4.5. Let α be an Arf-invariant, $1 \leq i, j \leq l$ and $t \in K$, $\xi \in K^*$

Table 4.5: Elementary matrices for $O^-(2l, K)$

char(k)		Elementary matrices	
both	$x_{i,j}(t)$	$I + t(e_{i,j} - e_{-j,-i})$,	for $i \neq j$
	$x_{i,-j}(t)$	$I + t(e_{i,-j} - e_{j,-i})$,	for $i < j$
	$x_{-i,j}(t)$	$I + t(e_{-i,j} - e_{-j,i})$,	for $i < j$
	w_i	$I - e_{i,i} - e_{-i,-i} + e_{i,-i} + e_{-i,i}$	for $2 \leq i \leq l$
odd	$x_{i,1}(t)$	$I + t(e_{1,i} - 2e_{-i,1}) - t^2 e_{-i,i}$	for $2 \leq i \leq l$
	$x_{1,i}(t)$	$I + t(-e_{1,-i} + 2e_{i,1}) - t^2 e_{i,-i}$	for $2 \leq i \leq l$
	$x_{i,-1}(t)$	$I + t(e_{-1,i} - 2\epsilon e_{-i,-1}) - \epsilon t^2 e_{-i,i}$	for $2 \leq i \leq l$
	$x_{-1,i}(t)$	$I + t(-e_{-1,-i} + 2\epsilon e_{i,-1}) - \epsilon t^2 e_{i,-i}$	for $2 \leq i \leq l$
	$x_1(t, s)$	$I + (t-1)e_{1,1} - (t+1)e_{-1,-1}$ $+s(e_{-1,1} + \epsilon e_{1,-1})$,	where $s^2 + \epsilon t^2 = 1$
	x_2	$I - 2e_{-1,-1}$	
even	$x_{1,-i}(t)$	$I + te_{1,-i} + te_{i,-1} + \alpha t^2 e_{i,-i}$	for $2 \leq i \leq l$
	$x_{-1,-i}(t)$	$I + te_{-1,-i} + te_{i,1} + \alpha t^2 e_{i,-i}$	for $2 \leq i \leq l$
	x_0	$I + (t-1)e_{1,1} + (s-1)e_{-1,-1} + pe_{1,-1}$ $+re_{-1,1}$,	where $ts + pr = 1$.

Elementary matrices for the orthogonal group in even characteristics differs from that of odd characteristics so in above table we made that distinction and listed them separately in the different rows according to the characteristics of K. The elementary matrices are described in table 4.5. The Weyl group elements w_i which we use to interchange columns or rows can not be constructed using the Chevalley generators, so we include them in the table of elementary matrices. Let us note the effect of multiplying g by elementary matrices. We write an element $g \in O^-(2l, K)$ as $g = \begin{pmatrix} A_0 & X & Y \\ E & A & B \\ F & C & D \end{pmatrix}$, where

A,B,C,D are $(l-1)\times(l-1)$ matrices, X and Y are $2\times(l-1)$ matrices, E and F are $(l-1)\times 2$ matrices and A_0 is a 2×2 matrix.

Table 4.6: The row operations for $O^-(2l,K)$

	Row operations
ER1$_{\text{(both)}}$	$i^{\text{th}} \mapsto i^{\text{th}} + tj^{\text{th}}$ row and and $-j^{\text{th}} \mapsto -j^{\text{th}} - t(-i)^{\text{th}}$ row
ER2 $_{\text{(both)}}$	$i^{\text{th}} \mapsto i^{\text{th}} + t(-j)^{\text{th}}$ row and $j^{\text{th}} \mapsto j^{\text{th}} - t(-i)^{\text{th}}$ row
ER3$_{\text{(both)}}$	$-i^{\text{th}} \mapsto -i^{\text{th}} - tj^{\text{th}}$ row and $-j^{\text{th}} \mapsto -j^{\text{th}} + ti^{\text{th}}$ row
ER4$_{\text{(odd)}}$	$1^{\text{st}} \mapsto 1^{\text{st}} - t(-i)^{\text{th}}$ row and $i^{\text{th}} \mapsto i^{\text{th}} + 2t1^{\text{st}} - t^2(-i)^{\text{th}}$ row
ER5$_{\text{(odd)}}$	$1^{\text{st}} \mapsto 1^{\text{st}} + ti^{\text{th}}$ row and $(-i)^{\text{th}} \mapsto (-i)^{\text{th}} - 2t1^{\text{st}} - t^2i^{\text{th}}$ row
ER6$_{\text{(odd)}}$	$(-1)^{\text{th}} \mapsto (-1)^{\text{th}} - t(-i)^{\text{th}}$ row and $i^{\text{th}} \mapsto i^{\text{th}} + 2\epsilon t(-1)^{\text{th}} - \epsilon t^2(-i)^{\text{th}}$ row
ER7$_{\text{(odd)}}$	$(-1)^{\text{th}} \mapsto (-1)^{\text{th}} + ti^{\text{th}}$ row and $(-i)^{\text{th}} \mapsto (-i)^{\text{th}} - 2\epsilon t(-1)^{\text{th}} - \epsilon t^2i^{\text{th}}$ row
ER8 $_{\text{(even)}}$	$1^{\text{st}} \mapsto 1^{\text{st}} + t(-i)^{\text{th}}$ row and $i^{\text{th}} \mapsto i^{\text{th}} + t(-1)^{\text{th}} + \alpha t^2(-i)^{\text{th}}$ row
ER9$_{\text{(even)}}$	$(-1)^{\text{th}} \mapsto (-1)^{\text{th}} + t(-i)^{\text{th}}$ row and $i^{\text{th}} \mapsto i^{\text{th}} + t1^{\text{st}} + \alpha t^2(-i)^{\text{th}}$ row
$w_{i\text{(both)}}$	Interchange i^{th} and $(-i)^{th}$ row

Table 4.7: The column operations for $O^-(2l,K)$

	Column operations
EC1$_{\text{(both)}}$	$j^{\text{th}} \mapsto j^{\text{th}} + ti^{\text{th}}$ column and $-i^{\text{th}} \mapsto -i^{\text{th}} - t(-j)^{\text{th}}$ column
EC2$_{\text{(both)}}$	$-i^{\text{th}} \mapsto -i^{\text{th}} - tj^{\text{th}}$ column and $-j^{\text{th}} \mapsto -j^{\text{th}} + ti^{\text{th}}$ column
EC3$_{\text{(both)}}$	$j^{\text{th}} \mapsto j^{\text{th}} + t(-i)^{\text{th}}$ column and $i^{\text{th}} \mapsto i^{\text{th}} - t(-j)^{\text{th}}$ column
EC4$_{\text{(odd)}}$	$1^{\text{st}} \mapsto 1^{\text{st}} + 2ti^{\text{th}}$ column and $(-i)^{\text{th}} \mapsto (-i)^{\text{th}} - t1^{\text{st}} - t^2i^{\text{th}}$ column
EC5$_{\text{(odd)}}$	$1^{\text{st}} \mapsto 1^{\text{st}} - 2t(-i)^{\text{th}}$ column and $i^{\text{th}} \mapsto i^{\text{th}} + t1^{\text{st}} - t^2(-i)^{\text{th}}$ column
EC6 $_{\text{(odd)}}$	$(-1)^{\text{th}} \mapsto (-1)^{\text{th}} + (2\epsilon t)i^{\text{th}}$ column and $(-i)^{\text{th}} \mapsto (-i)^{\text{th}} - t(-1)^{\text{th}} - \epsilon t^2i^{\text{th}}$ column
EC7 $_{\text{(odd)}}$	$(-1)^{\text{th}} \mapsto (-1)^{\text{th}} - 2\epsilon t(-i)^{\text{th}}$ column and $i^{\text{th}} \mapsto i^{\text{th}} + t(-1)^{\text{th}} - \epsilon t^2(-i)^{\text{th}}$ column
EC8 $_{\text{(even)}}$	$(-1)^{\text{th}} \mapsto (-1)^{\text{th}} + ti^{\text{th}}$ column and $(-i)^{\text{th}} \mapsto (-i)^{\text{th}} + t1^{\text{st}} + \alpha t^2i^{\text{th}}$ column
EC9$_{\text{(even)}}$	$1^{\text{st}} \mapsto 1^{\text{st}} + ti^{\text{th}}$ column and $(-i)^{\text{th}} \mapsto (-i)^{\text{th}} + t(-1)^{\text{th}} + \alpha t^2i^{\text{th}}$ column
$w_{i\text{(both)}}$	Interchange i^{th} and $(-i)^{th}$ column

Note that any isometry g satisfying the quadratic form Q also satisfy ${}^Tg\beta g = \beta$. The main reason the following algorithm works is the closed

condition ${}^Tg\beta g = \beta$ which gives the following relations:

$$ {}^TA_0\beta_0 A_0 + {}^TFE + {}^TEF = \beta_0, \tag{4.8} $$

$$ {}^TA_0\beta_0 X + {}^TFA + {}^TEC = 0, \tag{4.9} $$

$$ {}^TA_0\beta_0 Y + {}^TFB + {}^TED = 0, \tag{4.10} $$

$$ {}^TX\beta_0 X + {}^TCA + {}^TAC = 0, \tag{4.11} $$

$$ {}^TX\beta_0 Y + {}^TCB + {}^TAD = I_{l-1}. \tag{4.12} $$

and the effect of ER1 (both) and EC1 (both) which is the usual Gaussian elimination on A. Using this operation, one can reduce A to a diagonal matrix.

The Gaussian elimination algorithm for $O^-(2l, K)$ is as follows:

- **Step 1**: Use ER1 and EC1 to make A into a diagonal matrix. However, the process changes matrices A_0, A, B, C, D, E, F, X, and Y as well. For the sake of notational convenience, we keep calling these changed matrices as A_0, A, B, C, D, E, F, X, and Y.

- **Step 2**: Now there will be two cases depending on the rank r of matrix A. The rank of A can be easily determined using the number of non-zero diagonal entries.

- **Step 3**: Use ER3 and non-zero diagonal entries of A to make corresponding r rows of C zero.

  (i) If $r = l - 1$ then C becomes zero matrix.

  (ii) If $r < l - 1$ then interchange all zero rows(i^{th}) of A with corresponding rows ($-i^{th}$) of C using w_i so that the new C becomes zero matrix.

  Once C becomes zero one can note that the relation ${}^TX\beta_0 X + {}^TCA + {}^TAC = 0$ if char(k) is odd or the relation $Q(g(v)) = Q(v)$ and the

fact that $\alpha t^2 + t + \alpha$ is irreducible if char(k) is even guarantees that X becomes zero [see lemma 4.2.1].

Then the relation ${}^TA_0\beta_0 X + {}^TFA + {}^TEC = 0$ shows that F becomes zero as well and the relation ${}^TX\beta_0 Y + {}^TCB + {}^TAD = I_{l-1}$ guarantees that A has full rank $l-1$ which also makes D to be diagonal with full rank $l-1$. Now we diagonalize A again to the form $\begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda \end{pmatrix}$ as in step 1.

- **Step 4**: Use EC4 (odd) and EC6 (odd) if char(K) is odd or use EC8 (even) and EC9 (even) if char(K) is even to make E a zero.

Note that the relation ${}^TA_0\beta_0 A_0 + {}^TFE + {}^TEF = \beta_0$ shows that A_0 is invertible. Thus the relation ${}^TA_0\beta_0 Y + {}^TFB + {}^TED = 0$ guarantees that Y becomes zero.

- **Step 5**: Use ER2 to make B a zero matrix. Thus the matrix g reduces to: $g = \begin{pmatrix} A_0 & 0 & 0 \\ 0 & A & 0 \\ 0 & 0 & D \end{pmatrix}$, where $A = \text{diag}(1, 1, \cdots, 1, \lambda)$ and $D = \text{diag}(1, 1, \cdots, 1, \lambda^{-1})$. Now if char$(K)$ is odd then go to step 6 otherwise go to step 7.

- **Step 6**: Using the relation ${}^TA_0\beta_0 A_0 = \beta_0$, check that A_0 has the form: $A_0 = \begin{pmatrix} t & -\epsilon s \\ s & t \end{pmatrix}$ or $A_0 = \begin{pmatrix} t & \epsilon s \\ s & -t \end{pmatrix}$. If the determinant of A_0 is -1 then multiply g by elementary matrix x_2 to get new g of the above form in which A_0 has determinant 1. Now using the elementary matrix $x_1(t, s)$ we can reduce g to $\begin{pmatrix} I_2 & 0 & 0 \\ 0 & A & 0 \\ 0 & 0 & D \end{pmatrix}$, where A and D are as above.

- **Step 7**: Using elementary matrix x_0 we can reduce g to $\begin{pmatrix} I_2 & 0 & 0 \\ 0 & A & 0 \\ 0 & 0 & D \end{pmatrix}$, where $A = \text{diag}(1, 1, \cdots, 1, \lambda)$ and $D = \text{diag}(1, 1, \cdots, 1, \lambda^{-1})$.

Lemma 4.2.1. Let k be a field of characteristics 2 and let $g = \begin{pmatrix} A_0 & X & Y \\ E & A & B \\ F & 0 & D \end{pmatrix}$, where $A = \text{diag}(1, 1, \cdots, 1, \lambda)$, be an element of $O^-(2l, k)$ then $X = 0$.

Proof: Let $\{e_1, e_{-1}, e_2, \cdots, e_l, e_{-2}, \cdots, e_{-l}\}$ be the standard basis of the vector space V. Recall that the action of the quadratic form Q on a column vector x is given by $Q(x) = \alpha(x_1^2 + x_{-1}^2) + x_1x_{-1} + \ldots + x_lx_{-l}$, where $x = (x_1, x_{-1}, x_2, \cdots, x_l, x_{-2}, \cdots, x_{-l})^t$ and $\alpha t^2 + t + \alpha$ is irreducible over $k[t]$. By definition, for any $g \in O^-(2l, k)$ we have $Q(g(x)) = Q(x)$ for all $x \in V$. Let $X = \begin{pmatrix} x_{11} \cdots x_{1(l-1)} \\ x_{21} \cdots x_{2(l-1)} \end{pmatrix}$ be a $2 \times (l-1)$ matrix. Computing $Q(g(e_i)) = Q(e_i)$ for all $2 \leq i \leq l$, we can see that $\alpha(x_{1i}^2 + x_{2i}^2) + x_{1i}x_{2i} = 0$. If $x_{2i} = 0$ then we can see that $x_{1i} = 0$. Suppose $x_{2i} \neq 0$ for some i then we rewrite the equation by dividing it by x_{2i} as $\alpha(\frac{x_{1i}}{x_{2i}})^2 + \frac{x_{1i}}{x_{2i}} + \alpha = 0$, which is a contradiction to the fact that $\alpha t^2 + t + \alpha$ is irreducible over $k[t]$. Thus, $x_{2i} = 0$ for all $2 \leq i \leq l$ and hence $X = 0$.

4.2.4 Time complexity of the algorithms

We establish that the time complexity of the above algorithms is $\mathcal{O}(l^3)$.

- Making A as a diagonal matrix by row-column operations has a complexity $\mathcal{O}(l^3)$.
- In worst case making C and D to be zero has complexity $\mathcal{O}(l^2)$.
- Making E zero has complexity $\mathcal{O}(l)$.

- Thus the complexity of the above algorithm is $\mathcal{O}(l^3)$.

As a consequence of our algorithms, we can note the following applications.

Remark 4.1. In an orthogonal group, above algorithms can be used to check if an element has determinant 1, i.e., belongs to the special orthogonal group. This can be done in case of $O^+(2l, K)$ and $O(2l+1, K)$ by counting the number of times the elementary matrix w_i was used, and in case of $O^-(2l, K)$ can be done by counting the number of times the elementary matrices w_i and x_2 were used. Recall that all elementary matrices, other than w_i and x_2 are of determinant 1 and w_i, x_2 are of determinant -1. Hence, if we used an even number of w_i then the determinant of that element is 1, else -1.

Remark 4.2. The above algorithm can be used to determine, for an orthogonal group in odd characteristics, if an element belongs to the commutator subgroup Ω. This can be done using the Spinor norm. We compute Spinor norm explicitly [Appendix A] and we show that the image of λ, the output in Step 5 for $O(2l+1, K)$ and step 4 for $O^+(2l, K)$ in $K^\times/K^{\times 2}$ is the Spinor norm of the matrix. In case of $O^-(2l, K)$, if $s = 0$ in the step 6, then the image of λ (output of step 6) in $K^\times/K^{\times 2}$ is the spinor norm of g. Otherwise, an image of $2(1-t)\lambda\epsilon$ in $K^\times/K^{\times 2}$ is the spinor norm of g. So if the image of λ is 1 then that element belongs to the kernel of the Spinor norm. Furthermore, if it has determinant 1, it is in the commutator Ω. This gives an efficient membership test for Ω as a subset of the orthogonal group.

Chapter 5

MOR cryptosystem with finite orthogonal groups

In this chapter, we discuss the MOR cryptosystem with orthogonal groups and analyze its security. In section 5.1, we discuss how to choose right automorphisms for MOR cryptosystem with orthogonal groups and show that for a secure MOR cryptosystem we must look at automorphisms that act by conjugation, as the inner automorphisms. In section 5.2, we analyze the security of proposed MOR cryptosystem. We present a reduction attack which reduces the discrete logarithm problem in $\langle\phi\rangle$ to the discrete logarithms in finite extensions of $\mathbb{F}_q$. We show that embedding degree of orthogonal group $O(d, q)$ with q even and d odd is equal to $d-1$ and also we conjecture that embedding degree for other orthogonal groups is d^2-1. The basic idea of the reduction attack is to recover the conjugating matrix of the conjugation automorphism, once we recover the conjugating matrix then discrete logarithm problem in $\langle\phi\rangle$ reduces to the discrete logarithm problem in matrices. Next, we use the Menezes and Wu's algorithm to reduce the DLP in matrices to a DLP in finite field $\mathbb{F}_{q^d}$.

Recall that in MOR cryptosystem public key is presented as an action of ϕ on the generators, thus in practical implementation of the MOR cryptosystem, two things matter the most. First, to compute $\phi(g)$, there should be an efficient algorithm to solve the word problem. Second, smaller the number

of generators better is the MOR cryptosystem.

We have addressed the first problem of solving the word problem in chapter 4. We have developed a Gaussian elimination algorithm to solve the word problem in $O(d, q)$. However, the generators used to solve the word problem are large in numbers but no need to worry. In section 5.2, we show that one can reduce the key size effectively using the Steinberg generators in case of orthogonal groups over the prime field $\mathbb{F}_p$, where $p \equiv 3 \pmod 4$.

5.1 Security of the proposed MOR cryptosystem

The purpose of this section is to show that for a secure MOR cryptosystem over the orthogonal groups we have to look at automorphisms that act by conjugation, like the inner automorphisms. There are other automorphisms that also act by conjugation, like the diagonal automorphism and the graph automorphism for odd-order orthogonal groups. Then we argue about the hardness of our security assumptions.

Let ϕ be an automorphism of one of the orthogonal groups G: $O(2l+1, q)$, $O^+(2l, q)$, or $O^-(2l, q)$. The automorphisms of these groups are described in chapter 4. It is well known that $\phi = c_\chi \iota\delta\gamma\theta$ [55, theorem 3.2.2], where c_χ is a central automorphism, ι is an inner automorphism, δ is a diagonal automorphism, γ is a graph automorphism, and θ is a field automorphism.

The group of central automorphisms is too small, and the discrete logarithm in field automorphisms reduces to a discrete logarithm in the field $\mathbb{F}_q$. So there is no benefit of using these in a MOR cryptosystem. Also, there are not many graph automorphisms in orthogonal groups other than odd-order orthogonal groups. In the odd-order orthogonal groups graph automorphisms act by conjugation. Recall here that, our automorphisms are presented as action on generators. It is clear [Section 2.4.2] that if we can recover the conjugating matrix from the action on generators, then the security is $\mathbb{F}_{q^d}$, if

not then the security is $\mathbb{F}_{q^{d^2-1}}$.

So from these, we conclude that for a secure MOR cryptosystem we must look at automorphisms that act by conjugation, as the inner automorphisms. Inner automorphisms form a normal subgroup of $Aut(G)$ and usually constitute the bulk of automorphisms. If ϕ is an inner automorphism, say $\iota_g \colon x \mapsto gxg^{-1}$, we would like to determine the conjugating element g.

For the special linear group, it was done in [29]. We will follow the steps there for the present situation too. However, before we do that, let us digress briefly to observe that $G \to \mathrm{Inn}(G)$ given by $g \mapsto \iota_g$ is a surjective group homomorphism. Thus, if G is generated by $g_1, g_2, \ldots, g_s$ then $\mathrm{Inn}(G)$ is generated by $\iota_{g_1}, \ldots, \iota_{g_s}$. Let $\phi \in \mathrm{Inn}(G)$. If we can find $g_j, j = 1, 2, \ldots, r$ generators, such that $\phi = \prod_{j=1}^{r} \iota_{g_j}$ then $\phi = \iota_g$ where $g = \prod_{j=1}^{r} g_j$. This implies that our problem is equivalent to solving the word problem in $\mathrm{Inn}(G)$. Note that solving word problem depends on how the group is represented and it is not invariant under group homomorphisms. Thus the algorithm described in previous chapter to solve the word problem in the orthogonal groups does not help us in the present case.

5.1.1 Reduction of security

In this subsection, we show that for orthogonal group $\mathrm{O}(d, q)$, where q is even and d odd, the security of the MOR cryptosystem is the hardness of the discrete logarithm problem in $\mathbb{F}_{q^{d-1}}$. This is the same as saying that we can find the conjugating matrix up to a scalar multiple. We further show that the method that works for $O(d, q)$ with q is even, d odd and special linear groups [section 2.4.2] does not work for other orthogonal groups. From now onwards, other orthogonal groups mean $O^+(2l, q)$, $O^-(2l, q)$ for all q and $O(2l + 1, q)$ with q odd.

Case of $\mathbf{O}(d, q)$ with q even and d odd:

Let $d = 2l + 1$ and ϕ be an automorphism that works by conjugation, i.e., $\phi = \iota_g$ for some g. We recover the matrix $h = \alpha(g)$ up to scalar multiple, where α is an isomorphism between $O(2l + 1, q)$ and $\mathrm{Sp}(2l, q)$ which maps $x_{i,j}(t)$ to $x_{i,j}(t)$, $x_{i,-j}(t)$ to $x_{i,-j}(t)$, $x_{-i,j}(t)$ to $x_{-i,j}(t)$, $I + te_{o,-i} + t^2e_{i,-i}$ to $I + t^2e_{i,-i}$ and $I + te_{o,i} + t^2e_{-i,i}$ to $I + t^2e_{-i,i}$ respectively (see the table 4.1 for generators of O(2l+1, q) and section 3.2.4 for generators of symplectic groups). Note that given an inner automorphism $\phi = \iota_g$ of $\mathrm{O}(2l+1, q)$, we get an inner automorphism $\alpha \circ \phi = \iota_h$. We follow the idea used in section 2.4.2 to recover h up to scalar multiple, by using the generators $I+se_{i,-i}$ and $I+se_{-i,i}$. Write h in the column form as $h = [C_1, \ldots, C_l, C_{-1}, \ldots, C_{-l}]$. Consider the action of $\alpha \circ \phi = \iota_h$ on elementary matrices $I + se_{i,-i}$ and $I + se_{-i,i}$: $h(I + se_{i,-i})h^{-1} = I + she_{i,-i}h^{-1}$. Note that $[C_1, \ldots, C_l, C_{-1}, \ldots, C_{-l}]e_{i,-i} = [0, \ldots, 0, 0, \ldots, 0, C_i, 0, \ldots, 0]$, where C_i is at the $-i^{th}$ place. Multiplying this further with h^{-1} we get each column as scalar multiple of C_i, say d_i. Similarly, we compute $[C_1, \ldots, C_l, C_{-1}, \ldots, C_{-l}]e_{-i,i} = [0, \ldots, 0, C_{-i}, 0, \ldots, 0, 0, \ldots, 0]$, where C_{-i} is at i^{th} place. Multiplying this further with h^{-1} we get each column as a scalar multiple of C_{-i}, say d_{-i}. Thus using the similar trick as in section 2.4.2 we compute $h(I + e_{i,-i})h^{-1}$, $h(I + e_{-i,i})h^{-1}$ and choose columns C'_i, C'_{-i} (which are up to scalar multiple of C_i, C_{-i} respectively) for each $i = 1, \ldots, l$. Now, we construct a matrix $B = [C'_1, \ldots, C'_l, C'_{-1}, \cdots, C'_{-l}]$. Since each C'_i and C'_{-i} are scalar multiple of C_i and C_{-i} implies $B = hD$, where D is a diagonal matrix $\mathrm{diag}(d_1, \ldots, d_l, d_{-1}, \ldots, d_{-l})$.

Next, we compute $B^{-1}\alpha \circ \phi(x_r(t))B = D^{-1}h^{-1}(hx_r(t)h^{-1})hD = I + D^{-1}e_rD$ which is equivalent to computing $D^{-1}e_rD$ for $r \in \Phi$. First compute $D^{-1}(e_{i,j} - e_{-j,-i})D$ to get $d_i^{-1}d_j$ and $d_{-i}^{-1}d_{-j}$ for $i \neq j$. Next, compute $D^{-1}(e_{i,-i} - e_{-i,i})D$ to get $d_id_{-i}^{-1}$, $d_{-i}d_i^{-1}$. Then we form a diagonal matrix $\mathrm{diag}(1, d_2^{-1}d_1, \ldots, d_l^{-1}d_1, (d_{-1}^{-1}d_{-2})(d_{-2}^{-1}d_2)(d_2^{-1}d_1), \ldots, (d_{-l}^{-1}d_{-1})(d_{-1}^{-1}d_1))$, where

$(-i)^{th}$ diagonal entry is $(d_{-i}^{-1}d_{-1})(d_{-1}^{-1}d_1)$ and multiply it with B to get d_1h. Thus we have determined h up to scalar multiple. Similarly, we determine h^x up to scalar say bh^x. Now, computing $(d_1h)^{q-1}$ and $(bh^x)^{q-1}$ we get rid of scalars, and if $h^{q-1} \neq I$ then solve DLP in $\langle h^{q-1} \rangle$ using Menezes, Wu [35] procedure by reducing it to the DLP in $\mathbb{F}_{q^{2l}}$ to get the value of x. Thus DLP in $\langle \phi \rangle$ reduces to the DLP in $\mathbb{F}_{q^{2l}}$. However, if we choose g such that $g^{q-1} = I$, then it seems that we might avoid this line of attack. No worries we can bypass this argument by recovering the scalars a and b and then to determine x we compute the discrete logarithm in $\langle g \rangle$ using Menezes and Wu's idea. We prove the following proposition.

Proposition 5.1.1. Given any $g \in \mathrm{Sp}(2l, q)$ up to scalar multiple ag, $a \in \mathbb{F}_q$. If $\gcd(d, q-1) = 1$, we can determine the scalar a. Otherwise one can find the scalar a by solving a discrete logarithm problem in $\mathbb{F}_q$.

Proof: We can recover the scalar a as follows: Let $\{\lambda_1, \ldots, \lambda_d\}$ be a set of eigenvalues of g then the eigenvalues of ag are $\{a\lambda_1, \ldots, a\lambda_d\}$. Set $\alpha = a\lambda_1 \cdots a\lambda_d$ and thus $\alpha = a^d$ as $\lambda_1 \cdots \lambda_d = det(g) = 1$. Suppose $\gcd(d, q-1) = r$, using extended Euclidean algorithm we find l and m such that $ld + m(q-1) = r$. Next, computing α^l, we get $a^{ld} = a^{r-m(q-1)} = a^r$. Thus, if $\gcd(d, q-1) = 1$ then we have recovered the scalar a otherwise we can recover the scalar by solving the discrete logarithm problem in $\mathbb{F}_q$. Let ξ be a primitive element in $\mathbb{F}_q$. Suppose $a = \xi^i$ for some i. Thus, $a^r = \xi^{ir}$ and hence we determine ir and hence i by solving the discrete logarithm in ξ and ξ^{ir}. Hence, we can recover the scalar a from ag. □

Thus, if $\gcd(d, q-1) = 1$ then using above proposition we can recover the scalars a and b from ag and bg^x respectively. Otherwise one need to solve discrete logarithm problem in $\mathbb{F}_q$ to recover the scalars. Now, we can recover g and g^x from ag and bg^x just by multiplying with scalar matrices $a^{-1}\mathbf{I}$ and $b^{-1}\mathbf{I}$ respectively. Finally, we recover x using Menezes and Wu's idea. Thus,

if we choose g such that $g^{q-1} = 1$ and $\gcd(d, q-1) \neq 1$ then to solve the discrete logarithm in $\langle\phi\rangle$ one needs to solve the discrete logarithm in $\mathbb{F}_q$ and $\mathbb{F}_{q^d}$.

Remark 5.1. In the context of MOR cryptosystem using $\mathrm{O}(d, q)$, where d-odd and q-even, if we choose g such that $g^{q-1} = I$ and $\gcd(d, q-1) \neq 1$ then to solve the discrete logarithm in $\langle\phi\rangle$ one needs to solve the discrete logarithm in $\mathbb{F}_q$ and $\mathbb{F}_{q^d}$. Thus, embedding degree for MOR cryptosystem is $d-1$.

Now, we will show that the above line of attack won't work for other orthogonal groups.

Theorem 5.1.1. Let $g \in GO(d, q)$ with the exception of case d-odd and q-even. Consider the conjugation automorphism $\phi : O(d, q) \longrightarrow O(d, q)$. Let $\{x_r\}$, $r \in \Phi$ be a set of Chevalley generators of $O(d, q)$ described earlier. Suppose that the public-key is presented as an action of ϕ on $\{x_r\}$ then it is impossible to recover a matrix gD, where D is a diagonal matrix using the above reduction.

Proof: We prove the theorem for $O^+(d, q)$, d even and the theorem follows for other cases similarly. Let $d = 2l$ and we write g in columns form as $g = [C_1, \ldots C_l, C_{-1}, \ldots, C_{-l}]$. To see the effect of conjugation by g on elementary matrices [Table 4.3], we compute ge_rg^{-1} which gives the following equations:

1. Note that $g(e_{i,j} - e_{-j,-i})g^{-1} = [0, \ldots, 0, C_i, 0 \ldots, 0, C_{-j}, 0, \ldots]\, g^{-1}$, where C_i is a at j^{th} place and C_{-j} is at a $-i^{\text{th}}$ place. After multiplying by g^{-1} we get a matrix whose all columns are linear combinations of columns C_i and C_{-j}.

2. Note that $g(e_{i,-j} - e_{j,-i})g^{-1} = [0, \ldots, 0, C_i, 0 \ldots, 0, C_j, 0, \ldots]\, g^{-1}$, where C_i is at a $-j^{\text{th}}$ place and C_j is at a $-i^{\text{th}}$ place. After multiplying by g^{-1} we get a matrix whose all columns are linear combinations of columns C_i and C_j.

3. Note that $g(e_{-i,j} - e_{-j,i})g^{-1} = [0, \ldots, 0, C_{-i}, 0 \ldots, 0, C_{-j}, 0, \ldots]\, g^{-1}$,

where C_{-i} is at a j^{th} place and C_{-j} is at a i^{th} place. After multiplying by g^{-1} we get a matrix whose all columns are linear combinations of columns C_{-i} and C_{-j}.

Suppose one can construct a matrix B from columns obtained above such that $B = gD$, where D is diagonal, then we can see that $d_iC_i = a_iC_j + b_jC_k$ for some i, j, k which is a contradiction as $\det(g) \neq 0$. Thus, it is not possible to construct a matrix B such that $B = gD$, where D is diagonal.

This conclusively proves that the attack on the special linear group [section 2.4.2] and symplectic groups won't work for most orthogonal groups. □

At this stage, the best we can do is the following: We can construct B such that $B = g(D_1 + PD_2)$, where D_1, D_2 are diagonal and P is a permutation matrix. We construct a matrix B as follows: For each $i = 1, \ldots, l-1$ compute $g(I + e_{i,i+1} - e_{-(i+1),-i})g^{-1} - I$ whose each of column is a linear combination of C_i and $C_{-(i+1)}$. Choose one of its column say $r_iC_i + s_iC_{-(i+1)}$ for each $i = 1, \ldots, l-1$. Similarly compute $g(I + e_{i+1,i} - e_{-i,-(i+1)})g^{-1} - I$ and choose $r_{-i}C_{-i} + s_{-i}C_{(i+1)}$ for each $i = 1, \ldots, l-1$. Further, we compute $g(I + e_{1,-l} - e_{l,-1})g^{-1} - I$ to get $r_lC_l + s_lC_1$ and $g(I + e_{-1,l} - e_{-l,1})g^{-1} - I$ to get $r_{-l}C_{-l} + s_{-l}C_{-1}$.

We set $B = [r_1C_1 + s_1C_{-2}, \ldots, r_{l-1}C_{l-1} + s_{l-1}C_{-l}, r_lC_l + s_lC_1, r_{-1}C_{-1} + s_{-1}C_2, \ldots, r_{-(l-1)}C_{-(l-1)} + s_{-(l-1)}C_l, r_{-l}C_{-l} + s_{-l}C_{-1}]$. Now it is easy to note that $B = g(D_1 + PD_2)$, where $D_1 = \text{diag}(r_1, \ldots, r_l, r_{-1}, \ldots, r_{-l})$, $D_2 = \text{diag}(s_1, \ldots, s_l, s_{-1}, \ldots, s_{-l})$ and P is permutation matrix corresponding to the permutation of indexing set $1 \to -2 \to 3 \to -4 \to \cdots \to l-1 \to -l \to -1 \to 2 \to -3 \to 4 \to \cdots \to -(l-1) \to l \to 1$.

Note that $B = D_1 + PD_2$ is not a diagonal matrix. Consider $B^{-1}\phi(x_r(t))B = (D_1 + PD_2)^{-1}e_r(D_1 + PD_2)$. Since $D_1 + PD_2$ is not diagonal and there are no elementary matrices of the form $I + e_{i,-i}$, $I + e_{-i,i}$ in orthogonal groups it seems that the only way to solve DLP in $\langle\phi\rangle$ is to solve

DLP in $\mathbb{F}_{q^{d^2-1}}$. One can do similar computations for $O(2l+1,q)$, q odd and $O^-(2l,q)$.

Remark 5.2. An observant reader would ask the question: why does this attack works for the special linear and symplectic groups but not for orthogonal groups? The answer lies in a closer look at the generators (elementary matrices) for these groups.

In the special linear groups, the generators are the elementary transvections of form $I + te_{i,j}$, where $i \neq j$ and $t \in \mathbb{F}_q$. Then the attack goes on smoothly as we saw earlier[section 2.4.2]. However, when we look at generators of the form $I + te_{i,j} - te_{-j,-i}$, where $t \in \mathbb{F}_q$ and $i \neq j$; conjugating by them gets us a linear sum of the i^{th} and $(-j)^{\text{th}}$ column, not a scalar multiple of one particular column. This stops the attack from going forward. However in the case of $O(2l+1,q)$ with q even we go to the symplectic group through an isomorphism α, which have generators of form $I + e_{i,-i}$ and $I + e_{-i,i}$ for $1 \leq i \leq l$. These generators make the attack possible for the symplectic groups and hence for $O(2l+1,q)$, q even. However, there are no such generators for other orthogonal groups, and so this attack turns out to be impossible for other orthogonal groups. We summarize the above discussion as follows:

Remark 5.3. MOR cryptosystem with orthogonal group $O(d,q)$ with q-even and d-odd has embedding degree equal to $d-1$. However, it seems likely that the embedding degree of other orthogonal groups is d^2-1.

Remark 5.4. Since there is no unique way to order the elements in finite field $\mathbb{F}_q$, we can avoid the reduction attack on MOR cryprosystem with any orthogonal group by choosing g such that $g^{q-1} = 1$ and d large. Thus, we can conclude that the embedding degree of MOR cryptosystem using orthogonal groups is d^2-1

5.2 Reduction of key-size

One serious objection against an MOR cryptosystem is the size of the key [39, section 7]. Recall that the size of the public-key depends on the number of generators. Less the number of generators better is the cryptosystem. In this

section, we address this issue for the orthogonal groups over prime field $\mathbb{F}_p$, where $p \equiv 3 \pmod 4$. Ree [48] proved that the elementary matrices without w_i-the row interchange matrices, generate Ω the commutator subgroup of the orthogonal group. However, in between the commutator and the whole group, there is another important subgroup, $W\Omega = \langle \Omega, w_i \rangle$ for some i. Let λ be the output of the algorithms in sections 4.2.1, 4.2.2, 4.2.3. From the algorithm point of view, it is the subgroup of all matrices for which λ is a square. Now once the λ is a square, and we can efficiently compute the square root, we can write this matrix down as a product of elementary matrices, and it is easy to implement the MOR cryptosystem. It is well known that if $p \equiv 3 \pmod 4$, then it is easy to compute the square root. Only, for this reason, we have chosen p such that $p \equiv 3 \pmod 4$. It is known [57] that the the finite orthogonal group is generated by two elements called Steinberg generators. We use Steinberg generators and present automorphism as an action on these generators. If one has to compute $\phi(g)$, then the word problem must be solved because the automorphisms used for the MOR cryptosystem are presented as action on generators. Thus, we need an efficient algorithm to solve the word problem in Steinberg generators. We use the algorithm developed in chapter 4 to resolve this issue. The basic idea is to write elementary matrices as words in Steinberg generators and then use the Gaussian elimination algorithm to solve the word problem. This approach is constructive and uses the straight-line programs technique [4], which is popular in computational group theory. Before we demonstrate the procedure, let us digress briefly to review the technique of straight-line programs.

Suppose we have a group $G = \langle X \rangle$. By definition, any element g of G can be expressed as a word in X, $g = x_1 x_2 \cdots x_k$, for $x_i \in X$. The words in X that define elements of G may be very long so in practice words are normally stored as straight-line programs. A straight-line program from X to g is a

sequence of expressions $z = (z_1, \ldots, z_k)$, where each z_i is either an element of a set X or is an expression of the form (z_j, z_k) or $(z_j, -1)$ for $j, k < i$. Let's make the notion of straight-line program precise, let $z = (z_1, \ldots, z_k)$ we define the group element $Eval(z)$ as : for $i, j < k$

$$Eval(z) := \begin{cases} x & \text{if } z_k = x, \\ Eval((z_1, \ldots, z_i))^{-1} & \text{if } z_k = (z_j, -1), \\ Eval((z_1, \ldots, z_i))^{-1} * Eval((z_1, \ldots, z_j))^{-1} & \text{if } z_k = (z_i, z_j). \end{cases}$$

For g we say z is a straight-line program(SLP) from X to g if $Eval(z) = g$. For example, straight-line program for $y_1^{-1}y_2y_1^{-1}y_2$ is $(z_1, z_2, z_3, z_4, z_5)$, where z_1 represent y_1, $z_2 = (z_1, -1)$, z_3 represents y_2, $z_4 = (z_2, z_3)$, and $z_5 = (z_4, z_4)$.

As we can see that straight-line programs are equivalent to words in X. However in practice, evaluating a straight-line program is much faster than evaluating the corresponding words since we evaluate repeated sub-words once. In above example, if we have evaluated $y_1^{-1}y_2y_1^{-1}y_2$ by simply calculating each part in succession it would require two inversions and three multiplications of group elements. However, evaluating the straight-line program requires only one inversion and two multiplications of group elements. For more detailed and compete discussion on straight-line programs we refer to [42] and [21, Section 2.2.5]

In a context of our scheme, we can use these straight-line programs as follows: Suppose we want to compute $x_{i,j}(t)$ for some $t \in \mathbb{F}_q$. We have loaded matrices $w^{i-1}x_{1,2}(\cdot)w^{i-1}$ in a memory in such a way that this formula takes as input t and put it in the $(1, 2)$ position of the matrix $x_{1,2}(\cdot)$ and do the matrix multiplication. This is one straight line program. Since these programs are loaded in memory, computation is much faster. We have built a series of these straight line programs, where one straight line program can use other straight line programs and have written down the length of these programs. The length is nothing but the number of matrices in the formula.

To avoid the complications in the straight line programs using Steinberg generators and their lengths computations, we describe these procedures for $O^+(2l,p), O^-(2l,p)$ and $O(2l+1,p)$ separately. Throughout this section, we assume that $p \equiv 3 \pmod 4$.

5.2.1 Orthogonal group $O^+(2l,p)$

It is known [57] that the group $O^+(2l,p)$ is generated by two elements x and w which we describe below. Recall that to solve the word problem in these generators, we need to know how to go back and forth between these two generating sets - Steinberg generators and elementary matrices defined earlier (chapter 4). We use the following generators which we refer as Steinberg generators.

$$x = x_{1,2}(1), \tag{5.1}$$

$$w = \left(\begin{array}{ccc|ccc} 0 & \cdots & 0 & 0 & \cdots & 1 \\ -1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \cdots & -1 & 0 & 0 & \cdots & 0 \\ \hline 0 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & -1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & -1 & 0 \end{array}\right). \tag{5.2}$$

Note that x itself is an elementary matrix, so we just need to write w as a word in elementary matrices. To write w as a product of elementary matrices is easy, just put this generator through our Gaussian elimination algorithm(section 4.1.3). Here we demonstrate the other way round, that is, how to write elementary matrices as a product of x and w. We provide a constructive way to go from Steinberg generators to elementary matrices. In the context of MOR cryptosystem, we do following computations using SLP technique mentioned above. In what follows, we denote the length of SLP's

by $L(d,i)$, where $d = j - i$ and $1 \leq i < j \leq l$.

$$\begin{aligned} d = 1, \quad x_{i,j}(t) &= w^{i-1} x_{1,2}(t) w^{-(i-1)}, \\ d = 2, \quad x_{i,j}(t) &= [x_{i,j-1}(t), x_{j-1,j}(1)], \\ d = 3, \quad x_{i,j}(t) &= [x_{i,j-1}(t), x_{j-1,j}(1)], \\ \vdots \qquad \vdots &\qquad \vdots \\ d = l-1, \quad x_{i,j}(t) &= [x_{i,j-1}(t), x_{j-1,j}(1)]. \end{aligned}$$

Here

$$L(d,i) = \begin{cases} 2i - 1 & \text{for } d = 1, \\ 2L(d-1) + 4(i+d) - 6 & \text{for } d = 2, 3, \ldots, l-1. \end{cases}$$

Now, $w^l = (-1)^{l-1} \begin{pmatrix} 0 & -I_l \\ -I_l & 0 \end{pmatrix}$ and $x_{j,i}(t) = w^l x_{i,j}(-t) w^{-l}$, so length of this SLP is $L(d,i) + 2l$. Hence we get all $x_{i,j}(t)$, for $1 \leq i \neq j \leq l$. Number of SLP $= (l-1) + 1 = l$. Next observe that, Thus we can generate

Table 5.1: SLP lengths

Elements	Indices	Equation	Length	
$x_{1,-l}(t)$		$w x_{l-1,l}(t) w^{-1}$	$2l - 1$	
$x_{1,-i}(t)$	$2 \leq i \leq l-1$	$[x_{i,l}(t), x_{1,-l}(1)]$	$2(L(l-i,i) + 2l - 1)$	
$x_{i,-j}(t)$	$2 \leq i \leq l-1$	$[x_{i,1}(t), x_{1,-j}(1)]$	$2L(i-1,1)+$	
			$2(2L(l-j,j) + 6l - 2)$	$j \neq l$
	$(i+1 \leq j \leq l)$		$2(L(i-1,1) + 4l - 1)$	j=l

all $x_{i,-j}(t)$, for $i < j$. Note that we can generate $x_{-i,j}(t)$ by computing $w^l x_{i,-j}(t) w^{-l} = x_{-i,j}(t)$, and the total number of SLP's required to generate it is equal to $l + 4$. Recall that the elementary matrices $x_{i,j}(t)$ generate $\Omega(2l,p)$, the commutator subgroup of $O(2l,p)$ of index 4. Hence we generate $\Omega(2l,p)$, using only two generators x and w. Observe that using the relation $w = w_{1,2}(1) w_{2,3}(1) \cdots w_{l-1,l}(1) w_l$, where $w_{i,j}(t) = x_{i,j}(t) x_{j,i}(-t^{-1}) x_{i,j}(t)$, we can generate w_l. Now we know $w_{l-1} = w_l w_{l,l-1}(1) w_{l-1,-l}(1)$, so we generate w_{l-1}. Now by induction, we generate $w_i = w_{i+1} w_{i+1,i}(1) w_{i,-(i+1)}(1)$ for $i = l-1, \ldots, 1$. Here $w_{i,-j}(t) = x_{i,-j}(t) x_{-i,j}(t^{-1}) x_{i,-j}(t)$, for $i < j$. Hence we

generate all the elementary matrices defined earlier (Table 4.3) using only two generators x and w. So we generate a new subgroup $W\Omega(2l,p)$ of $O(2l,p)$, which is indeed a normal subgroup of $O(2l,p)$. In our algorithm, output matrix is $d(\lambda) = \text{diag}\,(1,1,\cdots,\lambda,1,1,\ldots,\lambda^{-1})$. If $\lambda \in F_p^{\times 2}$, say $\lambda \equiv t^2 \pmod p$, then $t \equiv \lambda^{\frac{p+1}{4}} \pmod p$, since $p \equiv 3 \pmod 4$.

$$\begin{aligned}\text{Then}\quad d(\lambda) &= \text{diag}\,(1,\ldots,t^2,1,\ldots,t^{-2}) \\ &= w_{l-1,l}(1)\text{diag}\,(1,\ldots,t^2,1,1,\ldots,t^{-2},1)w_{l-1,l}(-1) \\ &= w_{l-1,l}(1)w_{l-1,l}(t)w_{l-1,l}(-1)w_{l-1,-l}(t)w_{l-1,-l}(-1)w_{l-1,l}(-1).\end{aligned}$$

Hence we generate $W\Omega(2l,p)$ using only two generators x and w.

Remark 5.5. Let $d(\zeta) = \text{diag}\,(1,\ldots,1,\zeta,1,\ldots,1,\zeta^{-1})$, where ζ is non-square in $F_p^{\times}$. Then the group $\langle W\Omega, d(\zeta)\rangle$ is the orthogonal group.

5.2.2 Orthogonal group $O(2l+1,p)$

We use the following generators which we refer as Steinberg generators.

$$x = x_{0,1}(1), \tag{5.3}$$

$$w = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -I_{2l-1} & 0 \end{pmatrix}, \tag{5.4}$$

$$w_l = I - e_{l,l} - e_{-l,-l} + e_{l,-l} + e_{-l,l}. \tag{5.5}$$

It is known [57] that the group $O(2l+1,p)$ is generated by these elements. However, in context of the MOR cryptosystem, we need to know how to go back and forth between these two generating sets – Steinberg generators and elementary matrices defined earlier (Table 4.1). The procedure is almost similar to the case of $O^+(2l,p)$. Again note that $x = x_{0,1}$ is an elementary matrix. Thus we just need to write w as a product of elementary matrices. However, computing w is fairly easy, just put this generator through our Gaussian elimination algorithm (section 4.1.2).

Here we demonstrate the other way round, that is, how to write ele-

mentary matrices as a product of w and x. First we compute, $x_{0,i}(t) = w^{i-1}x_{0,1}(1)w^{-(i-1)}$ which of length $2i-1$ for $1 \leq i \leq l$. Now we compute $x_{i,0}(t)$ using the relation $x_{i,0}(t) = w^l x_{0,i}(-t)w^{-l}$ for $1 \leq i \leq l$, where $w^l = (-1)^l \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & I_l \\ 0 & I_l & 0 \end{pmatrix}$ and length of this SLP is $2l + 2i - 1$. Thus, we get $x_{i,0}(t)$ and $x_{0,i}(t)$ for $i = 1, 2, \ldots, l$. Next, we compute $x_{1,2}(t)$ using the commutator formula $x_{1,2}(t) = [x_{1,0}(\frac{t}{2}), x_{0,2}(1)]$ and length of this SLP is $4l + 8$. In what follows, we denote the length of SLP's by $L(d, i)$, where $d = j - i$ and $1 \leq i < j \leq l$.

$$\begin{aligned}
d = 1, \quad x_{i,j}(t) &= \quad w^{i-1}x_{1,2}(t)w^{-(i-1)}, \\
d = 2, \quad x_{i,j}(t) &= \quad [x_{i,j-1}(t), x_{j-1,j}(1)], \\
d = 3, \quad x_{i,j}(t) &= \quad [x_{i,j-1}(t), x_{j-1,j}(1)], \\
\vdots \qquad \vdots &\qquad \vdots \\
d = l-1, \quad x_{i,j}(t) &= \quad [x_{i,j-1}(t), x_{j-1,j}(1)].
\end{aligned}$$

Here

$$L(d,i) = \begin{cases} 2i + 4l + 6 & \text{for } d = 1, \\ 2L(d-1,i) + 4(i + d + 2l + 2) & \text{for } d = 2, 3, \ldots, l-1. \end{cases}$$

As $x_{j,i}(t) = w^l x_{i,j}(-t)w^{-l}$, so length of this SLP is $L(d,i) + 2l$. Hence we get all $x_{i,j}(t)$ for $1 \leq i \neq j \leq l$. Number of SLP $= 3 + (l-1) + 1 = l + 3$.

Next we compute the remaining elementary matrices using the commutator formula and are listed in the following table. We compute exact SLP lengths required to generate elementary matrices. Thus we have generated all $x_{i,-j}(t)$ for $i < j$. Now using the formula $w^l x_{i,-j}(t)w^{-l} = x_{-i,j}(t)$, we get $x_{-i,j}(t)$ and total number of SLP's required is $l + 7$. It is shown in Ree [48] that the elementary matrices $x_{i,j}(t)$ generate $\Omega(2l+1, p)$, the commutator subgroup of $O(2l+1, p)$ of index 4. So we generate $\Omega(2l+1, p)$, using only two

Table 5.2: SLP lengths

Elements	Indices	Equation (SLP)	Length	
$x_{1,-l}(t)$		$wx_{l-1,l}(t)w^{-1}$	$6l+6$	
$x_{1,-i}(t)$	$2 \le i \le l-1$	$[x_{i,l}(t), x_{1,-l}(1)]$	$24l+20$	$i = l-1$
			$2L(l-i,i)+12(l+1)$	$i \neq l-1$
$x_{i,-j}(t)$	$2 \le i \le l-1$	$[x_{i,1}(t), x_{1,-j}(1)]$	$2L(i-1,1)+4(7l+6)$	
			$+4L(l-j-d, j-d)$	$j < l-1$
	$(i+1 \le j \le l)$		$2L(i-1,1)+4(7l+5)$	$j = l-1$
			$2L(i-1,1)+10l+6$	$j = l$

generators x and w. Now we know $w_{l-1} = w_l w_{l,l-1}(1) w_{l-1,-l}(1)$, so we generate w_{l-1}. Hence by inductively we can generate $w_i = w_{i+1} w_{i+1,i}(1) w_{i,-(i+1)}(1)$ for $i = l-1, \ldots, 1$. Here $w_{i,j}(t) = x_{i,j}(t) x_{j,i}(-t^{-1}) x_{i,j}(t)$ for $i \neq j$ and $w_{i,-j}(t) = x_{i,-j}(t) x_{-i,j}(t^{-1}) x_{i,-j}(t)$ for $i < j$. Hence we generate all the elementary matrices defined earlier (Table 4.1) using only two generators x and w and an extra element w_l. Hence we generate a new subgroup $W\Omega(2l+1,p)$ of the orthogonal group $O(2l+1,p)$, containing Ω, which is indeed a normal subgroup of $O(2l+1,p)$. In our algorithm, the output matrix is $d(\lambda) = \mathrm{diag}\,(1, 1, \cdots, \lambda, 1, \ldots, \lambda^{-1})$.

If $\lambda \in F_p^{\times 2}$, say $\lambda \equiv t^2 \pmod p$, here $t \equiv \lambda^{\frac{p+1}{4}} \pmod p$, since $p \equiv 3 \pmod 4$.

$$\begin{aligned} \text{Then} \quad d(\lambda) &= \mathrm{diag}\,(1, 1, \ldots, t^2, 1, \ldots, t^{-2}) \\ &= w_{l-1,l}(1)\mathrm{diag}\,(1, 1, \ldots, t^2, 1, 1, \ldots, t^{-2}, 1) w_{l-1,l}(-1) \\ &= w_{l-1,l}(1) w_{l-1,l}(t) w_{l-1,l}(-1) w_{l-1,-l}(t) w_{l-1,-l}(-1) w_{l-1,l}(-1). \end{aligned}$$

Hence we generate $W\Omega(2l+1,p)$ using x, w and w_l.

Remark 5.6. Let $d(\zeta) = \mathrm{diag}\,(1, 1, \ldots, 1, \zeta, 1, \ldots, 1, \zeta^{-1})$, where ζ is non-square in $F_p^{\times}$. Then the group $\langle W\Omega, d(\zeta)\rangle$ is the orthogonal group.

5.2.3 Orthogonal group $O^-(2l, p)$

We use the following generators (see table 4.5) which we refer as Steinberg generators.

$$\begin{aligned} x &= x_{1,2}(1), \\ x' &= x_{-1,2}(1), \\ w &= \begin{pmatrix} -I_2 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -I_{2l-3} & 0 \end{pmatrix}, \\ w_l &= I - e_{l,l} - e_{-l,-l} - e_{l,-l} - e_{-l,l}, \\ & x_1(t,s), \text{ where } t, s \in \mathbb{F}_p \text{ and } x_2. \end{aligned}$$

In context of the MOR cryptosystem, we need to know how to go back and forth between these generators and elementary matrices defined earlier (Table 4.5). The procedure is almost similar to the case of $O^+(2l, p)$. Again, note that $x = x_{1,2}$, $x' = x_{-1,2}$, $x_1(t, s)$ and x_2 are elementary matrices. Thus, we just need to write w as a product of elementary matrices. However, computing w is fairly easy, just put this generator through our Gaussian elimination algorithm (section 4.1.2). Here we demonstrate the other way round, that is, how to write elementary matrices as a product of w, x, x'. First we compute, $x_{1,i}(t) = w^{i-1}x_{1,2}(1)w^{-(i-1)}$ which of length $2i - 1$ for $2 \leq i \leq l$. Now we compute $x_{i,1}(t)$ using the relation $x_{i,1}(t) = w^{l-1}x_{1,i}(-t)w^{-(l-1)}$ for $2 \leq i \leq l$, where $w^{l-1} = (-1)^{l-1}\begin{pmatrix} I_2 & 0 & 0 \\ 0 & 0 & I_{l-1} \\ 0 & I_{l-1} & 0 \end{pmatrix}$ and length of this SLP is $2(l-1) + 2i - 1$. Thus, we get $x_{i,1}(t)$ and $x_{1,i}(t)$, for $i = 2, \ldots, l$. Similarly we compute $x_{i,-1}(t)$ and $x_{-1,i}(t)$ using the relations $x_{-1,i}(t) = w^{i-1}x_{-1,2}(1)w^{-(i-1)}$, $x_{i,-1}(t) = w^{l-1}x_{-1,i}(-t)w^{-(l-1)}$ for $2 \leq i \leq l$ and has SLP lengths $2i - 1$, $2(l-1) + 2i - 1$ respectively. Next, we compute $x_{2,3}(t)$ using the commutator formula $x_{2,3}(t) = [x_{2,1}(\frac{t}{2}), x_{1,3}(1)]$ and length of

this SLP is $4(l-1)+8$. In what follows, we denote the length of SLP's by $L(d,i)$, where $d=j-i$ and $2 \leq i < j \leq l$.

$$\begin{aligned} d=1, \quad x_{i,j}(t) &= w^{i-1}x_{2,3}(t)w^{-(i-1)}, \\ d=2, \quad x_{i,j}(t) &= [x_{i,j-1}(t), x_{j-1,j}(1)], \\ d=3, \quad x_{i,j}(t) &= [x_{i,j-1}(t), x_{j-1,j}(1)], \\ \vdots \qquad \vdots &\qquad \vdots \\ d=l-1, \quad x_{i,j}(t) &= [x_{i,j-1}(t), x_{j-1,j}(1)]. \end{aligned}$$

Here

$$L(d,i) = \begin{cases} 2i+4(l-1)+6 & \text{for } d=1, \\ 2L(d-1,i)+4(i+d+2(l-1)+2) & \text{for } d=2,3,\ldots,l-2. \end{cases}$$

As $x_{j,i}(t) = w^{l-1}x_{i,j}(-t)w^{-(l-1)}$, so length of this SLP is $L(d,i)+2(l-1)$. Hence, we get all $x_{i,j}(t)$ for $2 \leq i \neq j \leq l$ and number of SLPs is $l+2$. Next, we compute the remaining elementary matrices using the commutator formula and are listed in the table 5.3, let $r=l-1$.

Table 5.3: SLP lengths

Elements	Indices	Equation (SLP)	Length	
$x_{1,-l}(t)$		$wx_{l-1,l}(t)w^{-1}$	$6(l-1)+6$	
$x_{1,-i}(t)$	$2 \leq i \leq l-1$	$[x_{i,l}(t), x_{1,-l}(1)]$	$24(l-1)+20$	$i=l-1$
			$2L(r-i,i)+12(r+1)$	$i \neq l-1$
$x_{i,-j}(t)$	$2 \leq i \leq r-1$	$[x_{i,1}(t), x_{1,-j}(1)]$	$2L(i-1,1)+4(7r+6)$	
			$+4L(r-j-d, j-d)$	$j<l-1$
	$(i+1 \leq j \leq l)$		$2L(i-1,1)+4(7r+5)$	$j=l-1$
			$2L(i-1,1)+10r+6$	$j=l$

Thus, we have generated all $x_{i,-j}(t)$ for $i<j$. Now, using the formula $w^l x_{i,-j}(t)w^{-l} = x_{-i,j}(t)$, we get $x_{-i,j}(t)$ and total number of SLP's required is $l+6$. Now we know $w_{l-1} = w_l w_{l,l-1}(1)w_{l-1,-l}(1)$, so we generate w_{l-1}. Hence by inductively we can generate $w_i = w_{i+1}w_{i+1,i}(1)w_{i,-(i+1)}(1)$, for $i = l-1,\ldots,2$. Here $w_{i,j}(t) = x_{i,j}(t)x_{j,i}(-t^{-1})x_{i,j}(t)$, for $i \neq j$ and $w_{i,-j}(t) = x_{i,-j}(t)x_{-i,j}(t^{-1})x_{i,-j}(t)$, for $i<j$. Hence we generate all the elementary

matrices defined earlier (Table 4.5) using generators x, x', $x_1(t,s)$, x_2, w and an extra element w_l. In our algorithm the output matrix is $d(\lambda) = \text{diag}\,(1,1,1,\ldots,\lambda,1,\ldots,\lambda^{-1})$. If $\lambda \in F_p^{\times 2}$, say $\lambda \equiv t^2 \pmod p$, here $t \equiv \lambda^{\frac{p+1}{4}} \pmod p$, since $p \equiv 3 \pmod 4$.

$$\begin{aligned}\text{Then}\quad d(\lambda) &= \text{diag}\,(1,1,1,\ldots,t^2,1,\ldots,t^{-2}) \\ &= w_{l-1,l}(1)\text{diag}\,(1,1,1,\ldots,t^2,1,1,\ldots,t^{-2},1)w_{l-1,l}(-1) \\ &= w_{l-1,l}(1)w_{l-1,l}(t)w_{l-1,l}(-1)w_{l-1,-l}(t)w_{l-1,-l}(-1)w_{l-1,l}(-1).\end{aligned}$$

Remark 5.7. Let $d(\zeta) = \text{diag}\,(1,1,1,\ldots,\zeta,1,\ldots,\zeta^{-1})$, where ζ is non-square in $F_p^{\times}$. Then as a consequence of our algorithm 4.1.4, we can see that x, x', $x_1(t,s)$, x_2, w and w_l along with $d(\zeta)$ generates the twisted orthogonal group.

5.3 Implementation

In practice, the best public-key cryptosystem is the one that manages to keep a good balance between speed and security. Hence there is always a trade-off between speed and security. The implementation of the MOR cryptosystem we have in mind uses the row-column operations. Let $\langle g_1, g_2, \ldots, g_k\rangle$ be a set of generators for the orthogonal group as described before. Recall that the automorphisms ϕ and ϕ^x used in MOR cryptosystem are presented as action on generators, i.e., we have $\phi(g_i)$ and $\phi^x(g_i)$ as matrices, for $i = 1,2,\ldots,k$. Let $m \in G$ be a message, in order to encrypt this message; we compute ϕ^r. Now the question is how to compute large powers of ϕ effectively. We do that by *square-and-multiply* algorithm [58, Algorithm 5.5]. For this implementation, squaring and multiplying is almost the same. So we will refer to both squaring and multiplication as multiplication. Note that multiplication is composing of automorphisms. First, we write the matrix of $\phi(g_i)$ as a word in generators for each $i = 1,2,\ldots,k$, we can do these computations in parallel. Each thread computes $\phi^r(g_i)$ for $i = 1,2,\ldots,k$. Thus we compute images of ϕ^r by replacing all instances of $\phi^r(g_i)$ computed

using the parallel threads. This can be done very fast. However, the length of the replaced word can become very large. The obvious question is, how soon are we going to write this word as a matrix. This is a difficult question to answer at this stage and depends on available computational resources.

Once we decide how often we change back to matrices, how are we going to change back to matrices? There can be a fairly easy *time-memory* trade-offs. Write all words up to a fixed length and the corresponding matrix as a pre-computed table and use this table to compute the matrices. Once we have matrices, we can multiply them together to generate the final output. If writing all words is impossible, due to resource constraint, write some of it in a table. There are also many obvious relations among the generators of these groups. One can just store and use them. The best strategy for an efficient implementation is yet to be determined. It is clear now that there are many interesting and novel choices.

The benefits of this MOR cryptosystem are:

- This can be implemented in parallel easily.

- This implementation doesn't depend on the size of the characteristic of the field. This is an important property in light of Joux's recent improvement of the index-calculus attacks [6].

For parameters and complexity analysis of this cryptosystem, we refer [29, Section 8]. Assume that we take a prime of a size 2^{160}, and we are using two generators presentation of ϕ for the even-orthogonal group. Then the security is the discrete logarithm problem in $\mathbb{F}_{p^{d^2-1}}$. Now if we take $d = 4$, then the security is at most $\mathbb{F}_{2^{2560}}$. Our key size is about 8000 bits. Comparing with Monico [39, section 7], where he says an ElGamal will have about 6080 bits, our system is quite comparable. Moreover, the MOR cryptosystem is better suited to handle large primes and can be easily parallelized.

5.3.1 Further Research

We conclude this thesis with the following open direction for further research. What is the most efficient strategy to implement the MOR cryptosystem on orthogonal groups that we described earlier?

Appendices

Appendix A
Spinor norm

In this chapter, we explicitly compute the Spinor norm of the matrices in orthogonal group $O^-(d, K)$, for char(K) odd using the Gaussian elimination algorithms presented in section 4.2.3. For the spinor norm computation of other orthogonal groups, we refer to [8, section 5].

Spinor norm plays very important role in the study of orthogonal groups. The classical way to define spinor norm is via Clifford algebras [17, Chapters 8 and 9]. However, in practice, it is difficult to use that definition to compute the spinor norm. Wall [61], Zassenhaus [62] and Hahn [18] developed a theory to compute the spinor norm. For our exposition, we follow [59, Chapter 11].

Murray and Roney-Dougal [41] used the formula of Hahn [18, Proposition 2.1] to compute spinor norm. However, their algorithm works only for K finite. Our algorithm developed in chapter 4 works for infinite fields and outputs the spinor norm quickly. It is well-known that by Cartan-Dieudonne theorem [17, Theorem 6.6] every element of a orthogonal group can be written as a product of at most d reflections. Let Q be a quadratic form.

Definition A.0.1. (Spinor norm) The spinor norm is a group homomorphism $\Theta : O(d, K) \to K^\times/K^{\times 2}$ defined by $\Theta(g) = \prod_{i=1}^{d} Q(u_i)$, where $g = \rho_{u_1}\rho_{u_2}\cdots\rho_{u_d}$ with $Q(u_i) \neq 0$, for all $1 \leq i \leq d$.

Let $g \in O(d, K)$, denote $g' = I - g$, $V_g = g'(V)$ and $V^g = \text{Ker}(g')$. Using

the bilinear form β, we define Wall's bilinear form $[\ ,\ ]_g$ on V_g as follows:

$$[u, v]_g = \beta(u, y), \text{ where } v = g'(y).$$

This bilinear form satisfies following properties:

1. $[u, v]_g + [v, u]_g = \beta(u, v)$ and $[u, u]_g = Q(u)$ for all $u, v \in V_g$;
2. g is an isometry on V_g with respect to $[\ ,\ ]_g$;
3. $[v, u]_g = -[u, gv]$ for all $u, v \in V_g$;
4. $[\ ,\ ]_g$ is non-degenerate.

Then the spinor norm of $g \in O(d, K)$ is defined to be

$$\Theta(g) = \overline{disc(V_g, [\ ,\ ]_g)} \qquad \text{if} \quad g \neq I$$

extended to I by defining $\Theta(I) = \bar{1}$. This is another equivalent definition of spinor norm. An element g is called regular if V_g is a non-degenerate subspace of V with respect to the form β. As a consequence of Hahn [18, Proposition 2.1] formula we get the following proposition.

Proposition A.0.1. 1. For a reflection ρ_v, $\Theta(\rho_v) = \overline{Q(v)}$.

2. $\Theta(-1) = \overline{disc(V, \beta)}$.
3. For a unipotent element g the spinor norm is trivial, i.e., $\Theta(g) = \bar{1}$.

We use this proposition and algorithm 4.2.3 to compute the spinor norm of elements of $O^-(d, K)$. First we observe the following

Lemma A.0.2. The spinor norm of elementary matrices [Table 4.5] in $O^-(d, K)$ are:

1. $\Theta(x_{i,j}(t)) = \Theta(x_{-i,j}(t)) = \Theta(x_{i,-j}(t)) = \bar{1}$.
2. $\Theta(w_i) = \bar{1}$.
3. $\Theta(\text{diag}(1, 1, 1, \ldots, 1, \lambda, 1, \ldots, 1, \lambda^{-1})) = \bar{\lambda}$,
   $\Theta(\text{diag}(1, -1, 1, \ldots, 1, \lambda, 1, \ldots, 1, \lambda^{-1})) = \overline{\epsilon\lambda}$,
   $\Theta(\text{diag}(-1, 1, 1, \ldots, 1, \lambda, 1, \ldots, 1, \lambda^{-1})) = \bar{\lambda}$,
   $\Theta(\text{diag}(-1, -1, 1, \ldots, 1, \lambda, 1, \ldots, 1, \lambda^{-1})) = \overline{\epsilon\lambda}$.

4. $\Theta(x_1(t,s)) = \overline{2(1-t)}$.
5. $\Theta(x_2) = \bar{\epsilon}$.

Proof: The first one follows from previous proposition as all elementary matrices are unipotent. The element $w_i = \rho_{e_i+e_{-i}}$ is a reflection thus $\Theta(w_i) = \overline{Q(e_i+e_{-i})} = \bar{1}$. For the third part we note that $\text{diag}(1,1,1,\ldots,1,\lambda,1,\ldots,1,\lambda^{-1}) = \rho_{e_l+e_{-l}}\rho_{e_l+\lambda e_{-l}}$ and hence,

$$\Theta(\text{diag}(1,1,1,\ldots,1,\lambda,1,\ldots,1,\lambda^{-1})) = \Theta(\rho_{e_l+e_{-l}})\Theta(\rho_{e_l+\lambda e_{-l}})$$
$$\Theta(\text{diag}(1,1,1,\ldots,1,\lambda,1,\ldots,1,\lambda^{-1})) = \overline{Q(e_l+\lambda e_{-l})}$$
$$\Theta(\text{diag}(1,1,1,\ldots,1,\lambda,1,\ldots,1,\lambda^{-1})) = \bar{\lambda}.$$

Observe that

$$\text{diag}(1,-1,1,\ldots,1,\lambda,1,\ldots,1,\lambda^{-1}) = \text{diag}(1,-1,1,\ldots,1,\lambda,1,\ldots,1,\lambda^{-1})\rho_{e_{-1}}$$
$$\text{diag}(-1,1,1,\ldots,1,\lambda,1,\ldots,1,\lambda^{-1}) = \text{diag}(1,-1,1,\ldots,1,\lambda,1,\ldots,1,\lambda^{-1})\rho_{e_1}$$

implies that

$$\Theta(\text{diag}(1,-1,1,\ldots,1,\lambda,1,\ldots,1,\lambda^{-1})) = \overline{\epsilon\lambda},$$
$$\Theta(\text{diag}(-1,1,1,\ldots,1,\lambda,1,\ldots,1,\lambda^{-1})) = \overline{\lambda}.$$

Similarly we can see that

$$\text{diag}(-1,-1,1,\ldots,1,\lambda,1,\ldots,1,\lambda^{-1})) = \text{diag}(1,-1,1,\ldots,1,\lambda,1,\ldots,1,\lambda^{-1})\rho_{e_1}$$

implies that $\Theta(\text{diag}(-1,-1,1,\ldots,1,\lambda,1,\ldots,1,\lambda^{-1})) = \overline{\epsilon\lambda}$. For the fourth part we observe that $x_1(t,s) = \rho_{(t-1)e_1+se_{-1}}$ and hence

$$\begin{aligned}\Theta(x_1(t,s)) &= \overline{Q(\rho_{(t-1)e_1+se_{-1}})} \\ &= (t-1)^2 + \epsilon s^2 \\ &= 2(1-t) \qquad \text{as} \quad t^2+\epsilon s^2 = 1.\end{aligned}$$

Note that $x_2 = \rho_{e_{-1}}$ and $\Theta(\rho_{e_{-1}}) = \bar{\epsilon}$ implies that $\Theta(x_2) = \bar{\epsilon}$.

Remark A.1. As we seen that our algorithm enables to write every element g of $O^-(d,K)$ as a product of elementary matrices and a diagonal matrix hence we can find the spinor norm of g.

Bibliography

[1] National security agency, cryptography today, august 2015 archived on 23 november 2015, tinyurl.com/suiteb.

[2] Miklós Ajtai. Generating hard instances of lattice problems. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 99–108. ACM, 1996.

[3] A. A. Albert and John Thompson. Two-element generation of the projective unimodular group. *Illinois Journal of Mathematics*, 3:421–439, 1959.

[4] László Babai and Endre Szemerédi. On the complexity of matrix group problems I. In *Foundations of Computer Science, 1984. 25th Annual Symposium on*, pages 229–240. IEEE, 1984.

[5] Ramachandran Balasubramanian and Neal Koblitz. The improbability that an elliptic curve has subexponential discrete log problem under the Menezes-Okamoto-Vanstone algorithm. *Journal of cryptology*, 11(2):141–145, 1998.

[6] Razvan Barbulescu, Pierrick Gaudry, Antoine Joux, and Emmanuel Thomé. A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 1–16. Springer, 2014.

[7] Daniel J Bernstein, Johannes Buchmann, and Erik Dahmen. *Post-quantum cryptography*. Springer Science & Business Media, 2009.

[8] Sushil Bhunia, Ayan Mahalanobis, and Anupam Singh. Gaussian elimination in symplectic and split orthogonal groups. *arXiv preprint arXiv:1504.03794*, 2015.

[9] Peter A Brooksbank. Constructive recognition of classical groups in their natural representation. *Journal of Symbolic Computation*, 35(2):195–239, 2003.

[10] Johannes Buchmann, Erik Dahmen, and Michael Szydlo. Hash-based digital signature schemes. In *Post-Quantum Cryptography*, pages 35–93. Springer, 2009.

[11] Roger W Carter. *Simple groups of Lie type*, volume 22. John Wiley & Sons, 1989.

[12] Joan-Josep Climent, Pedro R Navarro, and Leandro Tortosa. An extension of the noncommutative Bergman's ring with a large number of noninvertible elements. *Applicable Algebra in Engineering, Communication and Computing*, 25(5):347–361, 2014.

[13] Elliot Mark Costi. *Constructive membership testing in classical groups*. PhD thesis, Queen Mary, University of London, 2009.

[14] Jean Dieudonné, Lo-keng Hua, and Luogeng Hua. *On the automorphisms of the classical groups; J. Dieudonné Supplement by LK Hua*. Number 2. American Mathematical Soc., 1951.

[15] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE transactions on Information Theory*, 22(6):644–654, 1976.

[16] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE transactions on information theory*, 31(4):469–472, 1985.

[17] Larry C Grove. *Classical groups and geometric algebra*, volume 39. American Mathematical Soc., 2002.

[18] Alexander J Hahn. Unipotent elements and the spinor norms of wall and zassenhaus. *Archiv der Mathematik*, 32(1):114–122, 1979.

[19] Jeffrey Hoffstein, Jill Pipher, and Joseph H Silverman. *An introduction to mathematical cryptography*, volume 1. Springer, 2008.

[20] Antoine Joux. A new index calculus algorithm with complexity L(1/4+ o(1)) in small characteristic. In *International Conference on Selected Areas in Cryptography*, pages 355–379. Springer, 2013.

[21] William M Kantor and Ákos Seress. *Black box classical groups*, volume 708. American Mathematical Soc., 2001.

[22] Aviad Kipnis, Jacques Patarin, and Louis Goubin. Unbalanced oil and vinegar signature schemes. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 206–222. Springer, 1999.

[23] MA Knus, A Merkurjev, M Rost, and JP Tignol. The book of involutions.(with a preface in French by J. Tits.) American Mathematical Society Colloquium Publications 44. *American Mathematical Society, Providence, RI*, 1998.

[24] Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of computation*, 48(177):203–209, 1987.

[25] Neal Koblitz, Alfred Menezes, and Scott Vanstone. The state of elliptic curve cryptography. In *Towards a quarter-century of public key cryptography*, pages 103–123. Springer, 2000.

[26] In-Sok Lee, Woo-Hwan Kim, Daesung Kwon, Sangil Nahm, Nam-Seok Kwak, and Yoo-Jin Baek. On the security of Mor public key cryptosystem. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 387–400. Springer, 2004.

[27] Ayan Mahalanobis. A simple generalization of the elgamal cryptosystem to non-abelian groups. *Communications in Algebra*, 36(10):3878–3889, 2008.

[28] Ayan Mahalanobis. A simple generalization of the elgamal cryptosystem to non-abelian groups II. *Communications in Algebra*, 40(9):3583–3596, 2012.

[29] Ayan Mahalanobis. The MOR cryptosystem and finite p-groups. *Algorithmic Problems of Group Theory, Their Complexity, and Applications to Cryptography*, 633:81, 2015.

[30] Tsutomu Matsumoto and Hideki Imai. Public quadratic polynomial-tuples for efficient signature-verification and message-encryption. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 419–453. Springer, 1988.

[31] Gérard Maze, Chris Monico, and Joachim Rosenthal. Public key cryptography based on semigroup actions. *Advances in Mathematics of Communications*, pages 489–507, 2007.

[32] Gérard Maze, Chris Monico, Joachim Rosenthal, and JJ Climent. Public key cryptography based on simple modules over simple rings. In

Proceedings of the 15-th International Symposium on the Mathematical Theory of Networks and Systems, University of Notre Dame, 2002.

[33] Robert J McEliece. A public-key cryptosystem based on algebraic coding theory. *Coding Thv*, pages 114–116, 1978.

[34] Alfred J Menezes, Tatsuaki Okamoto, and Scott A Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Transactions on information Theory*, 39(5):1639–1646, 1993.

[35] Alfred J Menezes and Yi-Hong Wu. The discrete logarithm problem in GL (n, q). *Ars Combinatoria*, 47:23–32, 1997.

[36] Ralph C Merkle. Secure communications over insecure channels. *Communications of the ACM*, 21(4):294–299, 1978.

[37] Daniele Micciancio and Shafi Goldwasser. *Complexity of lattice problems: a cryptographic perspective*, volume 671. Springer Science & Business Media, 2012.

[38] Victor S Miller. Use of elliptic curves in cryptography. In *Conference on the Theory and Application of Cryptographic Techniques*, pages 417–426. Springer, 1985.

[39] Chris Monico. Cryptanalysis of a matrix-based MOR system. *Communications in Algebra*, 44(1):218–227, 2016.

[40] Christopher J Monico. *Semirings and semigroup actions in public-key cryptography*. PhD thesis, University of Notre Dame, 2002.

[41] Scott H Murray and Colva M Roney-Dougal. Constructive homomorphisms for classical groups. *Journal of Symbolic Computation*, 46(4):371–384, 2011.

[42] Alice C Niemeyer, Tomasz Popiel, and Cheryl E Praeger. Straight-line programs with memory and matrix Bruhat decomposition. *arXiv preprint arXiv:1305.5617*, 2013.

[43] H Ong and Claus-Peter Schnorr. Signatures through approximate representations by quadratic forms. In *Advances in Cryptology*, pages 117–131. Springer, 1984.

[44] Seong-Hun Paeng. On the security of cryptosystem using automorphism groups. *Information Processing Letters*, 88(6):293–298, 2003.

[45] Seong-Hun Paeng, Kil-Chan Ha, Jae Heon Kim, Seongtaek Chee, and Choonsik Park. New public key cryptosystem using finite non abelian groups. In *Annual International Cryptology Conference*, pages 470–485. Springer, 2001.

[46] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 333–342. ACM, 2009.

[47] Stephen Pohlig and Martin Hellman. An improved algorithm for computing logarithms over GF(p) and its cryptographic significance (corresp.). *IEEE Transactions on information Theory*, 24(1):106–110, 1978.

[48] Rimhak Ree. On some simple groups defined by C. Chevalley. *Transactions of the American Mathematical Society*, 84(2):392–400, 1957.

[49] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):34, 2009.

[50] Martin Rötteler. Quantum algorithms: A survey of some recent results. *Informatik-Forschung und Entwicklung*, 21(1-2):3–20, 2006.

[51] Oliver Schirokauer, Damian Weber, and Thomas Denny. Discrete logarithms: the effectiveness of the index calculus method. In *International Algorithmic Number Theory Symposium*, pages 337–361. Springer, 1996.

[52] Peter W Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on*, pages 124–134. IEEE, 1994.

[53] Vladimir Shpilrain and Gabriel Zapata. Combinatorial group theory and public key cryptography. *Applicable Algebra in Engineering, Communication and Computing*, 17(3-4):291–302, 2006.

[54] Joseph H Silverman and Joe Suzuki. Elliptic curve discrete logarithms and the index calculus. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 110–125. Springer, 1998.

[55] Robert Steinberg. Variations on a theme of Chevalley. *Pacific J. Math*, 9(19591):875 – 891, 1959.

[56] Robert Steinberg. Automorphisms of finite linear groups. *Canad. J. Math*, 12(4):606–616, 1960.

[57] Robert Steinberg. Generators for simple groups. *Canad. J. Math*, 14:277–283, 1962.

[58] Douglas R Stinson. *Cryptography: theory and practice*. CRC press, 2005.

[59] Donald E Taylor. *The geometry of the classical groups*, volume 9. Heldermann Verlag, 1992.

[60] Christian Tobias. Security analysis of the mor cryptosystem. In *International Workshop on Public Key Cryptography*, pages 175–186. Springer, 2003.

[61] Gordon E Wall. The structure of a unitary factor group. *Publications Mathématiques de L'IHÉS*, 1:7–23, 1959.

[62] Hans Zassenhaus. On the spinor norm. *Archiv der Mathematik*, 13(1):434–451, 1962.

List of Publications

1. Gaussian elimination in symplectic and orthogonal groups with Ayan Mahalanobis, Anupam Singh and Sushil Bhunia(Submitted to the Journal of Groups, complexity and cryptology).

2. The MOR cryptosystem in orthogonal and symplectic groups in odd characteristics with Ayan Mahalanobis, Anupam Singh and Sushil Bhunia. (preprint)

3. Bilinear cryptography using nilpotent groups of class 2 with Ayan Mahalanobis(Accepted to 16th IMA2017 International Conference on Cryptography and Coding to be held at Catherine's College, University of Orford).