République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université de Batna 2

Faculté des Mathématiques et de l'Informatique

Département de Mathématique

THÈSE

Pour obtenir le titre de

Docteur en Sciences

Option : Mathématiques

Présentée et soutenue par

Makhlouf Sassia

SUR LES CLASSES DE CONJUGAISON D'UN GROUPE ET APPLICATIONS

Soutenue le 29 / 11 / 2018

Jury :

Président:	Guedjiba Said	Professeur	Université de Batna 2
Rapporteur:	Noui Lemnouar	Professeur	Université de Batna 2
Examinateurs:	Amroune Abdelaziz	Professeur	Université de M'sila
	Trabelsi Nadir	Professeur	Université de Sétif 1
	Badis Abdelhafid	MCA	Université de Khenchela

People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

University of Batna 2
Faculty of Mathematics and Computer Science
Department of Mathematics

THESIS

Presented for the degree of

Doctor of Science

Option : Mathematics

By

Makhlouf Sassia

ON CONJUGACY CLASSES OF GROUP AND APPLICATIONS

Thesis defended on 29 / 11 / 2018

Jury :

President:	Guedjiba Said	Professor	University of Batna 2
Supervisor:	Noui Lemnouar	Professor	University of Batna 2
Examiners:	Amroune Abdelaziz	Professor	University of M'sila
	Trabelsi Nadir	Professor	University of Setif 1
	Badis Abdelhafid	MCA	University of Khenchela

Abstrat

Conjugation is an important action both on the elements and the subgroups of a group G.

The concept of the conjugacy plays a central role in representation theory and in some applications.

Several authors have studied the correspondence between special linear codes and conjugacy classes of subgroups of specific groups, more precisely, F. Manganiello et al. [35] have studied the correspondence between cyclic orbit codes and conjugacy classes of subgroups of the general group, J. A. Wood [52] has studied the interaction between self-orthogonal codes and conjugacy classes of maximal abelian subgroups of a compact Lie group.

In this thesis, after a brief survey of conjugacy classes, we focus on applications. We study the conjugacy classes in cryptography. We give characterization of subclass of self-orthogonal codes and particular codes suitable for secret sharing. A part of this dissertation is devoted to the study of the practical applications to secret sharing, encryption and key exchange.

keywords: Conjugacy class; Secret sharing scheme; Self-dual codes; MDS codes; Minimal codeword; Self-orthogonal code.

Résumé

La conjugaison est une action importante à la fois sur les éléments et les sous groupes d'un groupe G.

Le concept de conjugaison joue un rôle central dans la théorie des représentations des groupes et dans certaines applications.

Plusieurs auteurs ont étudié la correspondance entre quelques classes de codes linéaires spéciaux et les classes de conjugaison de sous groupes de groupes spécifiques, plus précisément, F. Manganiello et al. [35] ont étudié la correspondance entre les codes d'orbite cyclique et les classes de conjugaison de sous groupes du groupe linéaire, J. A. Wood [52] a étudié l'interaction entre les codes auto-orthogonaux et les classes de conjugaison de sous groupes maximaux abélians d'un groupe de Lie compact.

Dans cette thèse, aprés un bref aperçu sur les classes de conjugaison, nous étudions les classes de conjugaison en cryptographie. Nous donnons une caractérisation de sous classe de codes auto-orthogonaux et de codes particuliers convenables au partage de secrets.

Une partie de cette thèse est consacrée aux applications pratiques au partage de secrets, cryptage et l'échange de clés.

Mots clés: Classe de conjugaison; Schéma de partage de secret; Codes auto-duaux; Codes MDS; Mot de code minimal; Code auto-orthogonal.

Acknowledgments

First and foremost, I would like to give thanks and praise to Allah, for having made everything possible by giving me the strength, courage and patience to finish this thesis. Thank you, Allah, for loving me and for helping me all the time. Thank you for making me smile when I need to most. Thank you for everything.

I would like to express profoundly my sincere gratitude to my teacher and thesis supervisor, Prof. Noui Lemnouar, Professor at Mathematics Department of the University of Batna 2, for his patience, guidance, corrections, support and encouragement through my work on this thesis. During the various stages in the course of this research project, I learned a lot from working with him, especially how to be an independent researcher, I admire his patience and his calmness. Without his help, this thesis would not have been possible. I will be forever grateful.

Also, I would like to express specials thanks and sincere appreciation to the honourable members of the jury: Prof. Guedjiba Said from the University of Batna 2, Prof. Amroune Abdelaziz from the University of M'sila and Prof. Trabelsi Nadir from the University of setif 1 as well as Dr. Badis Abdelhafid from the University of Khenchela. I am grateful to all of them, for their reading and evaluating my work.

My thanks also go to all my teachers at all levels, since the primary school to the post graduate level.

For their unwavering support, I wish to express my thanks to my colleagues, friends and to my family members, especially to my sister, my best friend Fairouz. I thank their love and encouragement.

" Thanks a lot for all of you ".

Dedicated

to

the memory
of my parents

Contents

List of Figures

List of Tables

List of Symbols and Notations

Groups

Cryptography

Linear codes

Introduction

In a group $(G, \cdot)$, two elements a and b are called conjugate if $a = x\,b\,x^{-1}$ for some $x \in G$.

The conjugacy is an equivalence relation on G, the conjugacy class of a is $\overline{a} = \{x\,a\,x^{-1}, \quad x \in G\}$.

Two subgroups H_1 and H_2 of G are conjugate if there exists some g in G such that $H_2 = gH_1g^{-1}$.

Conjugation is an important action both on the elements and on the subgroups of a group G.

Conjugacy classes plays a key role in some applications. Several researchers have studied the interaction between particular codes and conjugacy classes of subgroups of a specific group, more precisely, [35], to classify the orbit codes which are applicable for communication, F. Manganiello use the classification of the conjugacy classes of cyclic subgroups of the general group. In [52], J. A. Wood has studied the interaction between self-orthogonal codes and conjugacy classes of maximal abelian subgroups of a compact Lie group. The other direction may also be useful, [52], using codes to study the topology of $Spin(n)$.

The first part of this thesis is concerned with the conjugacy classes, the second part focuses on applications, more precisely we study the conjugacy classes in cryptography, we give a characterization of subclass of self orthogonal codes and particular codes suitable for secret sharing, we describe a practical applications of conjugacy classes to secret sharing, encryption and key exchange.

The dissertation is organized as follows:

In the first chapter, some basic concepts and definitions of groups and cryptography are introduced: Cyclic groups, cosets and conjugacy, types of cryptographic algorithms, Diffie-Helman key exchange, ...

in chapter 2 we show the applications of conjugacy classes to cryptography, mainly concerning braid groups, the conjugacy problem and Mor cryptosystem.

Chapter 3 gives the basic knowledge in coding theory which is useful for code-based cryptography, in this chapter we describe some families of linear codes and we present the most successful code-based cryptography: McEliece cryptosystem.

The chapter 4 is devoted to contributions: Characterization of particular codes

suitable to secret sharing, use of cojugacy classes in McEliece cryptosystem which leads to obtain a dynamic keys, a new secret sharing applicable to secret image encryption, conjugacy classes and key exchange.

1 | Preliminaries

Contents

In this chapter we recall the basic notions concerning the theory of groups and cryptography, which will be used in thesis, with references [11, 26, 49], the first one, in our opinion, is a good reference for the theory of groups.

1.1 Groups

1.1.1 Definition and Examples

Definition 1.1. *A group $(G, *)$ is a non-empty set G with an operation $(*)$ on G satisfying the following conditions:*

1. `Clousure:` *for all $a, b \in G$, $a * b$ is an element of G.*
2. `Associativity:` *$(a * b) * c = a * (b * c)$, for all $a, b, c \in G$.*
3. `Identity:` *There exists an (unique) element $e \in G$ such that $a*e = a = e*a$, for all a in G.*
   The element e is called the identity of G.
4. `Inverses:` *For each a in G, there exist an (unique) inverse element b in G suth that $a * b = b * a = e$.*

Remark 1.1. *We will usually abbreviate the notation $(G, *)$ as G .*
Also there are two standard notations for the binary group operation:

- *The additive notation; we write $a + b$ for $a * b$, 0 for e, and the inverse of a is denoted by $(-a)$.*
- *The multiplicative notation; we write ab for $a * b$, 1 for e, and a^{-1} for the inverse of a.*

We will use the multiplicative notation for the operation in a group, unless otherwise specified.

Definition 1.2. *A group G is said to be abelian (or commutative) if for all a and b in G, $a\,b = b\,a$.*

Examples 1.1.

1. $(\mathbb{Z}, +)$ *is an abelian group.*

2. $(\mathbb{Z}, \cdot)$ *is not a group since there are elements which are not invertible in* $\mathbb{Z}$.

3. *The general linear group over* $\mathbb{R}$ *is the group of* $n \times n$ *invertible matrices with real numbers, and is denoted by* $GL_n(\mathbb{R})$. *It is not abelian for* $n > 1$.

4. *Given a set* $X = \{1, 2, \ldots, n\}$, *the set of all permutations of* X *is a group under composition of maps, called the symmetric group, denoted by* S_n. *It is not abelian for* $n > 2$. *We shall use the cyclic notation for permutations: let* $i_1, i_2, \ldots, i_k \in \{1, \ldots, n\}$ *be* k *distinct elements, we use* $(i_1 i_2 \ldots i_k)$ *to denote that* i_1 *is replaced by* i_2, i_2 *is replaced by* i_3, ..., *and* i_k *is replaced by* i_1.

Definition 1.3. *A group* G *is said to be finite if the number of elements in the set* G *is finite. The number of elements is called the order of* G, *denoted by* $|G|$, *otherwise the group is an infinite group.*

Remark 1.2. *Finite groups play an extremely important role in cryptography. As an example of finite groups: let* n *be an integer* ≥ 2 , *the set* $\mathbb{Z}_n = \{0, 1, 2, \ldots, n-1\}$ *is the finite additive group of residues modulo* n.

Example 1.1. $|S_n| = n!$, $|\mathbb{Z}_n| = n$.

Definition 1.4. *let* g *be an element of a group* G. *The order of* g, *is the smallest positive integer* n *such that* $g^n = e$, *denoted by* $o(g)$.
If no such integer n *exists,* g *is said to have infinite order,* $o(g) = \infty$.

Remark 1.3.

1. *Every group has at least one element of finite order (the identity element has order* 1. *So an element has order* 1 *if and only if it is the identity element).*

2. *Every element of a finite group has finite order.*

Example 1.2. *Let* $G = (\mathbb{R} - \{0\}, \cdot)$ *be a group of nonzero real numbers under multiplication, the order of every element except* 1 *and* -1 *is infinite.*

We have $(-1)^1 = -1$, $(-1)^2 = 1$, *so* $o(-1) = 2$.

Now $4^1 = 4$, $4^2 = 16$, $4^3 = 64$ *and so on. Thus there is no positive integer* n *such that* $4^n = 1$. *Therefore* $o(4) = \infty$.

1.1.2 Subgroups and Cyclic Groups

1.1.2.1 Subgroups

Definition 1.5. *A subgroup of a group G is a non-empty subset H of G that forms a group under the binary operation of G, and we write $H \leq G$.*

Remark 1.4. *Every group G has two trivial subgroups, G and $\{e\}$.*
We call the non-trivial subgroup of G, a proper subgroup of G.

Proposition 1.1. *Let G be a group, and let H be a non-empty subset of G. Then H is a subgroup of G if and only if the following conditions hold:*

1. *$a, b \in H \Rightarrow a\,b \in H$ for all a, b.*
2. *$a \in H \Rightarrow a^{-1} \in H$.*

Remark 1.5. *Any subgroup of an abelian group is also abelian.*

Examples 1.2.

1. *For all $n \in \mathbb{Z}$, $n\,\mathbb{Z} = \{n\,x,\ x \in \mathbb{Z}\}$ is a subgroup of $(\mathbb{Z}, +)$.*
2. *If we consider the group $G = \mathbb{Z}_6 = \{0, 1, 2, 3, 4, 5\}$ of integer with addition modulo 6, $H = \{0, 3\}$ is a subgroup of G.*
3. *The set of $n \times n$ matrices with real numbers and determinant of 1 is a subgroup of $GL_n(\mathbb{R})$, denoted by $SL_n(\mathbb{R})$, and named the special linear group.*

1.1.2.2 Homomorphisms

Definition 1.6. *let $(G, *)$ and $(G', \circ)$ be groups. A homomorphism from G to G' is a map $f : G \longrightarrow G'$ such that, for all $x, y \in G$,*

$$f(x * y) = f(x) \circ f(y) \tag{1}$$

Thus if G is additive group and G' is multiplicative group, then (4.1) becomes:

$$f(x + y) = f(x)f(y)$$

Definition 1.7. *Let $f : G \longrightarrow G'$ be a homomorphism. If f is bijective, then f is called an* `isomorphism`*. In this case, we say that G and G' are isomorphic, and we write $G \cong G'$.*

Definition 1.8. *Let $f : G \longrightarrow G'$ be a homomorphism. The subset:*

$$Ker(f) = \{x \in G : \ f(x) = 1\}$$

of G is called the `Kernel` *of the homomorphism f.*

Lemma 1.1. *Let f be a homomorphism from G to G'. Then $Ker(f)$ is a normal subgroup of G.*

Proposition 1.2. *Let $f : G \longrightarrow G'$ be a homomorphism. Then*

$$(f \, is \, injective) \Leftrightarrow (Ker(f) = 1)$$

Definition 1.9. *Let $f : G \longrightarrow G'$ be a homomorphism. The image of f is the subset:*

$$Im(f) = \{f(g) \in G' : \ g \in G\}$$

of G.

Lemma 1.2. *$Im(f)$ is a subgroup of G'.*

1.1.2.3 Cyclic Groups

Definition 1.10. *Let G be a group, and let g be an element of G. The set*

$$\langle g \rangle = \{g^n, \, n \in \mathbb{Z}\}$$

is called the cyclic subgroup of G generated by g.

Corollary 1.1. *Let g be an element of a group G. Then $o(g) = |\langle g \rangle|$.*

Definition 1.11. *A group G is called a cyclic group if there exists $g \in G$ such that $G = \langle g \rangle$, in this case g is called a generator of G.*

Examples 1.3.

1. $(\mathbb{Z}, +)$ *is an infinite cyclic group.*
   We have: $\forall\, a \in \mathbb{Z} :\ \langle a \rangle \ = \ \{a\, k,\, k \in \mathbb{Z}\} = a\, \mathbb{Z}$
   So
   $$(a\, \mathbb{Z} = \mathbb{Z}) \ \Leftrightarrow \ (a = \pm\, 1)$$
   Then
   $$\mathbb{Z} \ = \ \langle 1 \rangle \ = \ \langle -1 \rangle$$
   $\mathbb{Z}$ *has only two generators:* 1 *and* (-1).

2. $\mathbb{Z}_n = \{0, 1, 2, \ldots, n-1\}$ *with addition modulo* n *is a finite cyclic group.*
   $$\mathbb{Z}_n \ = \ \langle 1 \rangle \ = \ \langle n-1 \rangle$$
   other generators are possible depending on n. *For instance,* 3 *is a generator of* $\mathbb{Z}_{10}$ $\Big(\mathbb{Z}_{10} \ = \ \langle 1 \rangle \ = \ \langle 9 \rangle \ = \ \langle 3 \rangle \ = \ \langle 7 \rangle\Big)$.

Lemma 1.3. *Let* G *be a cyclic group generated by* g. *Then*

1. G *is abelian.*

2. *If* G *is infinite, the elements of* G *are*
   $$\ldots\, , g^{-2}\, , g^{-1}\, , e\, , g\, , g^2\, , \ldots$$

3. *If* G *is finite of order* n, *then the elements of* G *are*
   $$e\, , g\, , g^2\, , \ldots\, , g^{n-1}$$
   and $\ g^n = e$.

Theorem 1.1. *Let* G *be a cyclic group;*

1. *If* G *is infinite, then* G *is isomorphic to the group* $(\mathbb{Z}, +)$.

2. *If* G *is finite of order* n, *then* G *is isomorphic to the group* $(\mathbb{Z}_n, +)$.

Remark 1.6.

- *All cyclic groups of the same order are isomorphic.*
- *Every subgroup of a cyclic group is cyclic.*

1.1.3 Cosets and Conjugacy Classes

1.1.3.1 Cosets

Definition 1.12. *Let G be a group, H a subgroup of G. For $a \in G$ the subset*

$$a\,H = \{a\,h,\, h \in H\}$$

of G is called a left coset of H in G, or, more precisely, the left coset generated by a, and similarly the right coset of H generated by a is

$$H\,a = \{h\,a,\, h \in H\}$$

Remark 1.7. *In additive notation, we get $H + a$ (which usually implies that we deal with a commutative group where we do not need to distinguish left and right cosets).*

Example 1.3. *let $G = S_3 = \{e, (12), (13), (23), (123), (132)\}$ be the symmetric group of all permutations of $\{1, 2, 3\}$, $H = \langle (13) \rangle = \{e, (13)\}$ and $g = (12)$*

Then we have: $gH = \{(12), (132)\}$ while $Hg = \{(12), (123)\}$ and so $gH \neq Hg$. Thus for a general subgroup, right and left cosets are different.

Proposition 1.3. *Two cosets $a\,H$ and $b\,H$ are equal if and only if $a^{-1}b \in H$.*

Proof. Suppose that $a\,H = b\,H$, then $H = a^{-1}b\,H$, so $a^{-1}b \in H$. Conversely, if $a^{-1}b \in H$, then $H = a^{-1}b\,H$, so $a\,H = b\,H$. □

Definition 1.13. *A subgroup N of a group G is normal if and only if for all g in G,*

$$gN \;=\; Ng,$$

we denote it $N \lhd G$.

Remark 1.8.

- *In any group G, $\{1\}$ and G are normal subgroups.*
- *Any subgroup H of an abelian group G is normal.*

Example 1.4.

- *The special linear group* $SL_n(\mathbb{R})$ *is a normal subgroup of* $GL_n(\mathbb{R})$.
- *The subgroup* $\langle(123)\rangle = \{e, (123), (132)\}$ *in* S_3 *is normal.*

Definition 1.14. *The index of a subgroup* H *in* G *is the number of distinct left (right) cosets. It is usually denoted by* $[G : H]$.

Theorem 1.2 (Lagrange's Theorem)**.** *If* H *is a subgroup of a finite group* G*, then*

$$|G| = [G : H] \cdot |H|$$

- We note that $|H|$ and $[G : H]$ divides $|G|$.
- In particular: the order of every subgroup of a finite group divides the order of the group.

Proof. The left (right) cosets of H in G form a partition of G, so there are $[G : H]$ of them, and each left (right) cosets has $|H|$ elements. □

Example 1.5. *Since* $|S_3| = 3! = 6$*, so the only possible orders for a subgroup are* $1, 2, 3$ *and* 6*. And also they are the only possible values for the index of a subgroup of* S_3

Definition 1.15. *Let* N *be a normal subgroup of a group* G*. The set of all left costs of* N *in* G *is a group denoted by* G/N *under the operation* $g_1 N\, g_2 N = g_1 g_2 N$*. This group is called the quotient group of* G *by* N*. The identity element of* G/N *is* N *and* $(gN)^{-1} = g^{-1}N$*, i.e.,* $g^{-1}N$ *is the inverse of* gN*.*

Theorem 1.3.

1. *Let* G *be a finite group, if* g *in* G*, then the order of* g *divides* $|G|$*. In particular,* $g^{|G|} = 1$.
2. *If* G *has a prime order, then* G *is cyclic.*

Proof.

1. If g in G has order say n, hence the elements

$$1, g, g^2, \ldots, g^{n-1}. \, (g^n = 1)$$

form a cyclic group of G with order n. By lagrange's Theorem n divides $|G|$.

2. Since G has a prime order, the order of any subgroup is either one or p, so we may take $g \neq 1$ in G, and since the order of g divides $|G|$, so $o(g) = |G|$. Then the cyclic group generatde by g coincides with G.

□

1.1.3.2 Conjugacy Classes

Let G be a group, and consider the following relation $\sim$ on G:
Let $a, b \in G$, we put

$$\Big(a \sim b\Big) \quad \Leftrightarrow \quad \Big(there\, exists \quad g \in G \quad such\, that \quad a = g\, b\, g^{-1}\Big)$$

Thus:

$$\Big(a \sim b\Big) \quad \Leftrightarrow \quad \Big(a \quad is\, a\, conjugate\, of \quad b\Big)$$

Definition 1.16. *The relation $\sim$ is named the conjugacy relation.*

Lemma 1.4. *The conjugacy relation $\sim$ is an equivalence relation on G.*

Proof.

1. `Reflexivity:` $a \sim a$, $\forall a \in G$.
   We have; $e\, a\, e^{-1} = a$, i.e., $a \sim a$, so the relation is reflexive.

2. `Symmetry:` $a \sim b \Rightarrow b \sim a$, $\quad \forall\, a, b \in G$.

   Assume that $a \sim b$, then $\exists g \in G$ such that $g\, b\, g^{-1} = a$.
   But then, $g^{-1}\, a\, (g^{-1})^{-1} = b$, so that $b \sim a$.

3. `Transitivity:` $a \sim b \wedge b \sim c \Rightarrow a \sim c, \forall a, b, c \in G.$
   If $a \sim b$ and $b \sim c$, then we have $g_1, g_2 \in G$, such that $g_1 b g_1^{-1} = a$ and $g_2 c g_2^{-1} = b$.
   Then,

$$g_1 \left(g_2 c g_2^{-1} \right) g_1^{-1} = \left(g_1 g_2 \right) c \left(g_1 g_2 \right)^{-1} = a$$

   That is, $a \sim c$.

□

Definition 1.17. *The equivalence classes of this equivalence relation are called the conjugacy classes of G.*
For each a in G we denoted its conjugacy class by $\overline{a}$, and as such;

$$\overline{a} = \{g a g^{-1}, \quad g \in G\}.$$

Remark 1.9. *The equivalence classes form a partition of G.*

Examples 1.4.

1. *If G is an abelian group, then every element is a conjugacy of its own: $x g x^{-1} = g, \quad for\, all\ x \in G.$*

2. *let $G = S_3$, the conjugacy classes are:*

   - $\overline{e} = \{g e g^{-1}, \quad g \in G\} = \{e\}.$

   - $\overline{(12)} = \{\tau (12) \tau^{-1} : \tau \in G\}.$

τ	(1)	(12)	(13)	(23)	(123)	(132)
$\tau(12)\tau^{-1}$	(12)	(12)	(23)	(13)	(23)	(13)

   Therefore $\overline{(12)} = \{(12), (13), (23)\} = \overline{(13)} = \overline{(23)}.$

- $\overline{(123)} = \{\tau\,(123)\,\tau^{-1} : \tau \in G\}$.

τ	(1)	(12)	(13)	(23)	(123)	(132)
$\tau(123)\tau^{-1}$	(123)	(132)	(132)	(132)	(123)	(123)

Therefore $\overline{(123)} = \{(123), (132)\} = \overline{(132)}$.

In general, the number of conjugacy classes of the symmetric group S_n is equal to the number of partitions of n, where a partition of a positive number n is a sequence of positive integers $(i_1, i_2, \cdots, i_k)$ such that $i_1 \geq i_2 \geq \cdots \geq i_k$ and $\sum_{j=1}^{k} i_j = n$.

3. *Two permutations are conjugate in S_n if and only if they consist of the same number of disjoint cycles of the same lengths. Thus, cycles of the same length are always conjugate. For instance, in S_3, (12) and (13) are conjugate; (123) and (23) are not.*

Theorem 1.4. *Any two elements of a conjugacy class have the same order.*

Remark 1.10. *The converse of Theorem (1.4) is false: This is clear in abelian groups, where different elements could have the same order, but they are never conjugate.*

Definition 1.18. *Let $a \in G$, the centralizer $C_G(a)$ of a in G, is the set of all elements of G which commute with a. Thus*

$$C_G(a) = \{g \in G \mid g\,a = a\,g\}$$

Theorem 1.5. *Let G be a finite group and let a be an element of G. Then the centralizer $C_G(a)$ of a is a subgroup of G and*

$$|\overline{a}| = [G : C_G(a)] = \frac{|G|}{|C_G(a)|}$$

That is, the conjugacy class of a has the same size as the index of its centralizer.

In particular the size of each conjugacy class divides the order $|G|$ of the group.

Definition 1.19. *The set $Z(G)$ of elements which commute with every element of G is called the center of G. Thus,*

$$Z(G) = \{\, g \mid x\, g = g\, x \ for\, all\ x \in G\}$$

Note that $Z(G)$ is an abelian subgroup of G.

Remark 1.11.

1. *If $a \in Z(G)$ then $\overline{a} = \{a\}$.*
2. *$a \in Z(G)$ if and only if $C_G(a) = G$.*

Class Equation

Theorem 1.6 (Conjugacy Class Equation)**.** *Let G be a finite group. Then*

$$|G| \;=\; |Z(G)| \;+\; \sum_{a \in T} [G : C_G(a)]$$

Where T contains exactly one representative from every conjugacy class of G of size larger than one.

Example 1.6.

1. *Let $G = S_3$. The center of S_3 is $Z(S_3) = \{e\}$, and we have the nontrivial conjugacy classes:*

   - $\overline{(12)} = \{(12), (13), (23)\}$
   - $\overline{(123)} = \{(123), (132)\}$

   If we take one element from each nontrivial class, say (12) and (123). Then the centralizers are:

   - $C_G((12)) = \{e, (12)\}$.
   - $C_G((123)) = \{e, (123), (132)\}$.

   Therefore, class equation is:

   $|S_3| = |Z(S_3)| \;+\; [S_3 : C_G((12))] + [S_3 : C_G((123))] = 1 + 3 + 2 = 6.$

2. *If G is an abelian group of order n, then the class equation of G is: $n = 1 + 1 + \cdots + 1$ (n times). Since G is an abelian group, all the conjugacy classes are singleton sets of size* 1.

Remark 1.12.

1. $[G : C(a)] > 1$, *since* $[G : C(a)]$ *equals the number of elements in the conjugacy class* $\overline{a}$, *where* $a \in G \setminus Z(G)$.

2. $|C(a)| < |G|$, *because* $|G| = [G : C(a)] \cdot |C(a)|$, *and* $[G : C(a)] > 1$ ($a \in G \setminus Z(G)$).

Definition 1.20. *Let p be a prime number. A p-group is a group in which every element has order a power of p.*

Remark 1.13. *A finite group is a p-group if and only if its order is a power of p.*

Lagrange's Theorem shows that the order of every element of a p-group must also be a power of p.

Theorem 1.7. *Let $|G| = p^n$ where p is a prime. Let $Z(G)$ be the center of G. Then p divides $|Z(G)|$. In particular, $Z(G)$ is nontrivial.*

Proof. Use the class equation:

$$|G| = |Z(G)| + \sum_{a \in T} [G : C_G(a)]$$

Because $|G| = p^n$ where $n > 0$, p divides $|G|$ and is greater than 1, hence is divisible by p. Therefore p divides $|Z(G)|$. So, $|Z(G)| \neq 1$. □

Theorem 1.8. *Let p be a prime. Then every group of order p^2 is abelian.*

FC-Groups

It is well-known that the conjugacy classes in a group reflect properties of this group, there is a strong connection between the conjugacy class sizes and the structure of a group. For instance, the group in which every conjugacy class is finite, these groups are called FC-groups, introduced by Bear in [9]. In [43] B. H.

Neumann proved that if G is a group where its conjugacy classes are finite and of bounded size (that is $[G : C_G(a)] \leq n$ for each element $a \in G$ and for some fixed integer n), then the derived subgroup G' is finite. Another interesting connection between properties of groups and conjugacy classes can be founed, for example in [12], [29].

Definition 1.21. *A group G is called an FC-group if for every $g \in G$, the conjugacy class of g in G is finite.*

This is equivalent to saying that a group G is an FC-group if and only if the centralizer $C_G(a)$ is of finite index in G for each $a \in G$.

Example 1.7.

- *Finite groups;*
- *Abelian groups;*
- *groups whose derived subgroup G' is finite.*

Remark 1.14.

1. *Every subgroup of an FC-group is an FC-group.*
2. *If G is a finitely generated FC-group, then $G/Z(G)$ is finite.*

1.2 Cryptography

1.2.1 Definition

Cryptography is the science of secret communication and it is an ancient art. The word cryptography comes from the ancient Greek words " kryptos " (hidden or secret) and " graphein " (writing).

Cryptography is an applied branch of mathematics and it is the science of using mathematics to encrypt and decrypt information (data). The basic idea of cryptography is the use of so-called one way functions or mathematical functions: A function $y = f(x)$ is one way if it's easy to compute y from x, but it's very hard to compute the value of $f^{-1}(y)$. As an example: Discrete logarithm and hash functions.

In cryptography, the original message (plaintext or cleartext) is converted into a coded equivalent called " ciphertext", this process is called encryption. The process of turning ciphertext to its original plaintext is called decryption, Figure 1.1 illustrates this process.

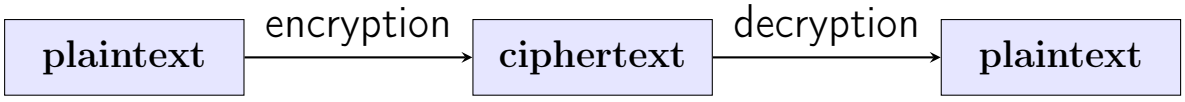


Figure 1.1: Encryption and decryption

One the information has encrypted, it can be transmitted through an insecure network, such as the internet or stored on an insecure media. So that it becomes unintelligible to anybody except the intended recipient.

In some situations, cryptography can be used to provide the following services:

- Confidentiality (Secrecy).
- Integrity.
- authentication.

- Non-repudiation.

A cryptographic algorithm works in combination with a key such as: number, word or phrase to encrypt the original message. There are several ways to classify cryptographic algorithms, the most types of algorithms are:

- Symmetric key cryptography: 1 key (Private key).
- Asymmetric key cryptography: 2 keys (Private key and public key).

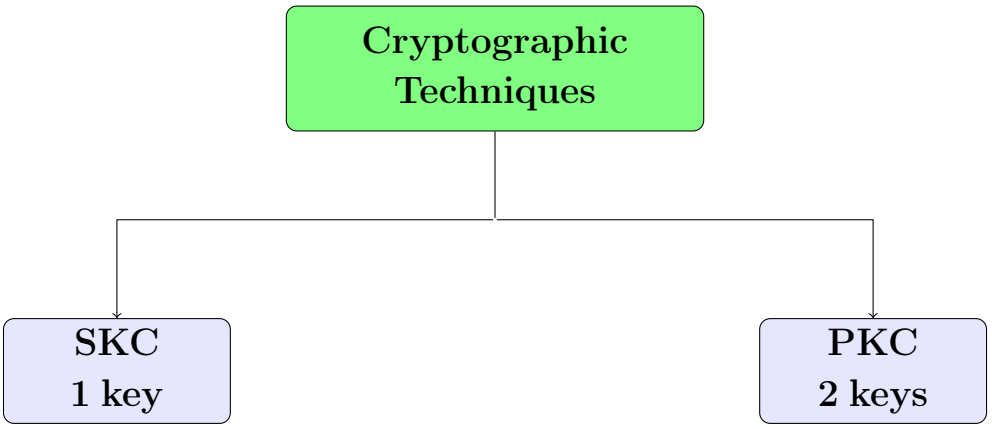


Figure 1.2: Cryptography techniques

1.2.2 Types of Cryptographic Algorithms

1.2.2.1 Secret Key Cryptography (SKC)

This type of cryptography technique uses a same key for both encryption and decryption. As shown in Figure 1.3, the sender uses the key to encrypt plaintext message and the receiver applies the same key to decrytpt the ciphertext. Because only single key is used, secret key cryptography is also known as " symmetric key cryptography ". The highest difficulty with this technique is the distribution of the secret (key).

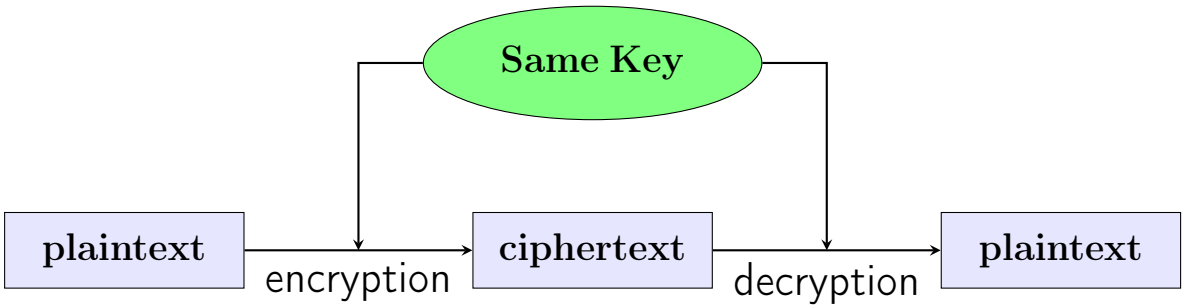


Figure 1.3: Secret key cryptography

Some examples of secret key cryptography are Data Encryption Standard (DES), International Data Encryption Algorithm (IDEA), and Advanced Encryption Standard (AES).

1.2.2.2 Public Key Cryptography (PKC)

The difficulty of key distribution is solved by public key cryptography, it was invented by Whitfield Diffie and Martin Hellman in 1976. The basic technique of public key cryptography was first discovered in 1973 by the British Secret service, but this was a secret until 1997. Public key cryptography also called Asymmetric key cryptography, uses two different keys. One public key for encryption and another different key " private or secret key " for decryption.

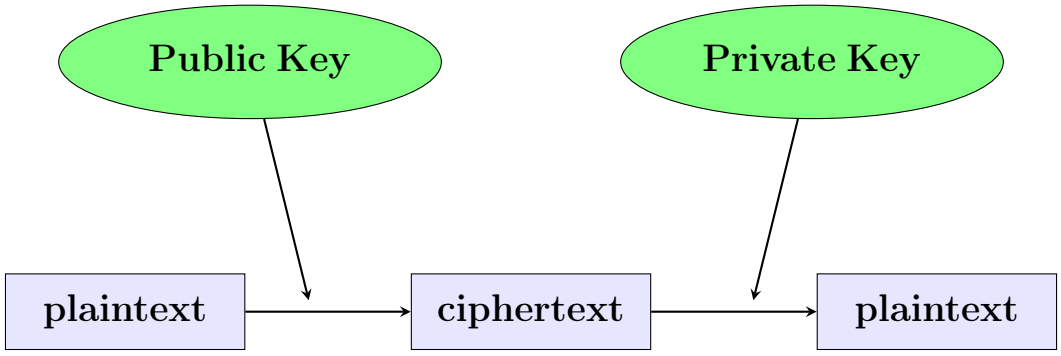


Figure 1.4: Public key cryptography

In public key cryptography, each participant has a pair of keys: a public key and a private key. The private key is kept secret while the public key is shared

with every one.

Some examples of public key algorithm are Digital Signature Algorithm (DSA), Algorithm RSA (named for its inventors; Ron Rivest, Adishamir and Leonard Adleman), and Elgamal (named for its inventor, Tahar Elgamal).

We describe in the following, the first published public-key algorithm: Diffie-Hellman key exchange.

Diffie-Hellman Key Exchange

In 1976, Whitfield Diffie and Martin Hellman introduced a key exchange protocol [18]. The protocol is a specific method of exchanging cryptographic keys. The Diffie- Hellman key exchange algorithm allows two parties to exchange a secret key over an insecure medium without any prior secrets. The Diffie- Hellman key algorithm based on the difficulty of computing discrete logarithms.

Problem Discrete logarithm problem (DLP)

let $G = \langle g \rangle$ be a cyclic group generated by g.

Given two elements g and $g^a \in G$, the problem is to find the a.

Algorithm Description

Suppose the users Alice (A) and Bob (B) wish to exchange a key:

1. Public information
   (a) A prime number p.
   (b) An integer g, where g is a primitive root of p (that is g is a generator of the multiplicative group $((\mathbb{Z}/p\mathbb{Z})^*, \cdot)$).
2. Key agreement
   (a) Alice chooses a secret integer a, and sends $Y_A = g^a \pmod p$ to Bob.
   (b) Bob chooses a secret integer b and sends $Y_B = g^b \pmod p$ to Alice.
   (c) Alice receives Y_B and computes the secret key: $K = Y_B^a \pmod p$.
   (d) Bob receives Y_A and computes the secret key: $K = Y_A^b \pmod p$.

Example 1.8.

1. *Suppose prime number* $p = 7$ *and the primitive root of* p *is* 3, $g = 3$.
2. *Alice chooses* $a = 2$ *and sends* $Y_A = 3^2 \pmod 7 = 2$.
3. *Bob chooses* $b = 4$ *and sends* $Y_B = 3^4 \pmod 7 = 4$.

   After the exchange public key:
4. *Alice compute:* $K = Y_B^a = 4^2 \pmod 7 = 2$.
5. *Bob compute:* $K = Y_A^b = 2^4 \pmod 7 = 2$.

1.2.2.3 Hash Functions (HF)

The Public key cryptography is 1000 times slower than secret key cryptography, it produces an volume of ciphertext at least double the size of the original plaintext message.

A cryptography hash function is a mathematical transformation that takes an arbitrary length of message and produces a fixed length (short) number.

Cryptographic hash functions are used to check the integrity of the file to ensure that the file has not be altered or changed.

There are sevral hash functions in use today: message Digest (MD) Algorithms: MD2, MD4, MD5, and Secure Hash Algorithms (SHA): SHA1, SHA2, SHA3.

1.2.3 Secret Sharing Scheme (SSS)

In cryptography, secure storage of sensitive data as the private key is an important problem. For example, a key that is saved by an individual can easily be lost, giving copies to several people increases the risk of compromise. As a solution of this problem is the secret sharing scheme.

A secret sharing scheme (SSS) is a method to distribute a sensitive secret to a number of parts (shares) in such a way that only some specified shares can reconstruct the secret. The goal of a secret sharing scheme is to ensure that unauthorized subset of shares do not be able to gain any information about the secret.

Secret sharing schemes were introduced independently by Adi Shamir [48] and G. Blakely [10] respectively in 1979. A (t, n) secret sharing scheme is a method to

distribute a secret S among n people such that any t or more can construct the secret S, but $(t-1)$ can not.

The first SSS was proposed by Shamir [48] based on Lagrange interpolation polynomial. To obtain a (t, n) secret sharing, a random polynomial of degree $(t-1)$ is generated over a finite field $\mathbb{F}$ where p is a prime number. To construct the polynomial

$$f(x) \;=\; a_0 + a_1 x + a_2 x^2 + \cdots + a_{t-1} x^{t-1}$$

$a_0 = S$ is the secret and the coefficients a_1 to a_{t-1} are randomly chosen in $\mathbb{F}_p$. The share $(i, f(i))$, $1 \leqslant i \leqslant n$, is given to user i. If t or more users come together, they can construct the polynomial using Lagrange interpolation and obtain the secret (see [48] or [49]).

Example Let the secret $S = 8$, let $n = 4$ and $t = 3$. Let $(t-1)$ that is 2 coefficients are 7 and 3. The polynomial is $f(x) = 8 + 7x + 3x^2$ over the filed $\mathbb{Z}_{13}$. The corresponding secret shares are:

$$\begin{aligned}
(\ 1,\ f(1)\ ) &= (\ 1,\ 8 + 7(1) + 3(1)^2 = 18 \ mod\ 13\ ) = (1, 5) \\
(\ 2,\ f(2)\ ) &= (\ 2,\ 8 + 7(2) + 3(2)^2 = 34 \ mod\ 13\ ) = (2, 8) \\
(\ 3,\ f(3)\ ) &= (\ 3,\ 8 + 7(3) + 3(3)^2 = 56 \ mod\ 13\ ) = (3, 4) \\
(\ 4,\ f(4)\ ) &= (\ 4,\ 8 + 7(4) + 3(4)^2 = 279 \ mod\ 13\ ) = (4, 6)
\end{aligned}$$

$t = 3$ so at least three users can find the secret by Lagrange's interpolation formula:

$$f(x) \;=\; \sum_{j=1}^{t} f(j) \left(\prod_{i=1, i \neq j}^{t} \frac{x - x_i}{x_j - x_i} \right)$$

Consider first three users $(1, 5)$, $(2, 8)$ and $(3, 4)$, the secret can be calculated as:

$$\begin{aligned}
f(0) \;&=\; \frac{5 \cdot 2 \cdot 3}{(1-2)(1-3)} \;+\; \frac{8 \cdot 1 \cdot 3}{(2-1)(2-3)} \;+\; \frac{4 \cdot 1 \cdot 2}{(3-1)(3-2)} \\
&=\; 8 \ mod\ 13
\end{aligned}$$

Therefore $S \;=\; 8$

Now let consider another combinations of shares as $(1, 5)$, $(3, 4)$ and $(4, 6)$:

$$f(0) = \frac{5 \cdot 3 \cdot 4}{(1-3)(1-4)} + \frac{4 \cdot 1 \cdot 4}{(3-1)(3-4)} + \frac{6 \cdot 1 \cdot 3}{(4-1)(4-3)}$$

$$= 8 \; mod \; 13, \; which \; is \; our \; secret.$$

Another approach is Blakely SSS, based on hyper plane geometry [10]: For implementation a (t, n) secret sharing scheme, a hyper-plane equation in a t dimensional space over a finite field is given to each of the n users such that each hyper-plane passes through a certain point. The intersection points of the hyper-plane gives the original secret. When t users come together, they can solve the system of equations to reconstruct the original secret.

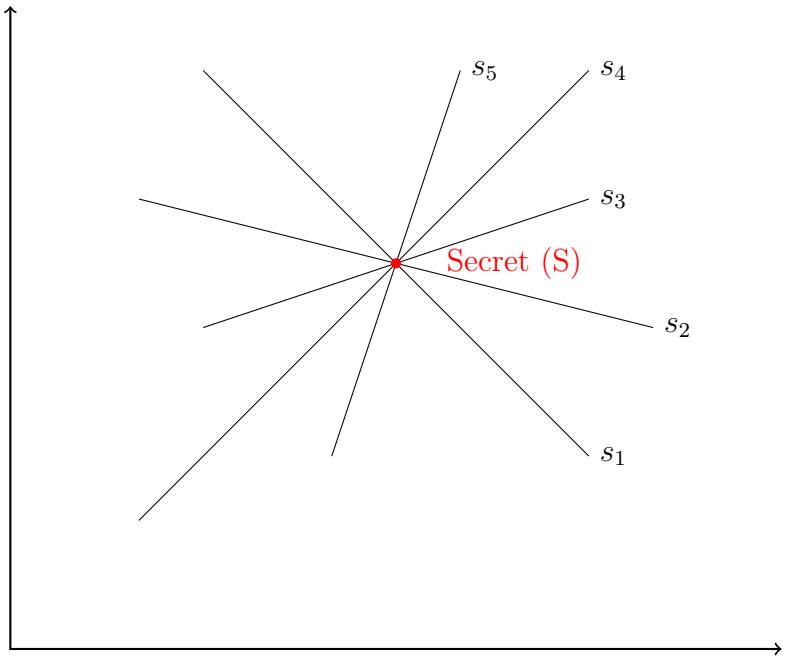


Figure 1.5: Blakley's scheme with t=2, n=5, and original secret S .

2 | Conjugacy Classes in Cryptography

Contents

Group-based cryptography is about the application and the use of group theory to cryptography. Most modern cryptography schemes use groups, they are based on algebraic structure of groups. Groups theory is used especialy in public key cryptography (PKC), such as:

- Diffie-Hellman key exchange uses finite cyclic group with generator g, which is based on the discrete logarithm problem (DLP).
- The Elgamal cryptographic algorithm, is based on the discrete logarithm problem (DLP)(as well as the Diffie-Hellman key exchange) in cyclic group of large prime order.

This type of encryption is called commutative cryptography. To improve the securety of cryptographic protocols, non-commutative cryptography was introduced.

The first use of non-abelian groups (non-commutative cryptography) in cryptography is proposed by Wagner and Magyarik [51] in 1985. In group-based cryptography, non-abelian groups are used in encryption and decryption. The primary sources for non-abelian groups: linear group theory and combinatorial group theory. Braid group cryptography [16], where encryption is done within braid groups, is one prominent example. Braid groups is one of the main platforms in braid-group cryptography (see [3], [5], [30], [31], [17]). The one-way function in braid group systems is based on the difficulty of solving group theoretic search and decision problems such as the conjugacy search problem (CSP). The conjugacy search problem (CSP) plays a major role in group-based cryptography. In this chapter we will present two well-known group-based key agreement protocols: The first is the Anshel, Anshel and Goldfeld protocol [3] in 1999, the other is Ko et al. protocol [30] in 2000. We also present the MOR cryptosystem.

2.1 Conjugacy Search Problem (CSP)

We describe some of mathematically hard problems related to conjugacy, which is may be interesting in cryptography:

2.1.1 Conjugacy Decision Problem (CDP)

given two elements h, g of a group G, the conjugacy decision problem (CDP) is to decide whether h and g are conjugate or not? that is, is there an x in G such that $h = xgx^{-1}$?

2.1.2 Conjugacy Search Problem (CSP)

An analogue to the discrete logarithm problem (DLP) in arbitrary groups is the conjugacy search problem (CSP): given two elements h, g of a group G such that h and g are conjugate, i.e. we have $h = kgk^{-1}$. The problem is to find an element x in G such that $h = xgx^{-1}$.

There exist a variant to the conjugacy search problem, namely the multiple simultaneous conjugacy search problem (SCSP), otherwise known as the generalized conjugacy search problem: given $(h_1, h_2, \ldots, h_n), (g_1, g_2, \ldots, g_n)$ of a group G such that $h_i = xg_ix^{-1}$ for some $x \in G$. The problem is to find an element $y \in G$ such that $h_i = yg_iy^{-1}$ for all $i = 1, \ldots, n$. The primary example of the conjugacy search problem being used in cryptography is the Anshel-Anshel-Goledfeld (AAG) protocol, we discuss AAG protocol in section 2.2.1.

2.2 Braid Groups and The Conjugacy Problem

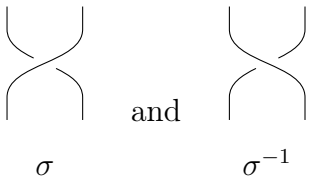
The braid groups were first introduced by E. Artin in 1926 [6], these groups play important role in many areas of mathematics, in particular public key cryptography. There are several definitions for braid groups, see [27], [42] for more details of these groups and an excellent introduction to their basic properties can be found.

Definition 2.1. *The braid group B_n is defined by the Artin presentation:*

$$B_n = \left\langle \sigma_1, \ldots, \sigma_{n-1} \;\middle|\; \begin{array}{rcll} \sigma_i\sigma_j\sigma_i & = & \sigma_j\sigma_i\sigma_j & if \mid i-j \mid = 1 \\ \sigma_i\sigma_j & = & \sigma_j\sigma_i & if \mid i-j \mid > 1 \end{array} \right\rangle$$

Where $\sigma_i,\ i = 1, \ldots, n-1$ is the generator of the group B_n, formed by crossing the ith string under the $(i+1)$th string (see Figure: 2.1).

Note that, there are only two types of crossings, the under-crossing (the right side passes under the left) noted by σ and the over-crossing (the left side passes under the right) noted by σ^1.



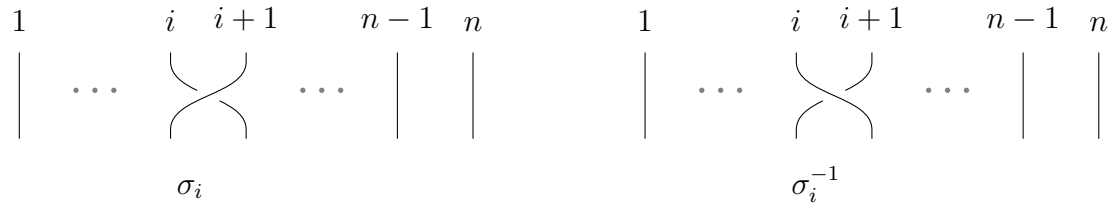


Figure 2.1: The elementary braids σ_i and σ_i^{-1}

Any element of B_n is called an n-braid, and every n-braid can be represented by a finite number of elementary braids σ_i and their inverse σ_i^{-1} (Figure: 2.1), an example is shown in Figure 2.2. Note that for $n > 1$, B_n is infinite. B_2 is isomorphic to $\mathbb{Z}$. For $n \geqslant 3$, B_n is non-abelian and $Z(B_n)$ is isomorphic to $\mathbb{Z}$.

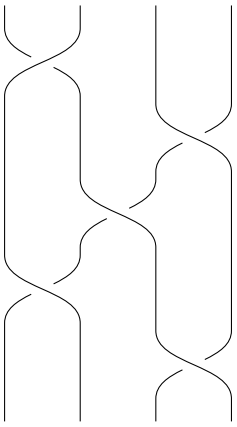


Figure 2.2: An example of a 4-braid : $\sigma_1^{-1}\sigma_3\sigma_2\sigma_1\sigma_3$

2.2.1 Key Agreement Protocols and Cryptosystem Based on Braid Groups

This paragraph gives the basic definitions for key agreement protocols and cryptosystem, which are based on the difficulty of the conjugacy problem for braid groups.

2.2.1.1 Anshel-Anshel-Goldfeld Protocol

This protocol was proposed by Anshel, Anshel and Goldfeld [3] in 1999, and then later in 2001, the protocol was implemented to the braid groups by the original authors and Fisher [4].

1. Public information:
   (a) The braid index n is fixed.
   (b) Alice publishes a subgroup of B_n, say $S_A = \langle s_1, s_2, \ldots, s_r \rangle$, where s_i are arbitrary.
   (c) Bob publishes a subgroup of B_n, say $S_B = \langle t_1, t_2, \ldots, t_y \rangle$, where t_j are arbitrary.
2. Key agreement:
   (a) Alice chooses a secret element $a = s_{i_1} s_{i_2} \ldots s_{i_k} \in S_A$ and sends $(t_1, at_1a^{-1}), \ldots, (t_y, at_ya^{-1})$ to Bob.
   (b) Bob chooses a secret element $b = t_{j_1} t_{j_2} \ldots t_{j_l} \in S_B$ and sends $(s_1, bs_1b^{-1}), \ldots, (s_r, bs_rb^{-1})$ to Alice.
   (c) Alice computes the shared key: $K = (bs_{i_1}b^{-1}) \ldots (bs_{i_k}b^{-1})a^{-1} = (bab^{-1})a^{-1}$.
   (d) Bob computes the shared key: $K = b[(at_{j_1}a^{-1}) \ldots (at_{j_l}a^{-1})]^{-1} = b[aba^{-1}]^{-1}$.

But, the shared keys computed by Bob and Alice may be different as bit strangs, so for extracting the same bit strang from $bab^{-1}a^{-1}$, Anshal et al. [3], used the colored Bureau representation of the braid group defined by Morton [41].

The second protocol we describe was introduced in 2000 by Ko et al. [30].

2.2.1.2 Diffie-Hellman Conjugacy Protocol

Ko et al. proposed a key agreement protocol, based on the conjugacy problem on braid groups, using commutative property of some of its elements.

For the braid group B_{2n}, consider the following subgroups:

$$LB_{2n} = \langle \sigma_1, \ldots, \sigma_{n-1} \rangle \quad and \quad RB_{2n} = \langle \sigma_{n+1}, \ldots, \sigma_{2n-1} \rangle$$

Where, for any $x \in LB_{2n}$ and $y \in RB_{2n}$, $xy = yx$.

1. Public information
   (a) A relatively complicated braid $x \in B_{2n}$ is published.
2. Key agreement
   (a) Alice chooses a secret braid $a \in LB_{2n}$ and sends $y_a = axa^{-1}$ to Bob.
   (b) Bobe chooses a secret braid $b \in RB_{2n}$ and sends $y_b = bxb^{-1}$ to Alice.
   (c) Alice receives y_b and computes the shared key: $K = ay_ba^{-1} = abxb^{-1}a^{-1}$.
   (d) Bob receives y_a and computes the shared key: $K = by_ab^{-1} = baxa^{-1}b^{-1}$.

In order to make sure that Alice and Bob generate the same secret key, Ko et al. [30] used the Garside normal form.

Remark 2.1. *Ko et al. also propose a public key encryption scheme [30], the security of which is also based on the Conjugacy Search Problem.*

2.3 MOR Cryptosystem

In this section we present the MOR cryptosystem. The concept of the MOR cryptosystem was introduced by Paeng et al. in 2001, see [44]. The system's security is based on the hardness of the special conjugacy search problem and the DLP in the inner automorphism group. Later in same year, Paeng et al. [45], generalized the MOR cryptosystem.

Let G be a non-abelian group with non trivial center $Z(G)$. We assume that $Z(G)$ is not small.

Definition 2.2. *Let g be an element of G. An automorphism*

$$\begin{aligned} Inn(g)\colon G &\longrightarrow G \\ x &\longmapsto gxg^{-1} \end{aligned}$$

is called the inner automorphism. Note that:

- $(Inn(g))^b = Inn(g^b)$.
- $Inn(g) = id_G \Leftrightarrow g \in Z(G)$.

The set of all inner automophisms of G, called the inner automorphism group and denoted by $Inn(G)$.

Problem Special Conjugacy Search Problem

Gven an element $Inn(g) \in Inn(G)$, the special conjugacy search problem (SCSP) is to find $g' \in G$ such that $Inn(g) = Inn(g')$.

2.3.1 Public Key Encryption Scheme: MOR

Alice chooses arbitrary elements $g \in G$ and $a \in \mathbb{N}$.

Alice's keys are as follows:

- Public key: $Inn(g)$, $Inn(g^a)$.
- Secret key: a.

Encryption

To send a message $m \in G$ to Alice:

- Bob chooses an arbitrary $b \in \mathbb{N}$ and compute $(Inn(g^a))^b$.
- Bob computes $E = Inn(g^{ab})(m) = (Inn(g^a))^b(m)$.
- Bob computes $\varphi = (Inn(g))^b$.
- Bob sends to Alice the cipher (E, φ).

Decryption

Alice knows a, so she compute:

$$m = \varphi^{-a}(E) = Inn(g^{-ab})(E) = Inn(g^{-ab})(Inn(g^{ab})(m)).$$

3 | Linear Codes

Contents

This chapter presents the basic knowledge about coding theory wich is useful for code-based cryptography. We introduce some basic concepts and definitions and describe some families of linear codes which play a role in this thesis. The following definitions are extracted from [38]. More details can be found in [24], [25].

Then we describe the first and the most succesful code-based cryptosystem: The McEliece cryptosystem.

3.1 Basic Definitions

An important class of codes are linear codes in the vector space $\mathbb{F}_q^n$, $\mathbb{F}_q$ is a finite filed of cardinality q , where $q = p^m$, p is a prime and m is a positive integer.

Definition 3.1. *A linear $[n, k]_q$ code C of length n and dimension k is a k-dimensional subspace of $\mathbb{F}_q^n$. The elements of the code are called codewords.*

Example 3.1. *The repetition code* $C = \{ \underbrace{(x, \ldots, x)}_{n} \mid x \in \mathbb{F}_q \}$ *is a linear code.*

Remark 3.1.

- *In a linear code, any linear combination of codewords is also a codeword.*
- *The zero vector* 0 *is always a codeword of any linear code.*
- *The size of a code is the number of codewords and equals q^k, denoted by $|C| = q^k$.*
- *A binary code C is linear if and only if contains the zero codeword* 0 *and the sum of any two codewords is another codeword, i.e.,*

  for all $c_1, c_2 \in C : \quad c_1 + c_2 \in C$

Definition 3.2 (Hamming distance)**.**

The Hamming distance $d(x, y)$ between two codewords $x = (x_1, x_2, \ldots, x_n)$ and $y = (y_1, y_2, \ldots, y_n)$ is the number of coefficients in which they differ, i.e.,

$$d(x, y) = |\{ i, x_i \neq y_i \}|$$

Example 3.2.

$In\ \mathbb{F}_2^6\ ;\quad d(\ 010101,\ 111011\ ) = 4$
$In\ \mathbb{F}_3^4\ ;\quad d(\ 0221,\ 2210\ ) = 3$

Definition 3.3 (Minimum distance)**.** *The minimum distance of a code C is*

$$d = min\{\ d(x,y)\ |\ \ x,y \in C\ ;\ \ x \neq y\ \}$$

A linear $[\ n,k\ ]$ code with minimum distance d is often denoted as $[\ n,k,d\ ]$.

Definition 3.4 (Support)**.** *The support of a codeword $c = (\ c_1,\ldots,c_n\ ) \in C$ is defined to be the set:*

$$supp(c) = \{\ 1 \leqslant i \leqslant n\,,\ c_i \neq 0\ \}$$

.

Definition 3.5 (Hamming weight)**.** *The Hamming weight $w(c)$ of a codeword $c \in C$ is the cardinality of its support, i.e.,*

$$w(c) = |\ supp(c)\ |$$

Definition 3.6 (Minimum weight)**.** *The minimum Hamming weight of C is the smallest of the weights of the nonzero codewords of C, denoted by $w(C)$, i.e.,*

$$w(C) = min\{\ w(c)\ |\ c \in C\ ;\ \ c \neq 0\ \}$$

Lemma 3.1. *If C is a linear code, then d is the minimum Hamming weight of C, i.e.,*

$$d(C) = w(C)$$

Generator and Parity-Check Matrices

Since a linear $[\ n,k\ ]$ code is a k-dimensional subspace of $\mathbb{F}_q^n$ for some integer k with $1 \leqslant k \leqslant n$, so we can describe it by giving a basis, wich consists of k linearly independent codewords in C.

It is customary to put the basis vectors into a matrix.

Definition 3.7 (Generator matrix)**.** *let C be an $[\ n, k, d\ ]$ code with a basis $A = \{\ a_1, \ldots, a_k\ \}$. If*

$$\begin{aligned} a_1 &= a_{11}\ a_{12}\ \ldots\ a_{1n} \\ a_2 &= a_{21}\ a_{22}\ \ldots\ a_{2n} \\ &\vdots \\ a_k &= a_{k1}\ a_{k2}\ \ldots\ a_{kn} \end{aligned}$$

then the $k \times n$ matrix;

$$G = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k1} & a_{k2} & \ldots & a_{kn} \end{pmatrix}$$

whose rows are the codewords in A, is called the generator matrix for C.

Note that a generator matrix for C must be a $k \times n$ matrix with rang k. Since the vector space can have many different bases, the generator matrix for linear code is not unique.

Proposition 3.1. *Any linear code can be defined by its generator matrix. Indeed, all the codewords can be generated by this matrix*

$$C = \{\ x\,G \mid x \in \mathbb{F}_q^k\ \}$$

Definition 3.8 (Kronecker product code)**.**

Let C_1 and C_2 be respectively $[n_1, k_1, d_1]$ and $[n_2, k_2, d_2]$ linear codes over $\mathbb{F}_q$ with generator matrices G_1 and G_2, the Kronecker product $C_1 \otimes C_2$ is an $[n_1 n_2, k_1 k_2, d_1 d_2]$ linear code over $\mathbb{F}_q$ [46] whose codewords consist of all $n_1 \times n_2$ matrices in which the columns are codewords of C_1 and the rows are codewords of C_2, such that the generator matrix is the Kronecker product $G_1 \otimes G_2$, see [38].

A linear code can also be characterized by an other matrix, the parity-check matrix.

Definition 3.9 (Parity-Check matrix)**.** *A parity-check matrix for a linear* $[\, n, k \,]$ *code is an* $(n-k) \times n$ *matrix* H *difined by;*

$$H\, c^{\top} = \begin{pmatrix} H_{11} & H_{12} & \dots & H_{1n} \\ H_{21} & H_{22} & \dots & H_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ H_{n-k\ 1} & H_{n-k\ 2} & \dots & H_{n-k\ n} \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix} = 0,\ \textit{for all}\ \ c = (\ c_1,\ c_2, \dots, c_n\ ) \in C$$

i.e., $C = \left\{\ x \in \mathbb{F}_q^n,\ H\, x^{\top} = 0\ \right\}$*, so any linear code is completely specified by a parity-check matrix.*

Note that any parity-check matrix for C must have $(n-k)$ rows, n columns, and rank $(n-k)$.

Remark 3.2. *Recall that, given a matrix* A*, we denote by* $A^{\top}$ *the transpose of* A*, the matrix wich is formed by turning all the rows of* A *into columns and vice-versa.*

Example 3.3.

$$A = \begin{pmatrix} 1 & 2 \end{pmatrix},\ A^{\top} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

$$A = \begin{pmatrix} 2 & 3 & 4 \\ 5 & 6 & 7 \end{pmatrix},\ A^{\top} = \begin{pmatrix} 2 & 5 \\ 3 & 6 \\ 4 & 7 \end{pmatrix}$$

Theorem 3.1. *Matrices* H *and* G *are parity-check and generator respectively for a linear code* C *if and only if*

i. The rows of a generator matrix are linearly independent;

ii. The rows of a parity-check matrix are linearly independent, and

iii. $H\ G^{\top} = 0$*.*

Definition 3.10.

i. A generator matrix of the form $G = [\, I \,|\, P \,]$ *where* I *is a* $k \times k$ *identity matrix and* P *is a* $k \times (n-k)$ *matrix, is said to be in standard form (also called systematic form of the generator matrix).*

ii. A parity-check matrix in the form $[\, B \,|\, I \,]$ *where* B *is a* $(n-k) \times k$ *matrix and* I *is a* $(n-k) \times (n-k)$ *identity matrix, is said to be in standard form.*

Theorem 3.2. *If* $G = [\, I \,|\, P \,]$ *is a generator matrix for the* $[n, k]$ *code* C *in standard form, then the parity-check matrix* H *will be defined as*

$$H = \left[-P^{\top} \,|\, I \,\right]$$

Since the choice of G *a generator matrix of a code is not unique, the parity-check matrix* H *will not be unique.*

3.2 Dual Codes

First we introduce the definition of inner product.

Definition 3.11 (Inner product and orthogonal)**.**

let $u = (\, u_1,\, u_2, \ldots, u_n \,)$, $v = (\, v_1,\, v_2, \ldots, v_n \,) \in \mathbb{F}_q^n$ *be two vectors. The inner product of* u, v *is denoted by* $\langle u, v \rangle$ *and is defined by*

$$\langle u, v \rangle = u_1\, v_1 + u_2\, v_2 + \ldots + u_n\, v_n = \sum_{i=1}^{n} u_i v_i$$

If $\langle u, v \rangle = 0$, then u and v are called orthogonal.

Example 3.4.

In $\mathbb{F}_2^4$; *we have*

$$\begin{aligned} \langle 1100,\, 1101 \rangle &= 1 + 1 + 0 + 0 = 0 \\ \langle 0111,\, 1111 \rangle &= 0 + 1 + 1 + 1 = 1 \end{aligned}$$

In $\mathbb{F}_3^4$;

$$\begin{aligned} \langle 1012,\, 2120 \rangle &= 2 + 0 + 2 + 0 = 1 \\ \langle 2122,\, 1210 \rangle &= 2 + 2 + 2 + 0 = 0 \end{aligned}$$

Lemma 3.2. *If* $u,\ v,\ w\ \in\ \mathbb{F}_q^n;\ \alpha,\ \beta\ \in\ \mathbb{F}_q$, *then*

1. $\langle u, v\rangle = \langle v, u\rangle$
2. $\langle \alpha\, u + \beta\, v, w\rangle = \alpha\ \langle u, w\rangle + \beta\ \langle v, w\rangle$

Definition 3.12 (Dual code)**.**

Let C *be an* $[\ n, k\ ]$ *code, then the dual code of* C, *denoted by* $C^{\perp}$ *is the set of all vectors in* $\mathbb{F}_q^n$ *which are orthogonal to every codeword of* C *(* $C^{\perp} \subset \mathbb{F}_q^n$*):*

$$C^{\perp} = \left\{\ v\ \in \mathbb{F}_q^n\,;\, \langle v, c\rangle = 0,\ \ for\ all\ codewords\ \ c\ \in\ C\right\}$$

Lemma 3.3. *Let* C *be an* $[\ n, k\ ]$ *code with generator matrix* G, *then*

$$C^{\perp} = \left\{\ v\ \in \mathbb{F}_q^n\,;\, G\, v^{\top} = 0\ \right\}$$

Theorem 3.3. *Let* C *be an* $[\ n, k\ ]$ *code, then the dual code* $C^{\perp}$ *of* C *is a linear* $[n, n-k]$ *code.*

Note that:

$$dim\, C\ +\ dim\, C^{\perp} = n$$

Example 3.5. *For the* $[\ n, 1\ ]$ *repetition code* C, *with the generator matrix*

$$G = (\,1,\,1,\,\ldots,\,1\,)$$

The dual code $C^{\perp}$ *is* $[\ n, n-1\ ]$ *code with the generator matrix* $G^{\perp}$*:*

$$G^{\perp} = \begin{pmatrix} 1 & 1 & 0 & 0 & \ldots & 0 \\ 1 & 0 & 1 & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & 0 & \ldots & 1 \end{pmatrix}$$

Proposition 3.2. *For any linear* $[\ n, k\ ]$ *code* C, $\left(C^{\perp}\right)^{\perp} = C$.

Proposition 3.3. *If* H, G *are a parity-check matrix and a generator matrix for* C *respectively, then they are a generator matrix and a parity-check matrix for* $C^{\perp}$ *respectively.*

Example 3.6. *Let G be a generator matrix of a code C in standard form over $\mathbb{F}_2$:*

$$G = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}$$

So we can obtain the parity-check matrix:

$$H = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

which is the generator matrix for the dual code $C^{\perp}$ of C.

3.2.1 Self-Dual Codes

The interaction between self-orthogonal codes and conjugacy classes of maximal abelian subgroups of a compact lie group was studied by J. A. Wood [52]. The classification of self dual codes is well known in some cases, [13], [23].

Definition 3.13.

A linear code is self-orthogonal if $C \subseteq C^{\perp}$.

A linear code is self-dual if $C = C^{\perp}$.

Theorem 3.4. *If C is a $[\, n, k, d \,]$ self-dual code over $\mathbb{F}_q$, then n must be even and the dimension is $n/2$, i.e., C is $[\, 2k, k, d \,]_q$ linear code for some k.*

3.3 MDS Codes

3.3.1 Singleton Bound

The simplest upper bound on code sizes is:

Theorem 3.5 (Singleton bound)**.**

Let C be a linear $[\, n, k, d \,]_q$ code, then the distance

$$d \leq n - k + 1$$

Proof. Let H be the parity-check matrix for the code, so $(d-1)$ columns of H are linearly independent, i.e., the column rank of $H \geq d-1$. But since the column rank equal to the row rank of H, and H has row rank equal to $(n-k)$, we obtain;

$$d \leq n-k+1$$

□

Definition 3.14. *A linear* $[\, n,k,d \,]_q$ *code such that*

$$d = n-k+1$$

is called a maximum distance separable (MDS) code.

Example 3.7 (Trivial families of MDS codes)**.**

The trivial MDS codes with parameters $[\, n,n,1 \,]_q$, $[\, n,1,n \,]_q$ *, and* $[\, n,n-1,2 \,]_q$ *have* k *non-zero weights with the exeception of the dual of the binary repetition code of length* $n > 2$ *which contains only words of even weights [20]. Several authors have studied MDS codes [25], [32].*

Weight Enumerator for a Linear Codes

The weight distribution of a code is important, it has a number of applications in the study of codes.

Definition 3.15. *let* A_i *be the numbers of codewords of weight* i *in a linear* $[\, n,k,d \,]$ *code* C*. The numbers* A_i, $i = 0, 1, \ldots, n$ *are called the weight distribution of the code* C*.*

Note that

$A_0 = 1.$

$A_1, \ldots, A_{d-1}$ *are all zero.*

A_d *is not zero.*

Definition 3.16. *The weight enumerator of* C *is given by:*

$$W_C(Z) = \sum_{i=0}^{n} A_i \, Z^i$$

Remark 3.3. *Another way to express the weight enumerator $W_C(Z)$ is:*

$$W_C(Z) = \sum_{c \in C} Z^{\,w(c)}$$

Example 3.8. *Let C be the code:*

$$C = \{\ 000,\ 101,\ 011,\ 110\ \}$$

Then:

$$W_C(\ Z\ ) = \sum_{i=0}^{3} A_i\ Z^i = \sum_{c \in C} Z^{\,w(c)} = 1 + 3\,Z^2$$

Definition 3.17. *The homogeneous weight enumerator of C is:*

$$W_C(\ X,\ Y\ )\ =\ \sum_{i=0}^{n} A_i\ X^{n-i}\ Y^i$$

Example 3.9. *The weight enumerator of $[\ 8,\ 4,\ 4\ ]$ hamming code is:*

$$W_C(\ X,\ Y\ )\ =\ \sum_{i=0}^{8} A_i\ X^{n-i}\ Y^i = X^8 + 14\,X^4\,Y^4 + Y^8$$

Remark 3.4. *Note that $W_C(\ X,\ Y\ )$ and $W_C(Z)$ are equivalent by the following equations:*

$$\begin{aligned} W_C(Z) &= W_C(1,\ Z) \\ W_C(\ X,\ Y\ ) &= X^n\ W_C\left(\ X^{-1}\ Y\ \right) \end{aligned}$$

Theorem 3.6 (MacWilliams)**.**

Let C be a $[\ n,\ k]$ linear code over $\mathbb{F}_q^n$, and $W_C(\ X,\ Y\ )$ be the weight enumerator of C. Then

$$W_{C^\perp}(\ X,\ Y\ ) = \frac{1}{|C|}\ W_C(\ X\ +\ (q-1)\ Y,\ X-\ Y\ ).$$

So by the MacWilliams Theorem we can determine the weight enumerator of a dual code from the weight enumerator of a code. This is very useful in practice, there are codes for which it is easier to compute the weight enumerator of one than the other.

Definition 3.18 (Minimal codeword)**.** *A non-zero codeword c_1 is minimal if it only covers $\mathbb{F}_q \cdot c_1$, i.e., if $\forall c_2 \in C$, $(supp(c_2) \subset supp(c_1)) \Rightarrow (c_1, c_2)$ linearly dependent.*

Definition 3.19 (Minimal linear code)**.** *A linear code C is minimal if every non-zero codeword $c \in C$ is minimal.*

There exists a sufficient condition for all non-zero codewords of a linear code to be minimal.

The following lemma gives sufficient condition on weight for a linear code to be minimal. More precisely, if the weights of a linear code are close enough to each other, then all non-zero codewords of the code are minimal, as described in the following statement.

Lemma 3.4 (Ashikhman-Barg Lemma [8])**.**

Let C be an $[n, k, d]$ code. Let w_{min} and w_{max} be the minimum and maximum non-zero weights, respectively. If

$$\frac{w_{min}}{w_{max}} > \frac{q-1}{q} \qquad (*)$$

Then C is minimal.

Remark 3.5. *Note that the converse is false (the previous condition is only necessary), for this we take the minimal tetracode $T[4, 2, 3]$ code over $\mathbb{F}_3$ with generator matrix*

$$G = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \end{pmatrix}$$

This code is also self dual. Its Kronecker square T^2, is a $[16, 4, 9]_3$ minimal code with generator matrix G^2, where $G^2 = G \otimes G$ is the Kronecker product of G by G.

Consider now the Kronecker product $T^4 = T^2 \otimes T^2$.

The code $T^4[256, 16, 81]$ is minimal [14], but it does not verify (*):

$$\frac{w_{min}}{w_{max}} \leqslant \frac{81}{144} < \frac{q-1}{q} = \frac{2}{3}.$$

Remark 3.6. *The Ashikhman-Barg Lemma is so useful in determining the minimal codewords.*

We finally define the notion of equivalence for linear codes:

Definition 3.20 (Equivalence of codes)**.** *Let C_1 and C_2 be two $[n, k]$ linear codes over $\mathbb{F}_q^n$, we say that C_1 and C_2 are equivalent, denoted by $C_1 \sim C_2$, if there exists a permutation $\tau \in S_n$, such that:*

$$C_2 = \tau(C_1) = \{(c_{\tau(1)}, c_{\tau(2)}, \ldots, c_{\tau(n)});\ (c_1, c_2, \ldots, c_n) \in C_1\}$$

If G_1 and G_2 are the generator matrices of two equivalent linear codes, then G_2 can be obtained from G_1 by permuting its columns.

Note that two equivalent linear codes have many of the same properties: Same dimenssion, minimum distance, weight distribution, etc. So equivalent linear codes have the same error correction capabilities.

3.4 Code-Based Cryptography

3.4.1 McEliece Cryptosystem

The most important property of the linear codes is their error correction capability. This property ensures that if errors are introduced for example during transmission into the codeword, so it is possible to decode correctly this codewords, if the numbers of errors is less than the correction capability.

Definition 3.21 (Error-correcting linear code)**.** *Let C be a linear $[n, k, d]$ code over $\mathbb{F}$ with a generator matrix G. We say that C can correct up to t errors, if there exists a decoding algorithm $D_{Alg} : \mathbb{F}^n \to C$ such that $\forall\ u \in \mathbb{F}^k,\ \forall\ e \in \mathbb{F}^n$, where the weight $w(e) \leq t$, the word:*

$$y \ = \ uG + e$$

is always correctly decoded as $D_{Alg}(y) \ = \ u$.

The error- correcting capability of a code C is directly related to its minimum distance.

Theorem 3.7. *Let C be a linear $[n, k, d]$ code over $\mathbb{F}$. There exists a decoding algorithm $D_{Alg} : \mathbb{F}^n \to C$ that correctly decodes codewords with up to $[\frac{(d-1)}{2}]$ erors.*

Thus, for every linear $[n, k, 2t+1]$ code, there exists a decoding algorithm that corrects up to t errors.

3.4.1.1 The Cryptosystem

The McEliece cryptosystem is based on error-correcting linear code, is one of the most successful cryptosystem based on notions of coding theory [39]. The original version of this system uses generator matrices and encodes the messages into codewords of Goppa codes. The main components of McEliece system are:

- Any $[n, k]_q$ code families for which efficient decoding algorithms are known.
- An efficient decoding algorithm: D_{Alg}.
- Number of errors: t.
- A private $(k \times n)$ linear block code generator matrix: G.
- A $(k \times k)$ secret non-singular scrambling matrix: S.
- A $(n \times n)$ secret permutation matrix: P.
- Plaintext message: m.
- Ciphertext: c.

Note that an $(n \times n)$ permutation matrix P is a binary matrix that has exactly one entry of 1 in each row and each column and the other elements are zeroes. If $(k \times n)$ matrix A is multiplied by P, the result is a matrix say $A' = AP$ which contains the same columns as A, but in different order. The McEliece scheme consists of three algorithms: A key generation algorithm which produces a public and a private key, an encryption algorithm and a decryption algorithm.

Key Generation:

- Choose a random $[n, k]_q$ code C, for which there exists an efficient decoding algorithm D_{Alg}, that can correct up to t errors.
- Compute a $(k \times n)$ generator matrix G for the code C.

- Generate a random $k \times k$ binary non-singular matrix S.
- Generate a random $(n \times n)$ permutation matrix P.
- Compute the $(k \times n)$ matrix $G' = SGP$.
- The public key is (G', t) and the private key is (S, G, P, D_{Alg}).

Encryption:

To encrypt a binary plaintext $m \in \{0,1\}^k$, generate a random vector $e \in \{0,1\}^n$ with hamming weight t and compute the ciphertext

$$c = mG' + e$$

Decryption:

For a received ciphertext c, first calculate:

$$cP^{-1} = (mS)G + eP^{-1}$$

Next use the decoding algorithm D_{Alg} to recover $c' = mS$ and calculate the message m with:

$$m = c'S^{-1}$$

4 | Contributions

Contents

In this chapter we present our main results, [34]:

4.1 Overview

In cryptography, secret sharing scheme refers to any method which distribute a secret among a group of participants individuals in such a way that only authorized subset of participants can reconstruct the secret by collectively combining their shares of the secret.

More formally, in secret sharing schemes there exist a dealer D, n participants $P_1, P_2, \ldots, P_n$ and a reconstructor R . In the secret sharing phase the dealer D splits a secret S into n parts, called shares, and sends privately one share t_i to each participant P_i, an access group is a subset of participants that are qualified to recover the secret S. In (t, n) threshold access group, $1 \leqslant t \leqslant n$, t or more than t players can recover the secret, while fewer than t players cannot know any information about the secret S.

Since the concept of secret scharing was introduced, several constructions have been proposed. In 1981 McEliece and Sarwate [40] first investigated the relation between linear codes and secret sharing. They observed that Shamir's scheme is closely related to Reed-Solomon codes. later many secret sharing schemes based on linear error correcting codes, have been proposed. In [28] Karnin et al. realize threshold schemes using linear codes. In [47] Renvall and Ding consider the access structures of secret sharing schemes based on MDS code. A similar construction is proposed in [19] by Ding et al. Massey [36], [37], firstly utilized linear codes for secret sharing schemes and pointed out the relationship between the access structure and the minimal codewords of the dual code. However, it is very hard in general to determine the minimal codewords of linear code. This was done only for a small number of classes of special linear codes.

Several authors have investigated the minimal codewords for several classes of codes and characterized the access structures of the schemes based on their dual codes [2], [7], [8], [47], [53].

4.1.1 A Link Between Secret Sharing Schemes and linear Codes

There are many methods to use linear codes to construct secret sharing schemes [36], [47]. One of them is the following.

Let C be a linear $[\ n,\ k,\ d]$ code with generator matrix $G\ =\ [\ g_0,\ g_1, \ldots, g_{n-1}]$, where $g_i\ \neq\ 0$ for $i\ =\ 0, 1, \ldots, n-1$, the secret S is an element of $\mathbb{F}_q$ and there are $(n-1)$ participants $P_1,\ P_2,\ \ldots,\ P_{n-1}$ and a dealer D are involved.

In order to determine the $n-1$ shares of the secret S, the dealer randomly choose an element $u\ =\ (\ u_0,\ u_1,\ \ldots,\ u_{k-1})\ \in \mathbb{F}_q^k$ such that $S\ =\ ug_0$. Note that u can be chosen in q^{k-1} ways. Then the dealer D treats u as an information vector and he computes the codeword corresponding to u as:

$$c\ =\ uG\ =\ (c_0,\ c_1,\ \ldots,\ c_{n-1}),\ c_i = ug_i$$

Then the dealer gives c_i to participant P_i as share for each $i \geqslant 1$.

Note that $c_0\ =\ ug_0\ =\ S$, then a subset of shares $\{\ c_{i_1},\ c_{i_2},\ \ldots,\ c_{i_m}\}$ determines the secret if and only if the column g_0 of the generating matrix G of the code C is a linear combination of $\ g_{i_1},\ g_{i_2},\ \ldots,\ g_{i_m}$.

4.2 Secret Sharing and Conjugacy Classes

In this section the secret sharing and conjugacy search problem are combined.

Let consider a secret sharing based on linear code C with generator matrix G.

Let us fix a random matrix $S_0 \in GL_k(\mathbb{F}_q)$ and define:

$$\begin{aligned} g:\ GL_k(\mathbb{F}_q) &\longrightarrow M_{k\times n}(\mathbb{F}_q) \\ S_i &\longmapsto g(S_i) = S_i^{S_0}\, G = G_i' \end{aligned}$$

Here $S_i^{S_0}$ stands $S_0\, S_i\, S_0^{-1}$.

Corollary 4.1. *For the map g, the following properties hold:*

1. $g(S_iS_j) = S_i^{S_0}\, g(S_j)$, *for all* $S_i, S_j \in GL_k(\mathbb{F}_q)$.

2. $g(S_i^n) = (S_i^{n-1})^{S_0}\, g(S_i)$*, for all* $S_i \in GL_k(\mathbb{F}_q)$.

Proof.

1.
$$\begin{aligned} g(S_iS_j) &= (S_iS_j)^{S_0}\, G \\ &= S_0\, S_i\, S_j\, S_0^{-1}\, G \\ &= S_0\, S_i\, S_0^{-1}\, S_0\, S_j\, S_0^{-1}\, G \\ &= S_i^{S_0}\, S_j^{S_0}\, G \\ &= S_i^{S_0}\, g(S_j). \end{aligned}$$

2.
$$\begin{aligned} g(S_i^n) &= g(S_i^{n-1}S_i) \\ &= (S_i^{n-1})^{S_0}\, g(S_i). \end{aligned}$$

□

Remark 4.1. *According to the last corollary, a variety of matrices are obtained from the matrix* G*. So we have a family of matrices that can be used to secret sharing. Then we can construct a sequence of dynamic (non-static) keys which can be dynamically and automatically updated, and which enhances the performance and the security of secret sharing.*

Proposition 4.1. *Let* C *be a minimal code with generator matrix* G*. Then the code* C_i' *with the generator matrix* $G_i' = g(S_i^n) = (S_i^n)^{S_0}\, G$ *is minimal.*

Proof.

We have: $G_i' = (S_i^n)^{S_0}\, G$, thus C and C' are equivalent codes, so they have the same weight distribution, then C' is minimal.

□

Proposition 4.2. *The access structure of the proposed code approach secret sharing depends only on the code, it does not depend on the choice of the generator matrix.*

Proof. We know that the access structure of secret-sharing schemes based on codes depends on the weight distribution of their dual codes, meaning that for equivalent codes where we have the same weight distribution, the access structure doesn't depend on the choice of the generator matrix but on the code. □

4.3 Characterization of MDS Codes Verifying the Property (*)

In the MDS codes, the notions of minimal and the property (*) coincide:

Proposition 4.3. *Let C be an $[n, k, d]_q$ MDS code, then C is minimal if and only if C verifies the property (*).*

Proof. It is obvious that C verifies the property (*) implies C is minimal.

Conversely; suppose that C is minimal, then from [14],

$$w_{max} \leqslant n - k + 1$$

As

$$w_{min} = n - k + 1 \leqslant w_{max}$$

So

$$\frac{w_{min}}{w_{max}} = 1 > \frac{q-1}{q}$$

The property (*) is verified. □

A linear code C is constant weight code if $w(x) = w(y)$ for all non-zero codewords $x, y \in C$.

As example of constant weight code we consider the simplex code $S_k(q)$ generated by $k \times [(q^k - 1)/(q - 1)]$ matrix G over $\mathbb{F}_q$, whose columns consist of one non-zero vector from each one-dimensional subspaces of $\mathbb{F}_q^k$.

According to the last Proposition:

Corollary 4.2. *Every minimal MDS code is constant weight code.*

Proof. According to the proof of the last Proposition, we have: $w_{min} = w_{max}$, then every codeword has the same weight. □

Theorem 4.1. *Let C be an $[n, k, d]_q$ MDS code, then C verifies the property (*) if and only if C is 2-dimensional q-ary simplex code with parameters $[q + 1, 2, q]_q$ or C is a $[n, 1, n]_q$ repetition code.*

Proof. If C is simplex code so it is constant weight code, the property (*) is verified. For the repetition code the result is obvious.
Conversely;

(i) Assume that $k \neq 1$;

If the MDS code C verifies the property (*), then C is constant weight code: $\forall c \in C - \{0\}; \quad w(c) = n - k + 1$, from [24] we deduce that C is equivalent to a padding of a replication of $[(q^k - 1)/(q - 1), k]$ simplex code $S_k(q)$ and each non-zero element of C has weight $mq^{(k-1)}$, where m is the multiplier of the replication, so $d = n - k + 1$ implies that $d = mq^{(k-1)} = [m((q^k - 1)/(q - 1)) + B] - k + 1$, where B is the number of zeros used in the padding [24].

Then,

$$k = mq^{(k-2)} + \ldots + mq + m + B + 1 \tag{4.1}$$

Which is true only for $k = 2$, $m = 1$ and $B = 0$, indeed if $k > 2$ we have

$$k = mq^{(k-2)} + \ldots + mq + m + B + 1 > mq^{(k-1)} - q + 2 \geqslant k$$

a contradiction. Hence $k = 2$, the equality (4.1) becomes: $2 = m + B + 1$, so $m = 1$ and $B = 0$, that is C is $[q + 1, 2, q]_q$ code.

(ii) If $k = 1$, the MDS code C is $[m, 1, m]_q$ code generated by $G = (g_1, \ldots, g_m)$. Every non-zero codeword of C equals $yG = (yg_1, \ldots, yg_m)$ where $y \in \mathbb{F}_q - \{0\}$, then $w(yG) = w(g_1, \ldots, g_m)$ and C is constant weight code with parameters $[m(q^k-1)/(q-1), k, mq^{(k-1)}]$ and $k = 1$, consequently C is $[m, 1, m]$ repetition code with $m = n$.

□

4.4 Characterization of Self-dual Codes Verifying the Property (*)

We distinguish two cases:

- Case A:

  $\mathbb{F}_q$ is a field of characteristic 2.

Theorem 4.2. *Let $C = [n, k, d]$ be a self-dual code over $\mathbb{F}_{2^f}$, C verifies the property (*) if and only if C is $[2, 1, 2]_q$ repetition code.*

To proof the Theorem 4.2, we will need the following Lemmas:

Lemma 4.1.

(i) A linear code $C = [n, k, d]$ over $\mathbb{F}_{2^f}$ is self-dual if and only if its standard generator matrix equals $G = [\, I \mid A \,]$, where the matrix A is orthogonal.

(ii) Let $C = [2k, k, d]$ be a self-dual code over $\mathbb{F}_{2^f}$, then C contains the codeword $1 = (1, 1, \ldots, 1)$.

Proof.

(i) As the length of a self-dual code satisfies $n = 2k$ where k is the dimension of the code, the matrix A must be square. Moreover, C will be a self-dual if G is also a parity-check matrix of the code, i.e., $G \cdot G^\top = 0$. But $G \cdot G^\top = I + A \cdot A^\top$, so that C is self-dual if $A \cdot A^\top = I$ ($\mathbb{F}_{2^f}$ is of characteristic 2 for which $I = -I$), i.e., A is orthogonal.

(ii) Let $G = [I_k : A]$ be a generator matrix of C, where I_k is the identity matrix of order k, let $A = [a_{(i,j)}]$, then $H = [A^\top : I_k]$ is a generator matrix of $C^\perp (C^\perp = C)$, we have: $A \cdot A^\top = I$.

Noting that for example for the first element of $A \cdot A^\top$:

$$a_{11}^2 + \cdots + a_{1k}^2 = (a_{11} + \cdots + a_{1k})^2 = 1$$

Then we have :

$$a_{11} + \cdots + a_{1k} = 1$$

For the matrix G, we denote its i-th row by L_i. It is easy to verify that:

$$L_1 + \cdots + L_k = (1, \ldots, 1)$$

Then

$$(1, \ldots, 1) \in C$$

□

Lemma 4.2. *Let $C = [2k, k, d]$ be a self-dual code over $\mathbb{F}_{2^f}$, if C verifies the property (*), then the code C is MDS code, i.e., $d = k + 1$.*

Proof. By Lemma 4.1 we have: $w_{max} = n$, so $\frac{d}{n} > \frac{2^f - 1}{2^f}$, it is easy to verify that $\frac{2^f - 1}{2^f} \geqslant \frac{1}{2}$ when $f \geqslant 1$, so we obtain $\frac{d}{n} > \frac{1}{2}$ which implies that $d > k$. On the other hand, $d \leqslant k + 1$ (the Singleton bound), so we obtain $d = k + 1$. □

Now we will prove Theorem 4.2.

Proof of Theorem 4.2. Suppose that $C = [2, 1, 2]_q$ is a repetition code over $\mathbb{F}_q$ of length 2, so we have:

$$\frac{w_{min}}{w_{max}} = 1 > \frac{q-1}{q}$$

Conversely; in the following, we consider two cases:

(i) Case 1:

Suppose $q = 2$, by Lemma 4.2 we have $d = k + 1$, this implies that $C = [2k, k, k + 1]$ is MDS code, so C meets the Singleton bound, and the only binary MDS codes are: $\mathbb{F}_2^n$ is $[n, n, 1]$, the $[n, n-1, 2]$ code and the repetition code $[n, 1, n]$;

(a) For $[n, n, 1]$ it is clear that it is not self-dual.

(b) For $[n, 1, n]$; we have $k = 1$, so $n = 2k = 2$.

(c) For $[n, n - 1, 2]$; we have $k = n - 1$, so $n = 2n - 2$ which implies that $n = 2$.

Hence, we have C is $[2, 1, 2]_2$ repetition code.

(ii) Case 2:

Suppose that $q > 2$, we have:

$$\frac{2^f - 1}{2^f} < \frac{w_{min}}{w_{max}} \leqslant \frac{k+1}{2k}$$

Therefore, we obtain:

$$\frac{k+1}{2k} > \frac{2^f - 1}{2^f}$$

Which implies that:

$$k < \frac{2^{f-1}}{2^{f-1}-1}$$

As $1 < f$, $k = 1$ and $n = 2$, C is $[2, 1, 2]_q$ repetition code.

This completes the proof. □

- Case B:

$\mathbb{F}_q$ is a field of arbitrary characteristic.

Theorem 4.3. *Let C be an $[n, k, d]_q$ self-dual code, then C verifies the property (*) if and only if C is $[2, 1, 2]$ repetition code over $\mathbb{F}_{2^f}$, or C is $[4, 2, 3]$ tetracode over $\mathbb{F}_3$, or C is $[2, 1, 2]$ repetition code over $\mathbb{F}_q$ if $q \equiv 1 \pmod 4$.*

Proof. It is easy to show that the tetracode and the repetition codes verifies the property (*).

Suppose now that C verifies the property (*);

(i) Case 1: if $k > 1$

According to [14] we have, $k + q - 2 \leqslant d \leqslant k + 1$, then $q \leqslant 3$:

(a) If $q = 2$; by Theorem 4.2: $k = 1$ and $n = 2$, so C is $[2, 1, 2]_2$ MDS repetition code.

(b) If $q = 3$; $k+1 \leqslant d \leqslant k+1$, then $d = k+1$, so C is $[2k, k, k+1]_3$ MDS code verifying (*), by Theorem 4.1 we deduce that C is $[4, 2, 3]_3$ tetracode.

(ii) Case 2: if $k = 1$

If q is a power of 2, so we have the $[2, 1, 2]$ repetition code over $\mathbb{F}_q$ with generator matrix $G = (1, 1)$. If q is not a power of 2, in this case $G = (1, a)$ with $a^2 = -1$, which is only possible if $q \equiv 1 \pmod 4$.

□

4.5 Special Codes

In this section, a special codes are examined and applied for secret sharing schemes. We studied the property (*) for a family of three and five weight cyclic codes.

4.5.1 Three Weight Codes

Let m and k be positive integers such that $m' = \frac{m}{e}$ is odd and $m' \geqslant 3$, with $e = gcd(m, k)$, let p be an odd prime. According to [55];

- When k is even and e is odd, there exists a three-weight p-ary cyclic code $C_{m,e}$ with parameters $[p^m - 1, 2m, p^m - p^{m-1} - \frac{p-1}{2}p^{(m+e-2)/2}]$.

- When k/e is odd, there exists a three-weight p-ary cyclic code $C'_{m,e}$ with parameters $[p^m - 1, 2m, p^m - p^{m-1} - (p-1)p^{(m+e-2)/2}]$.

Proposition 4.4. *The code $C_{m,e}$ verifies the property (*), then it's minimal code.*

Proof.

Put $a = p^m - p^{m-1}$ and $b = (\frac{p-1}{2})p^{(m+e-2)/2}$,

$C_{m,e}$ verifies (*) if and only if $\frac{a-b}{a+b} > \frac{p-1}{p} \Leftrightarrow a > b(2p-1) \Leftrightarrow p^{(m-e)/2} > p - \frac{1}{2}$.

Not that $m = em'$ with m' odd and $m' \geqslant 3$, so $\frac{m-e}{2} = \frac{e(m'-1)}{2} \geqslant e$,

hence $p^{(m-e)/2} > p - \frac{1}{2}$ and $C_{m,e}$ verifies (*). □

Proposition 4.5. *The code $C'_{m,e}$ verifies the property (*) if and only if $e \neq 1$ or $(e = 1$ and $m > 3)$.*

Proof.

Let $f = p^m - p^{m-1}$ and $g = (p-1)p^{(m+e-2)/2}$,

$C'_{m,e}$ verifies (*) $\Leftrightarrow f > g(2p-1) \Leftrightarrow p^{(m-e)/2} > 2p - 1$.

As below $\quad \dfrac{m-e}{2} = \dfrac{e(m'-1)}{2} \geqslant e,$

- If $e \neq 1$, the property (*) is verified.
- If $e = 1$, (*) is verified $\Leftrightarrow p^{(m-1)/2} > 2p-1 \Leftrightarrow \frac{m-1}{2} > 1 \Leftrightarrow m > 3.$

□

4.5.2 Five Weight Codes

Let m and k be positive integers such that $m' = \dfrac{m}{e}$ be odd and $m' \geqslant 5$, with $e = gcd(m,k)$, let p be an odd prime number.

According to [56], there exists a five-weight cyclic code $C_{(p,m,k)}$ over $\mathbb{F}_p$ with parameters $[p^m - 1, 3m, (p-1)(p^{m-1} - p)^{(m+3e-2)/2}]$.

Proposition 4.6. *The code $C_{(p,m,k)}$ verifies the property (*) if and only if $e \neq 1$ or $(e = 1 \quad and \quad m > 5)$.*

Proof.

From [56] we have $w_{min} = (p-1)(p^{m-1} - p^{(m+3e-2)/2})$ and $w_{max} = (p-1)(p^{m-1} + p^{(m+3e-2)/2})$.

Then $C_{(p,m,k)}$ verifies (*) if and only if $\quad p^{(m-3e)/2} > 2p-1 \Leftrightarrow m - 3e > 2.$

$m - 3e = (m'-3)e$, as $\quad m' \geqslant 5, \quad m - 3e \geqslant 2e.$

- If $e \neq 1$, the property (*) is verified.
- If $e = 1$, (*) is verified $\Leftrightarrow \quad m > 5e = 5.$

□

4.6 Application: Secret Sharing for Image Encryption

4.6.1 Example

To achieve a good security level, the encryption and sharing secret are combined, let I be an image of size N.

4.6.1.1 Arnold Discrete Cat Map

Particular codes are used for the sharing secret, in this example, the secrets to share are parameters of an image encryption scheme, more precisely $s_1 = a$, $s_2 = b$ determine the encryption matrix and $s_3 = r$ the number of iterations to be performed.

Indeed we will use Arnold's discrete cat map for encryption:

The Arnold's discrete cat map is a two-dimensional invertible chaotic

$$\Gamma : \quad \begin{pmatrix} x_n \\ y_n \end{pmatrix} \to \begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = A_0 \begin{pmatrix} x_n \\ y_n \end{pmatrix}$$

With $A_0 = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}$, it is used to shuffle the pixel positions of the plain image. The map Γ can be generalized to a map:

$$\Gamma_{a,b} : \quad \begin{pmatrix} x \\ y \end{pmatrix} \to \begin{pmatrix} x` \\ y` \end{pmatrix} = A_1 \begin{pmatrix} x \\ y \end{pmatrix}$$

Where $A_1 = \begin{pmatrix} 1 & b \\ a & ab+1 \end{pmatrix}$, a, b are positive integers, $\begin{pmatrix} x \\ y \end{pmatrix}$ is the original pixel position and $\begin{pmatrix} x` \\ y` \end{pmatrix}$ is the new pixel position after applying the map $\Gamma_{a,b}$.

After iterating this map r times, the correlation among the adjacent pixels can be disturbed, I becomes a random image, and we have $\begin{pmatrix} x` \\ y` \end{pmatrix} = A_1^r \begin{pmatrix} x \\ y \end{pmatrix}$.

The parameters $s_1 = a, s_2 = b, s_3 = r$ can be used as the secret keys.

Arnold encryption has periodicity which reduces its security, after several iterations, the encrypted image will be returned to the original image, hence it is important to add another processing to the algorithm in order to increase it's security, [1], [22].

We illustrate the proposed approach and we present the sharing process for the secrets a, b and r, the encryption and decryption techniques.

4.6.1.2 MDS Code Approach Secret Sharing

We will consider a secret sharing based on MDS code of Theorem 4.1. For this put $\mathbb{F}_q = \{0, \alpha_1, \ldots, \alpha_{q-1}\}$ the finite field of cardinality $q = p^m$ and

$$G = \begin{pmatrix} 1 & 0 & 1 & \cdots & 1 \\ 0 & 1 & \alpha_1 & \cdots & \alpha_{q-1} \end{pmatrix}$$

a $2 \times (q+1)$ generator matrix of a code C.

According to the Theorem 4.1, C is 2-dimensional simplex code over $\mathbb{F}_q$ and it is minimal, then $C^\perp$ is suitable for sharing secret, indeed if we represent G by its columns $G = (g_0, g_1, \ldots, g_q)$ and $G^\perp$ the generator matrix of $C^\perp$ by

$$G^\perp = (G_0, G_1, \ldots, G_q) = \begin{pmatrix} 1 & \alpha_1 & -1 & 0 & \ldots & 0 \\ 1 & \alpha_2 & 0 & -1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha_{q-1} & 0 & 0 & \ldots & -1 \end{pmatrix}$$

In the secret sharing scheme based on $C^\perp$, the secret $S \in \mathbb{F}_q$, q participants $p_1, p_2, \ldots, p_q$ and a dealer D are involved, the dealer chooses randomly a vector $u \in \mathbb{F}_q^{q-1}$ such that $S = uG_0$, puts $t_i = uG_i$ and $t = (t_0, \ldots, t_q)$ and gives t_i to participants P_i, $1 \leqslant i \leqslant q$ as share.

For the reconstruction phase, according to [54], if G_0 is a linear combination of $G_{i_1}, \ldots, G_{i_m}$:

$$G_0 = \sum_{j=1}^{m} \alpha_j G_{i_j}.$$

For at least $\alpha_j \neq 0$, then

$$S = \sum_{j=1}^{m} \alpha_j t_{i_j},$$

As rank $(G^{\perp}) = q-1$ and $C^{\perp}$ is MDS, G_0 with any $q-1$ columns are dependent, so the cardinality of any access group equals $q-1$.

For example if we take $q = p$ (p prime), $(P_1, P_2, \ldots, P_{p-1})$ is an access group, because

$$G_0 = \sum_{k=1}^{p-1} kG_{p-k},$$

So

$$S = \sum_{k=1}^{p-1} kt_{p-k}.$$

For tests we take $p = 13$, we obtain:

- $s_1 = a = 1 \cdot 36 + 2 \cdot 12 + 3 \cdot 24 + \cdots + 12 \cdot 177 = 1 \pmod{13}$
- $s_2 = b = 1 \cdot 24 + 2 \cdot 0 + 3 \cdot 24 + \cdots + 12 \cdot 104 = 3 \pmod{13}$
- $s_3 = r = 1 \cdot 36 + 2 \cdot 12 + 3 \cdot 24 + \cdots + 12 \cdot 158 = 12 \pmod{13}$

Then the used parameters are summarized in Table 1.

Table 4.1: The used parameters in the example.

Vector u_i	Secret	Shares t	
(1,0,2,4,0,6,7,1,2,1,3,0)	$s_1 = a = 1$	(177,12,0,24,48,0 ,72,84,12,24,12,36,0)	(mod 13)
(1,0,2,1,0,4,3,1,2,0,2,0)	$s_2 = b = 3$	(104,12,0,24,12,0 ,48,36,12,24,0 ,24,0)	(mod 13)
(2,0,2,4,2,1,7,1,2,1,3,0)	$s_3 = r = 12$	(158,24,0,24,48,24,12,84,12,24,12,36,0)	(mod 13)

For the reveal phase we take for example the access group $(P_1, P_2, \ldots, P_{12})$ and the secret is calculated by

$$s_i = \sum_{k=1}^{12} kt_{13-k}$$

Remark 4.2.

1. *The previous example can be used as secret sharing scheme with cheater detection, indeed when participants present their shares in the reconstruction phase, malicious participant (cheater) presents faked share, as* $t = (t_0, \ldots, t_q)$ *belongs to a* $[n, k, d]$ *code* $C^{\perp}$ *we make use of detection of errors (modified shares) up to* $d - 1$*. Thus this scheme offers verifiable mechanism to prevent* 2 *fraudulent participants. For the case of more than two dishonest participants we can use the codes* $C_{m,e}$*,* $C'_{m,e}$ *and* $C_{(p,m,k)}$*, because their minimum distance depend on the parameters* m, e, p, k*.*

2. *As mentioned in 4.2 we can use conjugacy classes to enhances the performance of secret sharing.*

4.7 Conjugacy Classes and Key Exchange

This section proposes a key agreement protocol. This protocol is based on the general idea of Diffie- Hellman key exchange algorithm [18] and A. Y. Mahmoud [33]. The proposed protocol is also based on few matrix multiplications, using commutative matrices and conjugation.

4.7.1 Initial Setup:

Let $GL(n, \mathbb{F})$ be the set of invertible $n \times n$ matrices with entries in $\mathbb{F}_q$. Let $GN(n, \mathbb{F})$ be the set of $n \times n$ matrices with entries in $\mathbb{F}_q$, with zero determinant value and having rank $n - 1$.

We denote by: A (Alice), B (Bob) two participants who share the secret, in our protocol we use the following symbols:

Let $M \in GL(n, \mathbb{F})$, $G \in GN(n, \mathbb{F})$, be two matrices publicly known.

$$
\begin{aligned}
r_A \in \mathbb{N}, X_A = MD_AM^{-1} (D_A \in GL(n, \mathbb{F}) \, is \, a \, diagonal \, matrix) &: A's \, private \, key \, pair \\
Y_A = X_A G^{r_A} X_A^{-1} \, (X_A G \neq G X_A) &: A's \, public \, key \\
r_B \in \mathbb{N}, X_B = MD_BM^{-1} (D_B \in GL(n, \mathbb{F}) \, is \, a \, diagonal \, matrix) &: B's \, private \, key \, pair \\
Y_B = X_B G^{r_B} X_B^{-1} \, (X_B G \neq G X_B) &: B's \, public \, key
\end{aligned}
$$

It is easy to see that:

$$\begin{aligned} X_A X_B &= M D_A M^{-1} M D_B M^{-1} \\ &= M D_A D_B M^{-1} \\ &= M D_B D_A M^{-1} \\ &= M D_B M^{-1} M D_A M^{-1} \\ &= X_B X_A. \end{aligned}$$

since diagonal matrices commute.

By using the following protocol, the two parties A and B can obtain the same shared key, k, without transference of k.

4.7.2 The Protocol

1. User A sends his public key to B and B his public key to A.
2. A computes the common secret key k_A using his private key pair and the received public key of B: $k_A = X_A Y_B^{r_A} X_A^{-1}$
3. B computes the key k_B: $k_B = X_B Y_A^{r_B} X_B^{-1}$

At the end A and B share the same secret key $k = k_A = k_B$.

Since $X_A X_B = X_B X_A$, then

$$\begin{aligned} k_A &= X_A Y_B^{r_A} X_A^{-1} \\ &= X_A X_B G^{r_B r_A} X_B^{-1} X_A^{-1} \\ &= X_B X_A G^{r_A r_B} X_A^{-1} X_B^{-1} \\ &= X_B Y_A^{r_B} X_B^{-1} \\ &= k_B. \end{aligned}$$

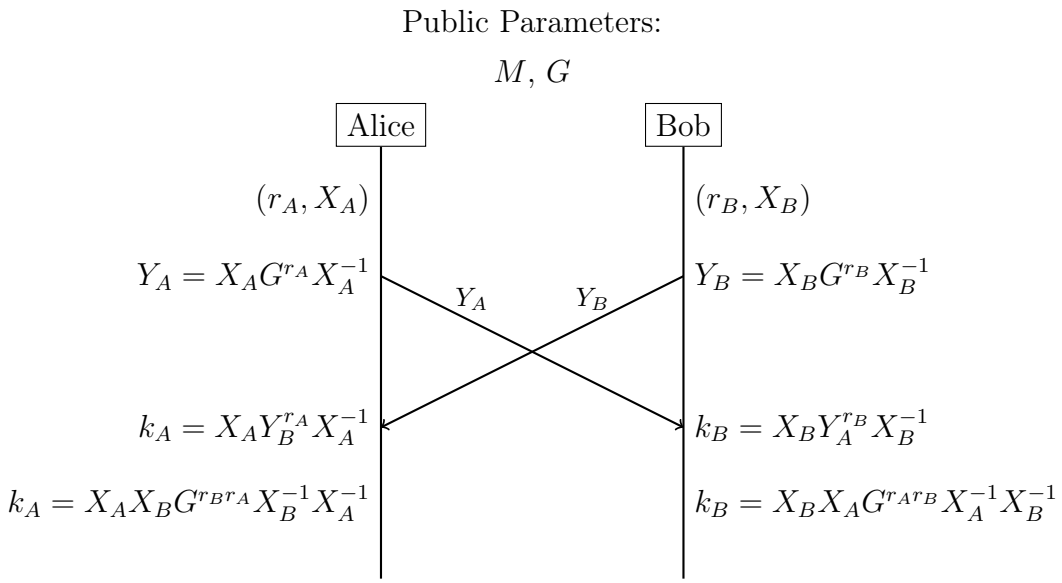
Figure 4.1 shows the agreement process.

As mentioned, our protocol is similar to [33]. Thus, our protocol is secure as [33]. For more details see [33].

According to [21], the complexity of $n \times n$ matrix multiplication is $O(n^{2.37286})$, this mean the complexity of our protocol, is $O(n^{2.37286})$.

Note that in our protocol:

- if $|G| \neq 0$, a possible eavesdropper Eve ($\mathbf{E}$) could solve the discrete logarithm problem by considering the determinantal equation:



$$k_A = X_A X_B G^{r_B r_A} X_B^{-1} X_A^{-1} = X_B X_A G^{r_A r_B} X_A^{-1} X_B^{-1} = k_B$$

Figure 4.1: Key agreement process using conjugate

$$\begin{aligned} |Y_A| &= \left|X_A\, G^{r_A} X_A^{-1}\right| \\ &= |X_A|\ |G^{r_A}|\ \left|X_A^{-1}\right| \\ &= |X_A|\ |G|^{r_A}\ |X_A|^{-1} \\ &= |G|^{r_A}\,. \end{aligned}$$

- In order to recover the private keys (e.g., X_A and r_A), **E** must be able to solve the following equation:

$$Y_A\, X_A = X_A\, G^{r_A}$$

but this is difficult because both X_A and G^{r_A} are not known.

4.7.3 Example

Let $n = 2$, $\mathbb{F} = \mathbb{Z}_7 = \{0, 1, \ldots, 5, 6\}$ and let:

$$r_1 = 2, r_2 = 3, M = \begin{pmatrix} 1 & 2 \\ 5 & 6 \end{pmatrix}, M^{-1} = \begin{pmatrix} 2 & 4 \\ 3 & 5 \end{pmatrix}, G = \begin{pmatrix} 1 & 6 \\ 3 & 4 \end{pmatrix},$$

$$D_A = \begin{pmatrix} d_1 & 0 \\ 0 & d_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} and\, D_B = \begin{pmatrix} d_3 & 0 \\ 0 & d_4 \end{pmatrix} = \begin{pmatrix} 2 & 0 \\ 0 & 3 \end{pmatrix}$$

Then:

$$X_A = MD_AM^{-1} = \begin{pmatrix} 1 & 2 \\ 5 & 6 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} 2 & 4 \\ 3 & 5 \end{pmatrix} = \begin{pmatrix} 0 & 3 \\ 4 & 3 \end{pmatrix}$$

$$X_A = \begin{pmatrix} 0 & 3 \\ 4 & 3 \end{pmatrix} \Rightarrow X_A^{-1} = \begin{pmatrix} 5 & 2 \\ 5 & 0 \end{pmatrix}$$

$$X_B = MD_BM^{-1} = \begin{pmatrix} 1 & 2 \\ 5 & 6 \end{pmatrix} \begin{pmatrix} 2 & 0 \\ 0 & 3 \end{pmatrix} \begin{pmatrix} 2 & 4 \\ 3 & 5 \end{pmatrix} = \begin{pmatrix} 1 & 3 \\ 4 & 4 \end{pmatrix}$$

$$X_B = \begin{pmatrix} 1 & 3 \\ 4 & 4 \end{pmatrix} \Rightarrow X_B^{-1} = \begin{pmatrix} 3 & 3 \\ 4 & 6 \end{pmatrix}$$

And the user's public keys are:

$$Y_A = \begin{pmatrix} 0 & 3 \\ 4 & 3 \end{pmatrix} \begin{pmatrix} 1 & 6 \\ 3 & 4 \end{pmatrix}^2 \begin{pmatrix} 5 & 2 \\ 5 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 6 \\ 0 & 4 \end{pmatrix}$$

$$Y_B = \begin{pmatrix} 1 & 3 \\ 4 & 4 \end{pmatrix} \begin{pmatrix} 1 & 6 \\ 3 & 4 \end{pmatrix}^3 \begin{pmatrix} 3 & 3 \\ 4 & 6 \end{pmatrix} = \begin{pmatrix} 2 & 6 \\ 6 & 4 \end{pmatrix}$$

Finally the secret shared key is:

$$k_A = X_A Y_B^{r_A} X_A^{-1} = \begin{pmatrix} 0 & 3 \\ 4 & 3 \end{pmatrix} \begin{pmatrix} 2 & 6 \\ 6 & 4 \end{pmatrix}^2 \begin{pmatrix} 5 & 2 \\ 5 & 0 \end{pmatrix} = \begin{pmatrix} 4 & 6 \\ 5 & 4 \end{pmatrix}$$

$$k_B = X_B Y_A^{r_B} X_B^{-1} = \begin{pmatrix} 1 & 3 \\ 4 & 4 \end{pmatrix} \begin{pmatrix} 0 & 6 \\ 0 & 4 \end{pmatrix}^3 \begin{pmatrix} 3 & 3 \\ 4 & 6 \end{pmatrix} = \begin{pmatrix} 4 & 6 \\ 5 & 4 \end{pmatrix}$$

Thus, $k = k_A = k_B$, $|k| = 0$ and $rank(k) = 0$.

4.8 Conjugacy Classes and McEliece

In this section, we use the conjugacy search problem in McEliece cryptosystem. We present a modification which increases the security of the system without increasing the key size. We show how the conjugation classes can be used to obtain a dynamic (non-static) keys.

4.8.1 The Proposed Cryptosystem

The proposed cryptosystem is similar to the McEliece cryptosystem, so there are also three steps: Key generation (a public and a private key), encryption and decryption.

Key Generation:

- Choose a random $[n, k]_q$ code C, for which there exists an efficient decoding algorithm D_{Alg}, that can correct up to t errors, and Compute a $(k \times n)$ generator matrix G for the code C.
- Chooses a random matrices $S \in GL_k(\mathbb{F}_q)$ and a $(n \times n)$ permutation matrix P.
- Let us fix a random matrix $S_0 \in GL_k(\mathbb{F}_q)$ and define $h : GL_k(\mathbb{F}_q) \longrightarrow M_{k\times n}(\mathbb{F}_q)$ by: $h(S_i) = S\, S_i^{S_0}\, G\, P = G'_i$. Here $S_i^{S_0}$ stands $S_0\, S_i\, S_0^{-1}$.
- The public key is (G'_i, t), and the private key is $(S\, S_i^{S_0}, G, P)$.

Encryption:

To encrypt a message m, generate a random vector e of weight t and compute

$$c = m\, G'_i + e$$

Decryption:

To decrypt the ciphertext c, first we compute:

$$cP^{-1} \;=\; (m\, S\, S_i^{S_0})\, G + eP^{-1}$$

And then recovers $c' = m\, S\, S_i^{S_0}$ by using the decoding algorithm, then compute:

$$m = c'\,(S\, S_i^{S_0})^{-1}$$

Corollary 4.3. *For the map h, the following properties hold:*

1. $h(S_iS_j) = S_i^{SS_0}\, h(S_j)$, *for all* $S_i, S_j \in GL_k(\mathbb{F}_q)$.
2. $h(S_i^n) = (S_i^{n-1})^{SS_0}\, h(S_i)$, *for all* $S_i \in GL_k(\mathbb{F}_q)$.

Proof.

1.
$$\begin{aligned} h(S_iS_j) &= S\,(S_iS_j)^{S_0}\, G\, P \\ &= S\, S_0\, S_i\, S_j\, S_0^{-1}\, G\, P \\ &= S\, S_0\, S_i\, S_0^{-1}\, S_0\, S_j\, S_0^{-1}\, G\, P \\ &= S\, S_i^{S_0}\, S^{-1}\, S\, S_j^{S_0}\, G\, P \\ &= S_i^{S\,S_0}\, h(S_j). \end{aligned}$$

2.
$$\begin{aligned} h(S_i^n) &= h(S_i^{n-1}S_i) \\ &= (S_i^{n-1})^{S\,S_0}\, h(S_i). \end{aligned}$$

□

Remark 4.3.

1. *The dynamic key generation scheme is based on the synchronous time between the sender and the receiver, the dynamic key has a short life time. For example the matrix S_i is valid with in that time period i. S_0 is an initial parameter.*
2. *According to the last corollary, a sequence of non-static cryptography keys is generated. Hence, in the proposed method every message in the system is encrypted by a different cryptography keys, which enhances the performance and the security of cryptographic systems.*

Bibliography

[1] A. O. Abdul-Majeed, Chaotic Scheme for Image Encryption Based on Arnold Cat's Map, *Int. J. Comput. Sci. Inf Security* **12** (3) (2014) 26.

[2] R. Anderson, C. Ding, T. Helleseth and T. Klove, How to build robust shared control systems, *Des. Codes Crypt.* **15** (2) (1998) 111–124.

[3] I. Anshel, M. Anshel, and D. Goldfeld, An algebraic method for public-key cryptography, *Math. Res. Lett* **6** (1999) 287–292.

[4] I. Anshel, M. Anshel, B. Fisher and D. Goldfeld, New key agreement protocols in braid group cryptography, CT-RSA 2001 (San Francisco), *Springer Lect. Notes in Comp. Sci.* **2020** (2001) 13–27.

[5] I. Anshel, M. Anshel, D. Goldfeld, and S. Lemieux, Key Agreement, The Algebraic EraserTM , and Lightweight Cryptography. *Algebraic Methods in Cryptography, Contemp. Math.* **418** (2006) 1–34.

[6] E. Artin, Theory of braids, *Annals of Mat h* **2** (48) (1947) 101–126.

[7] A. Ashikhmin, A. Barg, G. Cohen, L. Huguet, Variations on minimal codewords in linear codes, *International Symposium on Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes, Springer* (1995) 96–105.

[8] A. Ashikhmin and A. Barg, Minimal vectors in linear codes, *IEEE Trans. Inform. Theory* **44** (5) (1998) 2010–2017.

[9] R. Baer, Finiteness properties of groups, *Duke Math. J.* **15** (4) (1948) 1021–1032.

[10] G. R. Blakley, Safeguarding cryptographic keys, in *Proceedings of the national computer conference* **48** (1979) 313–317.

[11] J. Calais, Eléments de théorie des groupes, *Presses Universitaires de France* (1984) QA174. 2C25.

[12] A. Camina, R. Camina, The influence of conjugacy class sizes on the structure of finite groups: a survey, *Asian-Eur. J. Math.* **4** (4) (2011) 559–588.

[13] W. Cary Huffman, On the classification and enumeration of self-dual codes, *Finite Fields Appl* **11** (3) (2005) 451–490.

[14] G. D. Cohen, S. Mesnager and A. Patey, On minimal and quasi-minimal linear codes, in *IMA International Conference on Cryptography and Coding, Springer* (2013) 85–98.

[15] M. Dehn, Über unendliche diskontinuierliche Gruppen, *Math. Ann.* **71** (1) (1911) 116–144.

[16] P. Dehornoy, Braid-Based Cryptography, *Cont. Math.* **360** (2004) 5–34.

[17] P. Dehornoy, Using shifted conjugacy in braid-based cryptography, *Algebraic Methods in Cryptography, Contemp. Math.* **418** (2006) 65–74.

[18] W. Diffie and M. Hellman, New directions in cryptography, *IEEE trans. Inform. Theory* **22** (6) (1976) 644–654.

[19] C. Ding, T. Laihonen and A. Renvall, Linear multisecret-sharing schemes and error-correcting codes, *J. Universal Comput. Sci.* **3** (9) (1997) 1023–1036.

[20] M. F. Ezerman, M. Grassl and P. Solé, The weights in MDS codes, *IEEE Trans. Inform. Theory* **57** (1) (2011) 392–396.

[21] F. L. Gall, Powers of tensors and fast matrix multiplication, in *Proceedings of the 39th international symposium on symbolic and algebraic computation. ACM* 2014 296–303.

[22] Z. H. Guan, F. Huang and W. Guan, Chaos-based image encryption algorithm, *Physics Letters A* **346** (1) (2005) 153–157.

[23] M. Harada and A. Munemasa, A complete classification of ternary self-dual codes of length 24, *J. Combin. Theory Ser A* **116** (5) (2009) 1063–1072.

[24] D. Hoffman, Linear codes and weights, *Australas. J. Combin* **7** (1993) 37–45.

[25] W. C. Huffman and V. Pless, Fundamentals of error-correcting codes, *Cambridge university press* (2010).

[26] J. F. Humphreys, A course in group theory, *Oxford University Press on Demand* **6** (1996).

[27] S. Joan Birman, Braids, links and mapping class groups, *Ann. Math. Studies, Princeton Univ. Press* **82** (1974).

[28] E. Karnin, J. Greene, and M. Hellman, On secret sharing systems, *IEEE Trans. Inform. Theory* **29** (1) (1983) 35–41.

[29] T. Keller, Fixed conjugacy classes of normal subgroups and the k(GV)-problem, *J. Algebra* **305** (1) (2006) 457–486.

[30] K. H. Ko, S. J. Lee, J. H. Cheon, J. W. Han, J. Kang, and C. Park, New public-key cryptosystem using braid groups, in *Annual International Cryptology Conference, Springer, Berlin, Heidelberg* (2000) 166–183.

[31] K. H. Ko, D. H. Choi, M. S. Cho and J. W. Han, New signature scheme using Conjugality problem, *Cryptology eprint Archieve Report* (2002) (http://eprint.iacr.org/2002/168).

[32] J. I. Kokkala, D. S. Krotov and P. R. Ostergard, Classification of MDS Codes over Small Alphabets, in *Coding Theory and Applications, Springer* (2015) 227–235.

[33] A. Y. Mahmoud and A. G. Chefranov, Secure Hill cipher modifications and key exchange protocol, in *Automation Quality and Testing Robotics (AQTR) IEEE International Conference* **2** 2010 1–6.

[34] S. Makhlouf and L. Noui, Characterization of some minimal codes for secret sharing, *Asian-European Journal of Mathematics* **12** (2) 2018, in press.

[35] F. Manganiello, A. L. Trautmann and J. Rosenthal, On conjugacy classes of subgroups of the general linear group and cyclic orbit codes, in

IEEE International Symposium on Information Theory Proceedings (ISIT) (2011) 1916–1920.

[36] J. L. Massey, Minimal codewords and secret sharing, in *Proceedings of the 6th joint Swedish-Russian international workshop on information theory* (1993) 276–279.

[37] J. L. Massey, Some applications of coding theory in cryptography, *Codes and Ciphers: Cryptography and Coding IV* (1995) 33–47.

[38] F. J. MacWilliams and N. J. A. Sloane, The theory of error-correcting codes, *Elsevier* (1977).

[39] R. J. McEliece, A public-key cryptosystem based on algebraic coding theory, *Deep Space Network Progress Report* 44 (1978) 114–116 Jan.

[40] R. J. McEliece and V. D . Sarwate, On sharing secrets and Reed-Solomon codes, *Communications of the ACM* **24** (9) (1981) 583–584.

[41] H. R. Morton, The multivariable Alexander polynomial for a closed braid, *Contemp. Math.* **233** (1999) 167–172.

[42] K. Murasugi and B. Kurpita, A Study of Braids, *Springer Science & Business Media* **484** (1999).

[43] B. Neumann, Groups covered by permutable subsets, *J. London Math. Soc.* **1** (2) (1954) 236–248.

[44] S. H. Paeng, K. C. Ha, J. H. Kim, S. Chee and C. Park, New Public Key Cryptosystem Using Finite Non Abelian Groups, *Proc. of Crypto 2001, LNCS 2139, Springer-Verlag* (2001) 470–485.

[45] S. H. Paeng, D. Kwon, K. C. Ha and J. H. Kim, Improved public key cryptosystem using finite non abelian groups, *IACR EPrint* (2001).

[46] D. M. Rankin and T. V. Gulliver, Single parity check product codes, *IEEE Trans. Commun.* **49**(8) (2001) 1354–1362.

[47] Renvall, Ari, and C. Ding, The access structure of some secret-sharing schemes, in *Australasian Conference on Information Security and Privacy, Springer, Berlin, Heidelberg* (1996) 67–78.

[48] A. Shamir, How to share a secret, *Communications of the ACM* **22** (11) (1979) 612–613.

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

[50] T. Thomas and A. K. Lal, A zero-knowledge undeniable signature scheme in non-abelian group setting, *I. J. Network Security* **6** (3) (2008) 265–269.

[51] N. R. Wagner and R. M . Magyarik, A public key cryptosystem based on the word problem, *Lecture Notes in Computer Science* **196** (1985) 19–36.

[52] J. A. Wood, Self-orthogonal codes and the topology of spinor groups, in *Coding Theory and Design Theory, Springer* (1990) 219–239.

[53] J. Yuan and C. Ding, Secret sharing schemes from two-weight codes, *Electronic Notes in Discrete Mathematics* **15** (2003) 232.

[54] J. Yuan and C. Ding, Secret sharing schemes from three classes of linear codes, *IEEE Trans. Inform. Theory* **52** (1) (2006) 206–212.

[55] Z. Zhou and C. Ding, A class of three-weight cyclic codes, *Finite Fields Appl* **25** (2014) 79–93.

[56] Z. Zhou, C. Ding, J. Luo and A. Zhang, A family of five-weight cyclic codes and their weight enumerators, *IEEE Trans. Inform. Theory* **59** (10) (2013) 6674–6682.

Abstrat

Conjugation is an important action both on the elements and the subgroups of a group G. The concept of the conjugacy plays a central role in representation theory and in some applications.

Several authors have studied the correspondence between special linear codes and conjugacy classes of subgroups of specific groups. In this thesis, after a brief survey of conjugacy classes, we focus on applications. We study the conjugacy classes in cryptography. We give characterization of subclass of self-orthogonal codes and particular codes suitable for secret sharing. A part of this dissertation is devoted to the study of the practical applications to secret sharing, encryption and key exchange.

keywords: Conjugacy class, Secret sharing scheme; Self-dual codes; MDS codes; Minimal codeword; Self-orthogonal code.

Résumé

La conjugaison est une action importante à la fois sur les éléments et les sous groupes d'un groupe G. Le concept de conjugaison joue un rôle central dans la théorie des représentations des groupes et dans certaines applications.

Plusieurs auteurs ont étudié la correspondance entre quelques classes de codes linéaires spéciaux et les classes de conjugaison de sous groupes de groupes spécifiques. Dans cette thèse, aprés un bref aperçu sur les classes de conjugaison, nous étudions les classes de conjugaison en cryptographie. Nous donnons une caractérisation de sous classe de codes auto-orthogonaux et de codes particuliers convenables au partage de secrets. Une partie de cette thèse est consacrée aux applications pratiques au partage de secrets, cryptage et l'échange de clés.

Mots clés: Classe de conjugaison, Schéma de partage de secret; Codes auto-duaux; Codes MDS; Mot de code minimal; Code auto-orthogonal.