

THE DISCRETE LOGARITHM PROBLEM
IN NON-ABELIAN GROUPS

by
Ivana Ilić

A Dissertation Submitted to the Faculty of
The Charles E. Schmidt College of Science
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy

Florida Atlantic University
Boca Raton, Florida
December 2010

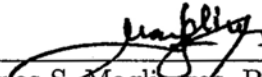
THE DISCRETE LOGARITHM PROBLEM
IN NON-ABELIAN GROUPS

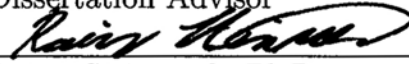
by

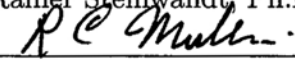
Ivana Ilić

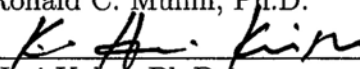
This dissertation was prepared under the direction of the candidate's dissertation advisor, Dr. Spyros S. Magliveras, Department of Mathematical Sciences, and it has been approved by the members of her supervisory committee. It was submitted to the faculty of the Charles E. Schmidt College of Science and was accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

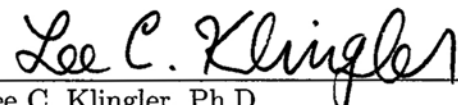
SUPERVISORY COMMITTEE:

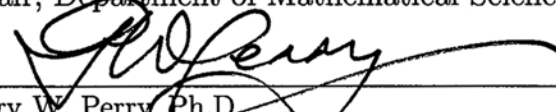

Spyros S. Magliveras, Ph.D.
Dissertation Advisor

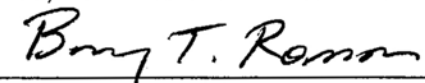

Rainer Steinwandt, Ph.D.

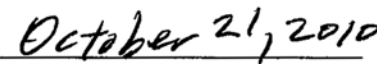

Ronald C. Mullin, Ph.D.


Hari Kalva, Ph.D.


Lee C. Klingler, Ph.D.
Chair, Department of Mathematical Sciences


Gary W. Perry, Ph.D.
Dean, The Charles E. Schmidt College of Science


Barry T. Rosson, Ph.D.
Dean, Graduate College


Date

ACKNOWLEDGEMENTS

I would like to thank my dissertation advisor, Dr. Spyros S. Magliveras, for the academic guidance, teaching and directions in the scientific research during the dissertation preparation and my graduate studies. I deeply appreciate what he has taught me.

I would also like to thank members of my supervisory committee: Dr. Spyros S. Magliveras, Dr. Rainer Steinwandt, Dr. Ronald C. Mullin and Dr. Hari Kalva for the dissertation supervision, academic advising and continuous support.

Finally, I would like to acknowledge The Department of Mathematical Sciences, Charles E. Schmidt College of Science, Florida Atlantic University and its faculty for the excellent graduate program in Mathematics and outstanding environment for graduate studies and research.

ABSTRACT

Author: Ivana Ilić

Title: The Discrete Logarithm Problem in Non-abelian Groups

Institution: Florida Atlantic University

Dissertation Advisor: Dr. Spyros S. Magliveras

Degree: Doctor of Philosophy

Year: 2010

This dissertation contains results of the candidate's research on the generalized discrete logarithm problem (GDLP) and its applications to cryptology, in non-abelian groups. The projective special linear groups $PSL(2, p)$, where p is a prime, represented by matrices over the field of order p , are investigated as potential candidates for implementation of the GDLP. Our results show that the GDLP with respect to specific pairs of $PSL(2, p)$ generators is weak. In such cases the groups $PSL(2, p)$ are not good candidates for cryptographic applications which rely on the hardness of the GDLP. Results are presented on generalizing existing cryptographic primitives and protocols based on the hardness of the GDLP in non-abelian groups. A special instance of a cryptographic primitive defined over the groups $SL(2, 2^n)$, the Tillich-Zémor hash function, has been cryptanalyzed. In particular, an algorithm for constructing collisions of short length for any input parameter is presented. A series of mathematical results are developed to support the algorithm and to prove existence of short collisions.

Contents

1	Introduction	1
2	Preliminaries	4
2.1	Group theory	4
2.1.1	Group Actions	4
2.1.2	Basis theorem for transitive permutation representations	8
2.1.3	Special linear group and projective special linear group	9
2.2	Cryptography	14
2.2.1	Traditional discrete logarithm problem	15
2.2.2	Algorithms for computing discrete logarithms	16
2.2.3	Diffie-Hellman problems and key exchange protocol	20
2.2.4	ElGamal cryptosystem	22
2.2.5	The Conjugacy problem	23
2.2.6	Cryptographic hash functions	23
2.2.7	Generalized discrete logarithm problem in finite groups	24
3	Weak generalized discrete logarithms	26
3.1	Weak GDLP in $PSL(2, p)$ with respect to two specific generators	26
3.2	Weak GDLP in $PSL(2, p)$ with respect to any two generators of order p	33
3.3	Weak GDLP in $PSL(2, p)$ with respect to two generators one of which is of order p	37

3.4	GDLP in $PSL(2, p)$ with respect to two generators none of which is of order p	39
3.4.1	A strategy for attacking GDLP in $PSL(2, p)$	40
3.4.2	The p-attack and its analysis	41
3.4.3	Analysis of special cases	45
3.5	Relations in the context of cryptography	47
4	Cryptographic primitives based on the generalized discrete logarithm problem in non-abelian groups	51
4.1	Algebra on the exponents	52
4.2	Two commuting operations	56
4.3	Diffie-Hellman problems based on the GDLP in non-abelian groups .	57
4.4	Diffie-Hellman key exchange based on the GDLP in non-abelian groups	58
4.5	ElGamal encryption scheme based on the GDLP in non-abelian groups	60
5	Cryptanalysis of the Tillich-Zémor hash function	63
5.1	Tillich-Zémor hashing scheme	63
5.1.1	Description of the Tillich-Zémor hashing scheme	64
5.1.2	Challenge parameters	64
5.1.3	Short relations	65
5.2	Experimental results	66
5.3	Finding short palindrome collisions	67
5.3.1	Collision preserving change of generators	68
5.3.2	Palindromic collisions	69
5.3.3	Maximal length chains in the Euclidean algorithm	74
5.4	Collisions for the challenge parameters	78

Bibliography	83
A Appendix on group $PSL(2, q)$ actions	91

Chapter 1

Introduction

In modern cryptography many cryptographic primitives are based on the intractability of the traditional discrete logarithm problem (DLP) in finite cyclic groups. However, a recent celebrated result of Peter Shor describes a quantum algorithm which solves DLP in polynomial time on a quantum computer [44]. Thus, the prospect of quantum computers becoming a practical reality in the future would render most of present-day public key cryptography totally obsolete. This motivates our research which explores the use of mathematical structures other than finite cyclic groups, in cryptology.

We investigate the hardness of the non-traditional discrete logarithm problem in certain finite non-abelian groups and possibilities to design secure cryptographic primitives in such mathematical structures. We develop algorithms which successfully attack the generalized DLP (GDLP) in the well known groups $SL(2, p)$ and $PSL(2, p)$, p a prime, with respect to special generators, when these groups are represented by 2×2 matrices, over a finite field of p elements. The two basic cases are when: i) the two generators are both of order p , ii) when one of two generators is of order p . The above two attacks constitute what we call basic p -attacks for the GDLP. The consequence is that such a representation of these groups together with the special

generators should not be used to design cryptographic primitives whose security relies on the intractability of the generalized DLP. We define the p -depth of a pair of generators and consider conditions on the p -depth under which a successful reduction to a basic p -attack can solve the GDLP.

Based on the assumption that the generalized discrete logarithm problem and conjugacy search problem are hard when the underlying groups, presentations and representations are chosen wisely, we generalize some existing cryptographic protocols to non-abelian groups. We generalize the computational Diffie-Hellman problem and the decision Diffie-Hellman problem to non-abelian groups and give a possible variant of the Diffie-Hellman key exchange protocol, and ElGamal encryption scheme based on the generalized discrete logarithm problem. We discuss the security of such protocols and cryptographic primitives built on the basis of the non-traditional generalized discrete logarithm and conjugacy problems.

We investigate the security of the Tillich-Zémor hash function, as a special instance of a cryptographic primitive defined over a finite, non-abelian group.

At CRYPTO '94, Tillich and Zémor proposed a cryptographic hash function [55] based on certain matrix groups. The Tillich-Zémor hash function has attracted considerable cryptanalytic interest, but stayed unbroken for more than fifteen years. The Tillich-Zémor hash function was considered to have good cryptographic properties. It allows parallelization in computation, has potentially efficient implementations, and stayed unbroken for a long time. In [30], the security of the Tillich-Zémor hash function was compared to SHA-1 and SHA-256 for certain parameters. Our breaking of the Tillich-Zémor hash function [12], was received as a considerable cryptographic breakthrough.

Based on extensive experimental results for small parameters, we restricted our search to collisions of palindromes of length $2n + 2$, where n represents the degree

of the input irreducible polynomial. We discovered that collisions are preserved if instead of the original Tillich-Zémor generators we use a different pair of symmetric generators obtained from the original generators by conjugation. We developed a series of results about the form of the hash value of the palindromes and connected our findings to a deep number theoretic result by Mesirov and Sweet about maximal length chains in the Euclidean algorithm over $\mathbb{F}_2[x]$, to produce our collisions. For the computation we used the standard algebra system Magma on a standard PC. We have constructed an algorithm for finding collisions for the Tillich-Zémor hash function for any chosen parameters. We have constructed actual collisions for the challenge parameters claimed to be secure in the scientific publications.

Our cryptanalytic attack shows that the Tillich-Zémor hash function should not be used in cryptographic applications where collision-resistance is essential.

Chapter 2

Preliminaries

2.1 Group theory

We assume familiarity with the standard notions of basic group theory and notation as one can find in [11, 13, 15, 42]. In particular, if G is a group we write $H \leq G$ to denote that H is a subgroup of G , and $H \trianglelefteq G$ that H is a normal subgroup of G . If $x \in G$, we denote by $|x|$ the *order* of x . For a set A , $|A|$ denotes the cardinality of A .

2.1.1 Group Actions

Let Ω be a set and G a group. A *right group action* is a map $\Omega \times G \rightarrow \Omega$, $(\alpha, g) \rightarrow \alpha^g$, which satisfies the following properties:

1. $\alpha^1 = \alpha$, for all $\alpha \in \Omega$, where $1 \in G$ is the identity of G .
2. $(\alpha^g)^h = \alpha^{gh}$, for all $g, h \in G$ and all $\alpha \in \Omega$.

Similarly, a *left group action* is defined to be a map $G \times \Omega \rightarrow \Omega$, $(g, \alpha) \rightarrow {}^g\alpha$, satisfying:

1. ${}^1\alpha = \alpha$, for all $\alpha \in \Omega$, where $1 \in G$ is the identity of G .

2. $g(h(\alpha)) = {}^{gh}\alpha$ for all $g, h \in G$ and all $\alpha \in \Omega$.

We will use right action notation to discuss basic properties of group actions. Clearly, analogous statements will hold for left actions. We denote a generic group action by $G|\Omega$ and say that “group G acts on Ω ”.

Let $G|\Omega$ be a group action and suppose that $A \subset \Omega$ and $H \subset G$. We write $A^H := \{a^h \mid a \in A \text{ and } h \in H\}$. Moreover, when $a \in \Omega$, $h \in G$, we write a^H for $\{a\}^H$, and A^h for $A^{\{h\}}$.

A group action $G|\Omega$ induces a relation \sim on Ω as follows: If $\alpha, \beta \in \Omega$, $\alpha \sim \beta$ if and only if there exists an element $g \in G$ such that $\beta = \alpha^g$. The relation \sim is an equivalence relation and decomposes Ω into equivalence classes, called the *orbits* of Ω under G . The orbit of a particular element $\alpha \in \Omega$ is easily seen to be the set α^G .

For a given group action $G|\Omega$ and $\alpha \in \Omega$, the set $G_\alpha = \{g \in G \mid \alpha^g = \alpha\}$ is called the *stabilizer* of α in G . It is easily seen that $G_\alpha \leq G$, for each $\alpha \in \Omega$. The *kernel* K of $G|\Omega$ is defined by $K = \{g \in G \mid \alpha^g = \alpha, \text{ for all } \alpha \in \Omega\}$. Thus, $K = \bigcap_{\alpha \in \Omega} G_\alpha$. We say $G|\Omega$ is *faithful* if and only if $|Ker(G|\Omega)| = 1$.

A group action $G|\Omega$ gives rise to a homomorphism $\pi : G \rightarrow \mathcal{S}_\Omega$ defined by:

$$\pi(g) = \begin{pmatrix} \cdots & \alpha & \cdots \\ \cdots & \alpha^g & \cdots \end{pmatrix}, \quad \alpha \in \Omega$$

where \mathcal{S}_Ω is the symmetric group on Ω . The kernel of π is the kernel of the action $G|\Omega$. In fact, the action and the homomorphism π are two different ways of viewing the same mathematical object. We say that the homomorphism π is a *permutation representation* of G . If π is faithful, we say that G is a *permutation group* on Ω .

For any group G , a particular group action of G on itself is centrally important. This is the action defined by $(x, y) \rightarrow x^y := y^{-1}xy$, and is well known as *conjugation*. When G acts on G by conjugation, the orbits are the *conjugacy* classes of G , and

a stabilizer G_x is the *centralizer* in G of x , that is, $G_x = \{y \in G \mid x^y = x\} = \{y \in G \mid xy = yx\}$.

For a group action $G|\Omega$ and element $g \in G$, the set $Fix(g) = \{\alpha \in \Omega \mid \alpha^g = \alpha\}$ is called the *Fix* of g in Ω . The function $\theta : G \rightarrow \mathbb{Z}$ defined by $\theta(g) = |Fix(g)|$ is called the *character* of the group action. A point $\alpha \in \Omega$ such that $\alpha^g = \alpha$, for all $g \in G$ is called a *fixed point* under G .

We say that two group actions $G|X$ and $Q|Y$ are *equivalent* if there exist bijections $\phi : G \rightarrow Q$ and $\lambda : X \rightarrow Y$, such that

- (i) ϕ is an isomorphism of G onto Q
- (ii) For all $g \in G$ and $x \in X$ we have that $\lambda(x^g) = \lambda(x)^{\phi(g)}$

We present a number of well known results without proof:

Lemma 2.1. *Suppose that $G|\Omega$ is a group action, $A \subset \Omega$, and that $x \in G$. Then $|A^x| = |A|$.*

Lemma 2.2. *Let $G|\Omega$ be a group action, and suppose that $\alpha \in \Omega$, and $x \in G$. Then $G_{\alpha^x} = (G_\alpha)^x$.*

Lemma 2.3. *Let $G|\Omega$ be a group action, and suppose that $x, y \in G$. Then (i) $Fix(x^y) = (Fix(x))^y$ and consequently (ii) $\theta(x^y) = \theta(x)$.*

Theorem 2.1. The orbit-stabilizer theorem. *If $G|\Omega$ is a group action and $\alpha \in \Omega$, then $|G| = |\alpha^G| |G_\alpha|$.*

Theorem 2.2. The Cauchy-Frobenius lemma. *(Also known as Burnside's lemma.) Let $G|\Omega$ be a group action. Then, the number of orbits of G on Ω is $\frac{1}{|G|} \sum_{g \in G} |Fix(g)|$.*

A group action $G|\Omega$ is said to be *transitive* if for any two elements $\alpha, \beta \in \Omega$, there exists an element $g \in G$ such that $\alpha^g = \beta$, i.e., there is a single orbit in Ω under the action of G , Ω itself. If the group action is not transitive it is called *intransitive*.

For a natural number k , a group action $G|\Omega$ is said to be *k-transitive* if for any two ordered k -tuples $(\alpha_1, \dots, \alpha_k)$, $(\beta_1, \dots, \beta_k)$ of distinct elements from Ω , there is an element $g \in G$ such that $(\alpha_1, \dots, \alpha_k) = (\beta_1, \dots, \beta_k)^g = (\beta_1^g, \dots, \beta_k^g)$.

$G|\Omega$ is *k-homogeneous* or *k*-transitive*, if G is transitive on the k -subsets of Ω .

Lemma 2.4. *A group action $G|\Omega$ is k -transitive if and only if the stabilizer G_α is $(k - 1)$ -transitive on $\Omega - \{\alpha\}$.*

Recall that when G acts on G by conjugation, the stabilizer G_g of element $g \in G$ is the centralizer of g in G , i.e. set of all elements $x \in G$ such that $gx = xg$. In this action, the orbit of $g \in G$ is the conjugacy class g^G . Based on the orbit-stabilizer Theorem 2.1, we have that $|g^G| = [G : C_G(g)]$. Interestingly, this action $G|G$ is in general not faithful as the kernel is clearly the center of G , $Z(G)$.

Let group G act on the set of its subgroups $\Omega = \{H \mid H \leq G\}$ by $H^g = g^{-1}Hg$, for $g \in G$. The orbits are called the *conjugacy classes of subgroups*. The orbit of subgroup H is the set $H^G = \{H^x \mid x \in G\}$. Two subgroups H and K are in the same conjugacy class if there exists a group element $g \in G$ such that $K = g^{-1}Hg$, we then say that H and K are *conjugate*. In this action, the stabilizer G_H is the subgroup $\{g \in G \mid Hg = gH\}$ of G , i.e., the *normalizer* of H in G , usually denoted $N_G(H)$. Thus, according to the orbit-stabilizer Theorem 2.1, the number of subgroups conjugate to subgroup H , is $[G : N_G(H)]$. The *centralizer* of subgroup H in G is the set $C_G(H) = \{g \in G \mid hg = gh \text{ for all } h \in H\} = \bigcap_{h \in H} C_G(h)$. Note that $C_G(H) \trianglelefteq N_G(H)$.

2.1.2 Basis theorem for transitive permutation representations

Let H be a subgroup of a group G . By a *right (left) transversal* of H in G we mean a complete set $T = \{x_i\}$ of distinct right (left) coset representatives of H . The following theorem characterizes the transitive actions of a group G , and discusses the corresponding character of such a group action.

Let G be a group, H a subgroup of G and let Ω be the collection of all distinct right cosets of H in G . Consider the group action $G|\Omega$ defined by $(Hx, g) \rightarrow Hxg$. For a discussion of induced representations, induced characters, and proofs of the various parts of the theorem that follows see [9], [13], [19] .

Theorem 2.3. (a) *The action $G|\Omega$ defined above is transitive, with kernel the core*

$$N \text{ of } H, \text{ that is, } N = \bigcap_{x \in G} H^x,$$

(b) *The character θ of the above action is the induced character $\theta = [1]_H \uparrow^G$ of the principal character of H to G ,*

(c) *If $g = |G|$, $h = |H|$, $x \in G$, $K_x = x^G$ is the conjugacy class of x in G , $g_x = |K_x|$ and $h_x = |K_x \cap H|$, Then*

$$\theta(x) = \frac{g \cdot h_x}{h \cdot g_x}$$

(d) *Any transitive action $G|X$ is equivalent to a group action $G|\Omega$ as above, where H can be chosen to be the stabilizer G_x for any given $x \in X$.*

(e) *Two transitive actions $G|X$ and $G|Y$ of a group G are equivalent if and only if, for $x \in X$, and $y \in Y$ the stabilizers G_x and G_y are conjugate in G .*

2.1.3 Special linear group and projective special linear group

Given a field \mathbb{F}_q , of q elements, and a fixed natural number n , the group of all $n \times n$ nonsingular matrices with respect to the operation of matrix multiplication is known as the *general linear group* of degree n over \mathbb{F}_q and is denoted by $GL(n, q)$. The order of the group is $|GL(n, q)| = \prod_{i=0}^{n-1} (q^n - q^i)$. The set of all matrices in $GL(n, q)$ of determinant 1 forms a subgroup of $GL(n, q)$, the *special linear group*, denoted by $SL(n, q)$. $SL(n, q)$ is the kernel of the homomorphism $\det : GL(n, q) \rightarrow \mathbb{F}_q^*$, and therefore $|SL(n, q)| = |GL(n, q)|/(q - 1)$. The center $Z(GL(n, q))$, of $GL(n, q)$, consists of all *scalar* matrices $\{\lambda I \mid \lambda \in \mathbb{F}_q^*\}$, thus the center of $SL(n, q)$ consists of all matrices $\{\lambda I \mid \lambda^n = 1\}$. The *projective special linear group* of degree n over \mathbb{F}_q , is the quotient group $PSL(n, q) = SL(n, q)/Z(SL(n, q))$. Here, we deal with the case $n = 2$, where q is odd, hence $|SL(2, q)| = (q^2 - 1)q$ and $|PSL(2, q)| = (q^2 - 1)q/2$.

For the group $PSL(n, q)$ it is also common to use the following notation: $PSL_n(q)$, $PSL_n(\mathbb{F}_q)$ or $PSL(n, \mathbb{F}_q)$, and similarly for the group $SL(n, q)$ to use the notation: $SL_n(q)$, $SL_n(\mathbb{F}_q)$ or $SL(n, \mathbb{F}_q)$.

In what follows we state without proof some of the properties of the groups $PSL(2, q)$ as they are the carrier groups in our study of the non-abelian discrete logarithm problem. The properties discussed below are well known and can be found in [6].

For $q \geq 4$ the groups $PSL(2, q)$ are simple. For q an odd prime power let $\mathbb{F} = GF(q)$ be Galois field of q elements and V the 2-dimensional vector space over \mathbb{F} . Two non-zero vectors $\mathbf{u}, \mathbf{v} \in V$ are defined to be *projectively equivalent* if $\mathbf{u} = s\mathbf{v}$ for some $s \in \mathbb{F}^*$. The $q + 1$ equivalence classes constitute the *projective line* \mathcal{L} , and $G = PSL(2, q)$ acts doubly transitively on \mathcal{L} by left matrix multiplication of the elements of V by the elements of G , modulo non-zero scalar multiples from \mathbb{F} , that is

$$((\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}), (\frac{x}{y})) \rightarrow ((\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}), (\frac{x}{y})) \rightarrow (\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}) (\frac{x}{y}) = (\frac{x'}{y'}) \rightarrow (\frac{x'}{y'})$$

As can be seen from the above line, a convenient way of viewing projectively the non-zero elements of V is to think of $(\frac{x}{y})$ as the quotient $\frac{x}{y} \in \mathbb{F} \cup \{\infty\}$, where we assign ∞ to quotients $\frac{x}{0}$ when $x \neq 0$, and the element $\frac{x}{y} \in \mathbb{F}$ otherwise.

It is also the case that there are exactly $q+1$ Sylow- p subgroups of G on which G acts doubly-transitively by conjugation. The latter representation of G is equivalent to the doubly transitive representation of G mentioned earlier. A Sylow- p subgroup Q of G is isomorphic to the additive group of \mathbb{F} , hence it is elementary abelian of order $q = p^m$. The normalizer in G of Q is of order $q(q-1)/2$ and is a split extension of Q by a cyclic group of order $(q-1)/2$.

When $q \equiv 3 \pmod{4}$, then in fact G acts as a 3-homogeneous group on the projective line \mathcal{L} . G is 2-transitive on \mathcal{L} but fails to be 3-homogeneous when $q \equiv 1 \pmod{4}$.

We gain further understanding of the structure of the groups $PSL(2, q)$ when we examine the following

Proposition 2.1. *For $q = p^m$, p an odd prime, consider the group $G = PSL(2, q)$ in its doubly transitive representation on the $q+1$ points of \mathcal{L} , and suppose that $x \in G$. Then x is semiregular on the points of \mathcal{L} not fixed by x , i.e. excluding the fixed points, all cycles of x have the same length. Moreover, exactly one of the following holds:*

- (a) x fixes 0 points, and belongs to a cyclic subgroup of order $(q+1)/2$, or
- (b) x fixes 1 point and is of order p , or
- (c) x fixes 2 points and belongs to a cyclic subgroup of order $(q-1)/2$.

We will need some additional well known facts about the groups $PSL(2, q)$, $q = p^m$, p an odd prime, which we state below without proof, as a proposition. In what follows ϕ stands for the Euler ϕ function.

Proposition 2.2. *Suppose that $G = PSL(2, q)$, $q = p^m$, p an odd prime. Then,*

- (a) *The Sylow- p subgroup of G is elementary abelian of order q ,*
- (b) *If $x \in G$ is of order d , then d divides $(q - 1)/2$, or $d = p$, or d divides $(q + 1)/2$,*
- (c) *There is a single conjugacy class of cyclic subgroups of order $(q - 1)/2$. Similarly, there is a single conjugacy class of cyclic subgroups of order $(q + 1)/2$.*
- (d) *If $x \in G$ is of order $d \neq 2$ dividing $(q \pm 1)/2$ then x belongs to one and only one cyclic subgroup of G of order $(q \pm 1)/2$.*
- (e) *If $d \neq 2$ divides $(q \pm 1)/2$ there are $\frac{\phi(d)}{2}$ conjugacy classes of element of order d in G . If $x \in G$ is an element of order d , then x is conjugate to x^{-1} .*
- (f) *If $x \in G$ is of order $d \mid (q \pm 1)/2$, $d \neq 2$, then the centralizer $C_G(x)$ is $\langle x \rangle$, while the normalizer $N_G(\langle x \rangle)$ is dihedral of order $q \pm 1$.*
- (g) *The centralizer of an element of order 2 has order $q - 1$ if $q \equiv 1 \pmod{4}$ and $q + 1$ if $q \equiv 3 \pmod{4}$.*

The following theorem collects in one place the information about the subgroup structure of $PSL(2, q)$ that can be found in various places in [6]. In what follows $q = p^m$, p an odd prime, $G = PSL(2, q)$, $M(q) = q(q^2 - 1)/2$, and d_{\mp} is any divisor of $(q \mp 1)/2$. For convenience G_d denotes a subgroup of order d .

Theorem 2.4. *Using the notation mentioned above, G has*

- (1) $q + 1$ conjugate elementary abelian groups of order q ,
- (2) $\frac{q(q\pm 1)}{2}$ conjugate cyclic groups of order $\frac{q\pm 1}{2}$,
- (3) $\frac{q(q\pm 1)}{2}$ conjugate cyclic groups $G_{d_{\mp}}$ of order d_{\mp} for every divisor d_{\mp} of $\frac{q\pm 1}{2}$,
- (4) $\frac{M(q)}{2 \cdot d_{\mp}}$ conjugate dihedral groups of order $2d_{\mp}$ for d_{\mp} odd,
- (5) Two systems each of $\frac{M(q)}{4 \cdot d_{\mp}}$ conjugate dihedrals $G_{2d_{\mp}}$ for d_{\mp} even, $d_{\mp} > 2$,
- (6) For $q \equiv \pm 3 \pmod{8}$, one set of $\frac{M(q)}{12}$ conjugate Klein 4-groups,
- (7) For $q \equiv \pm 1 \pmod{8}$, two sets each of $\frac{M(q)}{24}$ conjugate Klein 4-groups,
- (8) $\frac{(p^m-1)(p^m-p)\cdots(p^m-p^{t-1})}{(p^t-1)(p^t-p)\cdots(p^t-p^{t-1})}$ sets each of $\frac{p^{2m-1}}{(2,1)(p^k-1)}$ conjugate abelian groups of order p^t , where $(2,1)$ is read 2 or 1 according as $\frac{m}{k}$ is an even or odd integer. Here, k is a divisor of t depending on the particular G_{p^t} ,
- (9) Certain sets of $\frac{(p^{2m}-1)p^{m-t}}{(2,1)(p^k-1)}$ conjugate $G_{p^t d_-}$, where k and d_- depend on t ,
- (10) $(2,1)$ sets each of $\frac{M(q)}{(2,1)M(p^k)}$ conjugate $G_{M(p^k)} \cong PSL(2, p^k)$, where $k|m$,
- (11) Two systems each of $\frac{M(q)}{2M(p^k)}$ conjugate groups $G_{2M(p^k)} \cong PGL(2, p^k)$ when $\frac{m}{k}$ is an even integer,
- (12) For $q \equiv \pm 1 \pmod{8}$ two sets of $\frac{M(q)}{24}$ symmetric groups \mathcal{S}_4 ,
- (13) For $q \equiv \pm 1 \pmod{8}$ two sets of $\frac{M(q)}{24}$ alternating groups \mathbb{A}_4 ,
- (14) For $q \equiv \pm 3 \pmod{8}$, or $q = 2^m$, m even, $\frac{M(q)}{12}$ conjugate \mathbb{A}_4 's,
- (15) For $q \equiv \pm 1 \pmod{10}$, two sets of $\frac{M(q)}{60}$ conjugate alternating groups \mathbb{A}_5 .

An immediate Corollary of the information in Proposition 2.2 is the following

Corollary 2.1. *Let $G = PSL(2, p)$, p prime number. Suppose that $k \neq 2$ is a divisor of $(p \pm 1)/2$, then any conjugacy class of elements of order k has size $p(p \mp 1)$.*

We are now able to prove the following

Theorem 2.5. *Let $G = PSL(2, p)$, where p is a prime, and let H and K be any two cyclic subgroups of order $(p + 1)/2$ in G . Then, H and K are conjugate in G .*

Proof. Our goal is to show that the number of cyclic subgroups of order $(p + 1)/2$ in G is equal to the number of subgroups of G conjugate to H . As an immediate consequence we have that every two cyclic subgroups of order $(p + 1)/2$ in G are conjugate.

First, we count how many cyclic subgroups of order $(p + 1)/2$ there are in G . Let ν be the number of elements of order $(p + 1)/2$ in G , μ the number of elements of order $(p + 1)/2$ in any one conjugacy class of elements of order $(p + 1)/2$, and ρ the number of conjugacy classes in G of elements of order $(p + 1)/2$. Then, from Corollary 2.1 we have that

$$\nu = \mu \cdot \rho = p(p - 1) \cdot \frac{1}{2} \phi\left(\frac{p + 1}{2}\right).$$

But every $\phi\left(\frac{p + 1}{2}\right)$ elements of order $(p + 1)/2$ belong to the same cyclic subgroup of order $(p + 1)/2$ and determine the subgroup. Thus, there are in all

$$\nu / \phi\left(\frac{p + 1}{2}\right) = \frac{p(p - 1)}{2}$$

cyclic subgroups of order $(p + 1)/2$.

On the other hand, the class of subgroups conjugate to H in G has size

$$[G : N_G(H)] = \frac{p(p^2 - 1)/2}{(p + 1)} = \frac{p(p - 1)}{2}$$

Thus, there is a single conjugacy class of cyclic subgroups of order $(p + 1)/2$ in G . \square

A very similar argument establishes that any two cyclic subgroups of order $(p-1)/2$ are conjugate in $G = PSL(2, p)$. However, we give here an alternative proof of this fact.

Theorem 2.6. *Let $G = PSL(2, p)$, where $p > 3$ is an odd prime, and let A and B be elements of G , both of order $(p-1)/2$. Then, $\langle A \rangle$ and $\langle B \rangle$ are conjugate in G .*

Proof. Let X be the collection of all Sylow- p subgroups of G . Then $|X| = p+1$, and G acts doubly transitively on X by conjugation. Thus, if $K = \{(a, b) \in X \times X \mid a \neq b\}$, then $G|K$ is transitive, and K is a single orbit under this action. There are $(p+1)$ ways to choose a and p ways to choose b such that $a \neq b$ and $a, b \in X$. Therefore, the cardinality of the set K is $(p+1)p$. It follows that for any $x, y \in K$, $x \neq y$,

$$|G_{(x,y)}| = |G_{x,y}| = |G| / |K| = \frac{p(p^2-1)/2}{p(p+1)} = (p-1)/2.$$

Thus, the stabilizer of two points $G_{x,y}$ is of order $(p-1)/2$, and by Proposition 2.1 cyclic. The stabilizers of two points are of course conjugate as G is doubly transitive on X . □

Interestingly, we can compute the number of orbits of G acting by conjugation on pairs of cyclic subgroups in the slightly more general case where we are working with $G = PSL(2, q)$, q an odd prime power. More on these group actions can be found in the Appendix A.

2.2 Cryptography

The security of many present-day cryptosystems relies on the assumption of intractability of certain computational problems. Among these are the *discrete logarithm problem*, *integer factorization problem*, *Diffie-Hellman problem*, etc. We focus

our attention on the discrete logarithm problem.

2.2.1 Traditional discrete logarithm problem

Let G be a finite cyclic group generated by element α , and let β be an element of G . The traditional discrete logarithm problem (DLP) is to find a non-negative integer x such that $\alpha^x = \beta$.

When cryptographic primitives are built based on the discrete logarithm problem, it is required that the DLP be computationally intractable. Groups widely used in cryptographic applications in which the discrete logarithm problem is considered to be intractable are: the multiplicative group \mathbb{F}_q^* of the finite field \mathbb{F}_q , of order q , and a large cyclic subgroup of an appropriate elliptic curve \mathcal{E} over a finite field \mathbb{F}_q .

The intractability of the discrete logarithm problem depends on the group representation. For example, in the additive group \mathbb{Z}_n of integers modulo n , the discrete logarithm problem is easy to solve. Namely, for a given element β in \mathbb{Z}_n and generator α of \mathbb{Z}_n , it is easy to find the non-negative integer x such that $x\alpha = \beta$. Since $\gcd(n, \alpha) = 1$, the multiplicative inverse of α can be computed by means of the Extended Euclidean algorithm and hence the discrete logarithm revealed.

The difficulty of the traditional discrete logarithm problem is independent of the choice of the generator. Assume that α and γ are two generators of the cyclic group G of order n , and that there exist an algorithm which efficiently computes the discrete logarithm with respect to the generator α in G . Assume that $\alpha^x = \beta$ and that $\alpha^y = \gamma$. Then we can compute the discrete logarithm z of element β with respect to the generator γ . From $\alpha^x = \beta = \gamma^z = (\alpha^y)^z$ it follows $x = yz \pmod n$ and hence $z = xy^{-1} \pmod n$.

Algorithms which attack the discrete logarithm problem can be divided into three classes: (i) Algorithms which work in arbitrary groups: Exhaustive search, Baby-step

giant-step algorithm, Pollard's rho algorithm; (ii) Algorithms which work in arbitrary groups but are very efficient in groups whose order has only small prime factors: Pohlig-Hellman algorithm; (iii) Algorithms which work only in specific groups: Index-calculus algorithms.

2.2.2 Algorithms for computing discrete logarithms

To find the discrete logarithm of β with respect to base α , in a finite cyclic group of order n , generated by α , one could perform an exhaustive search and compute α^x for $x \in \{0, 1, \dots, n-1\}$ until an integer x is found such that $\alpha^x = \beta$. The expected running time for the exhaustive search is $O(n)$.

Baby-step giant-step algorithm

The *Baby-step giant-step* algorithm is a deterministic algorithm for solving the discrete logarithm problem in finite cyclic groups. The estimated running time of the algorithm is $O(\sqrt{n} \log \sqrt{n})$ group operations, where n is the order of the generator. Given generator α for the cyclic group G and an element $\beta \in G$, we want to find the least positive integer x such that $\alpha^x = \beta$. The Baby-step giant-step algorithm exploits the following property: for every non-negative integer $x \in \{0, 1, \dots, n-1\}$, there exist integers $i, j \in \{0, 1, \dots, m-1\}$ such that $x = im + j$, where $m = \lceil \sqrt{n} \rceil$, n being the order of the generator. Then, $\alpha^x = (\alpha^{mi})\alpha^j$ and consequently $\beta\alpha^{-j} = \alpha^{mi}$. To execute the algorithm we proceed as follows: first, the elements $\beta\alpha^{-j}$, $j = 0, 1, \dots, (m-1)$ are computed. For each j it is checked whether $\beta\alpha^{-j} = 1$. If this is the case, the solution of the discrete logarithm problem is j , otherwise $(j, \beta\alpha^{-j})$ is stored in a table, and we proceed with the next j . The table is then sorted according to the second component. Next, the elements α^{im} , $i = 0, \dots, (m-1)$ are computed. For each i in this range a binary search is conducted to determine whether α^{im} is in the table. If

the result is found in the table, say, $\alpha^{sm} = \beta\alpha^{-t}$, then, the discrete logarithm can be computed as $x = sm + t$, otherwise we continue with the next i . Clearly, the binary search will succeed for a unique $i \in \{0, 1, \dots, m-1\}$. It takes $\sqrt{n} \log \sqrt{n}$ steps to sort the list $\{\beta\alpha^{-j}\}$, and for each i it takes $\log \sqrt{n}$ steps to search for α^{im} in the above list. Therefore, there are $\sqrt{n} \log \sqrt{n} + \sqrt{n} \log \sqrt{n}$ steps in total which gives the worst case time complexity of the algorithm $O(\sqrt{n} \log \sqrt{n})$.

Pollard rho discrete logarithm algorithm

The *Pollard rho algorithm* is a randomized algorithm for computing discrete logarithms in finite cyclic groups. It requires negligible storage and has the same expected running time as the Baby-step giant-step algorithm. Since the Pollard rho algorithm does not require large storage it is considered more practical than the Baby-step giant-step algorithm.

Suppose that group G is cyclic of order n , generated by element α . For a given group element β , we want to compute the discrete logarithm $\log_{\alpha} \beta$.

The algorithm starts with partitioning the group G , based on some property that can be easily tested, into three sets: S_1, S_2, S_3 , all of approximately equal size, such that $1 \notin S_2$. Then, the sequence x_0, x_1, x_2, \dots is formed by iteratively applying function f :

$$x_{i+1} = f(x_i) = \begin{cases} \beta \cdot x_i, & \text{if } x_i \in S_1, \\ x_i^2, & \text{if } x_i \in S_2, \\ \alpha \cdot x_i, & \text{if } x_i \in S_3 \end{cases}$$

Sequences of numbers a_1, a_2, \dots and b_1, b_2, \dots such that $x_i = \alpha^{a_i} \beta^{b_i}$ are generated,

so that: $a_0 = 0, b_0 = 0$, and

$$a_{i+1} = \begin{cases} a_i, & \text{if } x_i \in S_1, \\ 2a_i, & \text{if } x_i \in S_2, \\ a_i + 1, & \text{if } x_i \in S_3 \end{cases}$$

$$b_{i+1} = \begin{cases} b_i + 1, & \text{if } x_i \in S_1, \\ 2b_i, & \text{if } x_i \in S_2, \\ b_i, & \text{if } x_i \in S_3 \end{cases}$$

We compute (x_i, a_i, b_i) and (x_{2i}, a_{2i}, b_{2i}) until for some $i \geq 1$, we reach $x_i = x_{2i}$. Then, $\alpha^{a_i} \beta^{b_i} = \alpha^{a_{2i}} \beta^{b_{2i}}$. It follows that $\alpha^{a_{2i}-a_i} = \beta^{b_i-b_{2i}}$. By taking logarithms with respect to base α of both sides of the equality, we obtain: $(b_i - b_{2i}) \log_\alpha \beta \equiv a_{2i} - a_i \pmod{n}$. If $\gcd(b_i - b_{2i}, n) = 1$, then $\log_\alpha \beta = (a_{2i} - a_i)(b_i - b_{2i})^{-1} \pmod{n}$.

Index calculus algorithm

The previously mentioned algorithms for computing discrete logarithms: *exhaustive search*, *Baby-step giant-step* and the *Pollard rho* algorithm, work in every cyclic group, regardless of its mode of representation. The *Index calculus* algorithm we are about to discuss, works only in cyclic groups exhibited in an appropriate representation mode. For example, the algorithm works in \mathbb{Z}_p^* , the multiplicative group of the integers modulo a prime p , and in $\mathbb{F}_{p^m}^*$, the multiplicative group of the finite field \mathbb{F}_{p^m} , where p is a prime. We assume that the group G is cyclic, generated by element α of order n , that the index calculus algorithm can be efficiently implemented in G and that an element $\beta \in G$ is given. The goal is to find an integer x such that $\beta = \alpha^x$, i.e., to find $x = \log_\alpha \beta$. In the **precomputation** phase of the algorithm, a factor base $\{p_1, \dots, p_r\} \subset G$ is selected, such that a significant number of elements from the

group G can be expressed as products of the elements of the factor base.

Then, relations of the form

$$\alpha^{s_i} = \prod_{j=1}^r p_j^{k_{ij}}$$

are generated for a collection of random integers $s \in \{0, \dots, n-1\}$. If for a particular s , α^s can not be expressed in terms of the factor base, then another random integer s is selected. The process stops when there are enough relations to solve for $\log_\alpha p_j$, $j \in \{1, \dots, r\}$, from the system of linear equations:

$$s_i = \sum_{j=1}^r k_{ij} \log_\alpha p_j \pmod n.$$

In the **computational** phase of the algorithm, a random integer $k \in \{0, \dots, n-1\}$ is repeatedly selected until $\beta\alpha^k$ is expressible as a product of elements of the factor base. For such a choice of k we then have

$$\beta\alpha^k = \prod_{i=1}^r p_i^{e_i}.$$

By taking logarithms of both sides of the previous equality, we obtain

$$\log_\alpha \beta + k = \sum_{i=1}^r e_i \log_\alpha p_i.$$

Then, the discrete logarithm x is:

$$x = \left(\sum_{i=1}^r e_i \log_\alpha p_i - k \right) \pmod n.$$

2.2.3 Diffie-Hellman problems and key exchange protocol

The Diffie-Hellman key exchange protocol enables two parties to establish a secret key over an insecure channel, without prior exchange of any secret information between them. The security of this cryptographic protocol relies on the assumption of hardness of the following problems: (i) the *discrete logarithm problem*, (ii) the *computational Diffie-Hellman problem* and (iii) the *decision Diffie-Hellman problem*. We have already discussed the discrete logarithm problem, we now present the other two problems: *computational Diffie-Hellman problem* and the *decision Diffie-Hellman problem*.

Computational Diffie-Hellman problem. Given a finite cyclic group generated by element α , and given α^x and α^y find α^{xy} .

Decision Diffie-Hellman problem. Given a finite cyclic group generated by element α of order n , and given α^x , α^y and α^z , determine whether $z \equiv xy \pmod{n}$.

If we can solve the discrete logarithm problem, then we can solve both the computational Diffie-Hellman problem and the decision Diffie-Hellman problem. If we can solve the computational Diffie-Hellman problem, then we can solve the decision Diffie-Hellman problem.

Assumptions that there do not exist polynomial time algorithms in the size of the order of the group which solve the computational Diffie-Hellman problem, and the decision Diffie-Hellman problem are referred to as the *computational Diffie-Hellman assumption* and the *decision Diffie-Hellman assumption*, respectively.

Diffie-Hellman key exchange protocol. This protocol enables two parties, Alice and Bob, to exchange a secret key without prior knowledge of each other. The protocol was originally invented by Whitfield Diffie and Martin Hellman and was published in [7]. The original implementation of the protocol used the multiplicative

group of integers modulo a prime p , however, we give here the description of the general Diffie-Hellman protocol where the underlying group is any finite cyclic group generated by element α .

Before starting the protocol, Alice and Bob agree on the finite cyclic group G and a generator α . These information items are publically known. To start the protocol, Alice selects a random positive integer a and sends α^a to Bob. Similarly, Bob selects a random positive integer b and sends α^b to Alice. Both, Alice and Bob, are able to compute the common secret key $\kappa = \alpha^{ab}$. Alice takes α^b she has received from Bob and computes $(\alpha^b)^a$. Bob takes α^a he has received from Alice and computes $(\alpha^a)^b$. Since, $(\alpha^a)^b = (\alpha^b)^a$, Alice and Bob hold the common secret key $\kappa = \alpha^{ab}$.

The original protocol is vulnerable to the *man-in-the-middle attack*. Namely, if a third party, named Oscar, intercepts the communication channel and obtain α^a , he may impersonate Bob by selecting a random positive integer c and by sending α^c to Alice. Alice, who thinks that she is communicating with Bob, establishes a secret key with Oscar, by computing $(\alpha^c)^a$. Since Oscar has obtained α^a from Alice, he computes $(\alpha^a)^c$. Oscar establishes a secret key with Alice and similarly, he establishes a secret key with Bob by impersonating the Alice. The original Diffie-Hellman protocol did not provide method or the authentication of the parties involved in the protocol. To prevent the man-in-the-middle attack, a method for **authentication** is needed such as **digital signatures**.

The secure protocol assumes that a method for authentication is used, that the discrete logarithm problem is hard in the underlying group, and that the decision Diffie-Hellman and the computational Diffie-Hellman problems are infeasible.

2.2.4 ElGamal cryptosystem

The ElGamal cryptosystem is a public key cryptosystem whose security is based on the computational infeasibility of the discrete logarithm problem and the computational Diffie-Hellman problem in the underlying group. Groups most commonly used for the ElGamal cryptosystem are: subgroups of the multiplicative group \mathbb{Z}_p^* of integers modulo a prime p , subgroups of the multiplicative group $\mathbb{F}_{p^n}^*$ of the finite field \mathbb{F}_{p^n} , where p is a prime, including the special case $p = 2$, the subgroup of points on an elliptic curve over a finite field, etc.

Suppose that Alice and Bob want to communicate via an insecure channel. We will assume that the cyclic group G of order n and generator α have been selected for this protocol by all entities and that it is commonly known how to multiply elements in the chosen group. It could also be the case that each entity selects the cyclic group G and the generator in the key generation phase. Prior to establishing communication, each party generates a (secret key, public key) pair by selecting a secret integer exponent $x \in \{0, \dots, n-1\}$ and computing α^x . The secret key, public key pair is $(x, (\alpha, \alpha^x))$.

Suppose that Bob wants to send a message to Alice. To encrypt the message, Bob obtains Alice's public key pair (α, α^a) , represents the message as a group element m , selects a random integer $b \in \{0, \dots, n-1\}$, computes $\omega = \alpha^b$, $\nu = m \cdot (\alpha^a)^b$ and sends (ω, ν) to Alice.

To decrypt the message and recover m from the cipher text she received from Bob, Alice computes ω^a and ω^{-a} and recovers the message as $m = \nu \cdot \omega^{-a} = m \cdot (\alpha^a)^b \cdot (\alpha^b)^{-a}$.

2.2.5 The Conjugacy problem

The *Conjugacy problem* is defined as follows: Given a group G and elements $x, y \in G$ decide whether there exists an element $g \in G$ such that $x^g = y$, i.e., $g^{-1}xg = y$.

The *Conjugacy search problem* is defined as follows: Given a group G and elements $x, y \in G$, which are known to be conjugate in G , find any element $g \in G$ such that $x^g = y$, i.e., $g^{-1}xg = y$.

The conjugacy search problem is considered to be computationally difficult in some groups, and as such, it has been used in cryptographic applications.

2.2.6 Cryptographic hash functions

Cryptographic hash functions play an important role in data integrity and message authentication. A hash function takes a bit string of arbitrary input length and outputs a short binary string of a particular fixed length, say 160 bits. The hash function should be easily computable. Denote by h the hash function. For message bit string x , the value $y = h(x)$ is called the hash value, message digest, digital fingerprint or hash. If the hash value y is kept in a secure place, and the message x is transmitted through an insecure channel, by using the hash value $h(x)$ it could be checked whether the original message x has been altered. The receiver of message x' computes $y' = h(x')$ and checks whether $y = y'$. If $y = y'$, the receiver accepts the message x' as the original one, i.e., $x = x'$. If the original message x has been altered, then with extremely high probability, $y \neq y'$.

To be considered secure, the hash function h has to satisfy the following three properties:

Preimage resistance. For a given hash value y it is computationally infeasible to find input x which hashes to y , i.e., such that $y = h(x)$.

Second Preimage resistance. For a given x , it is computationally infeasible to find x' such that $x \neq x'$ and that $h(x) = h(x')$.

Collision resistance. It is computationally infeasible to find any two different inputs x, x' which hash to the same output, i.e., such that $h(x) = h(x')$.

2.2.7 Generalized discrete logarithm problem in finite groups

The authors of [20] generalize the discrete logarithm problem from finite cyclic groups to arbitrary finite groups. We restate the definition.

Let G be a finite group generated by $\alpha_1, \dots, \alpha_t$, i.e., $G = \langle \alpha_1, \dots, \alpha_t \rangle$. Denote by $\alpha = (\alpha_1, \dots, \alpha_t)$, the ordered tuple of generators of the group G . As defined in [20], for a given $\beta \in G$, the generalized discrete logarithm problem (GDLP) of β with respect to α is to determine a positive integer k and a (kt) -tuple of non-negative integers $x = (x_{11}, \dots, x_{1t}, \dots, x_{k1}, \dots, x_{kt})$ such that

$$\beta = \prod_{i=1}^k (\alpha_1^{x_{i1}} \dots \alpha_t^{x_{it}}).$$

We can write this formally as $\beta = \alpha^x$. The (kt) -tuples $(x_{11}, \dots, x_{1t}, \dots, x_{k1}, \dots, x_{kt})$ are called the generalized discrete logarithms of β with the respect to $\alpha = (\alpha_1, \dots, \alpha_t)$.

Denote by

$$S_k = \left\{ \prod_{i=1}^k (\alpha_1^{x_{i1}} \dots \alpha_t^{x_{it}}) \mid x_{ij} \in \mathbb{Z}_{n_j} \right\}$$

where n_j denotes the order of element α_j . Then, the smallest positive integer k_0 such that for all $k \geq k_0$ $G \subseteq S_k$ is called the *depth* of group G with respect to $(\alpha_1, \dots, \alpha_t)$.

There could be more than one generalized discrete logarithm of β with respect to α . Actually, there will be infinitely many generalized discrete logarithms: if x

is a generalized discrete logarithm of β with respect to α and if $\alpha^{x'} = 1$, then, the catenations $x||x'$ and $x'||x$ are also generalized discrete logarithms of β with respect to α .

Chapter 3

Weak generalized discrete logarithms

In [20], the authors generalize the discrete logarithm from cyclic to any finite group. We assume that the generalized DLP is defined as in [20] and examine its tractability in the projective special linear group $PSL(2, p)$, where p is an odd prime. We show that in $PSL(2, p) = \langle \alpha, \beta \rangle$ the generalized DLP with respect to (α, β) is easy to solve for a specific group representation and specific choice of generators α and β .

As a consequence we have that such group representation of $PSL(2, p)$ together with particular generators should not be used in the design of cryptographic primitives whose security relies on the intractability of the GDLP.

3.1 Weak GDLP in $PSL(2, p)$ with respect to two specific generators

Consider the group $G = PSL(2, p)$ where p is an odd prime. Let α and β be any two non-commuting elements of order p in G , and let H and K be the subgroups of group G generated by α and β , respectively. In [20] the authors show that G is generated

by α and β and that $G = HKHK$. Thus, the depth of G with respect to generators α and β is two.

For the purpose of further analysis we assume that the group G is represented by matrices of $SL(2, p)$, up to a factor of $\pm I$, where I is the 2×2 identity matrix.

The matrices

$$A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

are both of order p , non-commuting and therefore generate G , i.e., $G = \langle A, B \rangle$. We show that the generalized discrete logarithm problem in G with respect to (A, B) can be solved efficiently.

Suppose that $M = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in G$, with $a, b, c, d \in \mathbb{F}_p$. Since the depth of G with respect to two generators of order p is two, M can be represented as $M = A^i B^j A^k B^\ell$ for some non-negative integers i, j, k, ℓ . Solving the generalized discrete logarithm problem means to find a tuple of non-negative integers (i, j, k, ℓ) such that $M = A^i B^j A^k B^\ell$.

Note that

$$A^i = \begin{pmatrix} 1 & i \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad B^j = \begin{pmatrix} 1 & 0 \\ j & 1 \end{pmatrix}.$$

Then,

$$A^i B^j A^k B^\ell = \begin{pmatrix} 1 & i \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ j & 1 \end{pmatrix} \begin{pmatrix} 1 & k \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ \ell & 1 \end{pmatrix}.$$

Hence,

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} 1 + ij + \ell((1 + ij)k + i) & (1 + ij)k + i \\ j + \ell(jk + 1) & jk + 1 \end{pmatrix}.$$

By equating corresponding entries of the matrices, we obtain the following system

of four equations with four unknowns i, j, k, ℓ in $\mathbb{F}_p = \mathbb{Z}_p$:

$$1 + ij + \ell((1 + ij)k + i) = a$$

$$(1 + ij)k + i = b$$

$$j + \ell(jk + 1) = c$$

$$jk + 1 = d$$

Indeed, the system of equations can be solved for i, j, k, ℓ using Gröbner basis computation. Let I be the ideal

$$I = \langle 1 + \ell k + ij + ijk\ell + i\ell - a,$$

$$k + ijk + i - b,$$

$$j + jk\ell + \ell - c,$$

$$jk + 1 - d \rangle.$$

A Gröbner basis GB for the ideal I is computed over the set of rational numbers:

$$GB = [\ell - jic + ja - c,$$

$$k + id - b,$$

$$jibc + ji - jab - a + bc + 1,$$

$$jid - jb + d - 1,$$

$$ad - bc - 1].$$

Therefore, solving the generalized discrete logarithm problem in the group $PSL(2, p)$ with respect to (A, B) is equivalent to solving the following system of equations in

$i, j, k, \ell \in \mathbb{Z}_p$.

$$\ell - jic + ja - c = 0$$

$$k + id - b = 0$$

$$jid - jb + d - 1 = 0$$

Generally, the system of equations has more than one solution. The following proposition provides a method for obtaining a solution when $M \in G$.

For the next proposition we continue to have $A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$, $B = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$ and $M = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$, elements of $PSL(2, p)$.

Proposition 3.1. *Let A, B and M be as above. Then, there exists a non-negative integer $n < p$ such that $nd - b \neq 0$ over \mathbb{Z}_p , and such that the 4-tuple (i, j, k, ℓ) with $i = n$, $j = (1 - d)(nd - b)^{-1}$, $k = b - nd$, $\ell = (1 - d)(nc - a)(nd - b)^{-1} + c$ provides a solution to $M = A^i B^j A^k B^\ell$.*

Proof. The proof consists of directly verifying that the given values for i, j, k, ℓ satisfy the above system of equations. The existence of n is ensured since $M \in PSL(2, p)$ and hence b and d can not simultaneously be equal to zero. \square

The example that follows illustrates the described method.

Example 3.1. *Consider the group $G = PSL(2, 7)$ represented by means of matrices of $SL(2, 7)$ modulo $\{\pm I\}$. Suppose $M, A, B \in G$ are as follows:*

$$M = \begin{pmatrix} 5 & 2 \\ 6 & 4 \end{pmatrix}, \quad A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}.$$

Computing the generalized discrete logarithm of matrix M with respect to the generators A and B corresponds to determining the tuple of non-negative integers (i, j, k, ℓ) such that $A^i B^j A^k B^\ell = M$.

The system we encountered earlier becomes:

$$1 + ij + \ell((1 + ij)k + i) = 5$$

$$(1 + ij)k + i = 2$$

$$j + \ell(jk + 1) = 6$$

$$jk + 1 = 4$$

Proposition 3.1 yields $(i, j, k, \ell) = (0, 5, 2, 2)$ for the choice $i = n = 0$. Simple matrix multiplication in \mathbb{Z}_7 shows that indeed $A^0 B^5 A^2 B^2 = M$. For a different choice of i , we could obtain different GDLP since the GDLP is not unique.

From the fact that the depth of $PSL(2, p)$ with respect to generating tuple (A, B) where $|A| = |B| = p$, is two, it follows that the depth of the same group with respect to generating tuple (B, A) is two as well. Therefore, every element $M \in PSL(2, p)$ can be written as $M = B^i A^j B^k A^l$. We show that this type of factorization of group elements can be obtained efficiently in this setup and therefore we obtain two different factorizations of the same group element:

$$A^{i_1} B^{j_1} A^{k_1} B^{\ell_1} = B^{i_2} A^{j_2} B^{k_2} A^{\ell_2}.$$

Assume that as before that $A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$, $B = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$ and $M = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$, elements of $PSL(2, p)$.

Non-negative integers i, j, k and ℓ such that $M = B^i A^j B^k A^l$ can be found efficiently by the method which is similar to one described for solving the GDLP with

respect to (A, B) . We give the outline of major steps.

$$B^i A^j B^k A^\ell = \begin{pmatrix} 1 & 0 \\ i & 1 \end{pmatrix} \begin{pmatrix} 1 & j \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ k & 1 \end{pmatrix} \begin{pmatrix} 1 & \ell \\ 0 & 1 \end{pmatrix}.$$

Hence,

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} 1 + jk & \ell(1 + jk) + j \\ i + k(ij + 1) & \ell(i + k(ij + 1)) + ij + 1 \end{pmatrix}.$$

By equating corresponding entries of the matrices, we obtain the following system of four equations with four unknowns i, j, k, ℓ in $\mathbb{F}_p = \mathbb{Z}_p$:

$$1 + jk = a$$

$$\ell(1 + jk) + j = b$$

$$i + k(ij + 1) = c$$

$$\ell(i + k(ij + 1)) + ij + 1 = d$$

Denote by I ideal generated by the previous equations:

$$I = \langle 1 + jk - a,$$

$$\ell(1 + jk) + j - b,$$

$$i + k(ij + 1) - c,$$

$$\ell(i + k(ij + 1)) + ij + 1 - d \rangle.$$

A Gröbner basis GB for the ideal I is computed over the set of rational numbers:

$$\begin{aligned} GB = [& \ell - jib + jd - b, \\ & k + ia - c, \\ & jia - jc + a - 1, \\ & jibc + ji - jcd + bc - d + 1, \\ & ad - bc - 1]. \end{aligned}$$

Therefore, solving the generalized discrete logarithm problem in the group $PSL(2, p)$ with respect to (B, A) is equivalent to solving the following system of equations in $i, j, k, \ell \in \mathbb{Z}_p$.

$$\ell - jib + jd - b = 0$$

$$k + ia - c = 0$$

$$jia - jc + a - 1 = 0$$

Therefore, we are able to state the following proposition. We continue to have $A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$, $B = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$ and $M = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$, elements of $PSL(2, p)$.

Proposition 3.2. *Let A, B and M be as above. Then, there exists a non-negative integer $n < p$ such that $na - c \neq 0$ over \mathbb{Z}_p , and such that the 4-tuple (i, j, k, ℓ) with $i = n$, $j = (1 - a)(na - c)^{-1}$, $k = c - na$, $\ell = (1 - a)(nb - d)(na - c)^{-1} + b$ provides a solution to $M = B^i A^j B^k A^\ell$.*

Proof. Again, the proof consists of directly verifying that the given values for i, j, k, ℓ satisfy the above system of equations. The existence of n is ensured since $M \in PSL(2, p)$ and hence a and c can not simultaneously be equal to zero. \square

3.2 Weak GDLP in $PSL(2, p)$ with respect to any two generators of order p

Suppose now that C and D are any two non-commuting elements of order p in $G = PSL(2, p)$, and that A and B are the matrices defined in the previous section. We have that $G = \langle C, D \rangle$, moreover, by exploiting the fact that G acts doubly transitively by conjugation on the $(p + 1)$ p -Sylow subgroups of G , we can efficiently solve the generalized discrete logarithm problem with respect to the generating tuple (C, D) . Thus, for any given $M \in G$ our goal is to determine non-negative integers i, j, k, ℓ such that:

$$M = C^i D^j C^k D^\ell.$$

Let Ω be the collection of all p -Sylow subgroups of G . Then $|\Omega| = p + 1$ and if $P \in \Omega$, then $|P| = p$. G has a doubly transitive representation on Ω by conjugation. Thus, the normalizer of $P \in \Omega$, $N_G(P)$, is of order $p(p - 1)/2$ and acts transitively on $\Omega \setminus \{P\}$.

Let $P, Q \in \Omega$ and let $g \in G$ such that $P^g = Q$, where $P^g = g^{-1}Pg$. There are in all $p(p - 1)/2$ elements $g \in G$ carrying P to Q by conjugation, for which holds $N_G(P)g = gN_G(Q)$. For any two pairs of p -Sylow subgroups, and hence for the particular pairs $(\langle A \rangle, \langle B \rangle)$ and $(\langle C \rangle, \langle D \rangle)$, there exists an element $g \in G$ such that

$$(\langle C \rangle, \langle D \rangle) = (\langle A \rangle^g, \langle B \rangle^g).$$

Then, C and D may be expressed as follows:

$$C = g^{-1}A^s g \quad D = g^{-1}B^t g$$

for some positive integers $s, t < p$.

To determine an element $g \in G$ such that $\langle A \rangle^g = \langle C \rangle$ and $\langle B \rangle^g = \langle D \rangle$, we proceed as follows. We determine an element $g_1 \in G$ such that $\langle A \rangle^{g_1} = \langle C \rangle$. Then $\langle B \rangle^{g_1} = \langle B_1 \rangle$. Now, $N_G(\langle C \rangle)$, acts transitively on $\Omega \setminus \{\langle C \rangle\}$. Therefore, there exists an element $g_2 \in N_G(\langle C \rangle)$, such that $\langle B_1 \rangle^{g_2} = \langle D \rangle$. Then, for $g = g_1 g_2$

$$\langle A \rangle^g = (\langle A \rangle^{g_1})^{g_2} = \langle C \rangle^{g_2} = \langle C \rangle, \quad \text{and}$$

$$\langle B \rangle^g = (\langle B \rangle^{g_1})^{g_2} = \langle B_1 \rangle^{g_2} = \langle D \rangle.$$

Note that the element g_2 can be chosen among the p elements of $\langle C \rangle$, i.e., from the centralizer of $\langle C \rangle$, as $\Omega \setminus \{\langle C \rangle\}$ is a single orbit of length p .

If we assume that the element $g \in G$ such that $gCg^{-1} \in \langle A \rangle$ and $gDg^{-1} \in \langle B \rangle$ has been found, then for some positive integers s and t , $A^s = gCg^{-1}$ and $B^t = gDg^{-1}$. On the other hand $A^s = \begin{pmatrix} 1 & s \\ 0 & 1 \end{pmatrix}$ and $B^t = \begin{pmatrix} 1 & 0 \\ t & 1 \end{pmatrix}$. Therefore, s is the $(1, 2)$ entry of the matrix gCg^{-1} and t is the $(2, 1)$ entry of the matrix gDg^{-1} .

Assume that we have computed element g . We may write:

$$\begin{aligned} M &= C^i D^j C^k D^\ell \\ &= (g^{-1} A^s g)^i (g^{-1} B^t g)^j (g^{-1} A^s g)^k (g^{-1} B^t g)^\ell \\ &= (g^{-1} A^{si} g) (g^{-1} B^{tj} g) (g^{-1} A^{sk} g) (g^{-1} B^{t\ell} g) \\ &= g^{-1} A^{si} B^{tj} A^{sk} B^{t\ell} g \end{aligned}$$

Let $x = si$, $y = tj$, $v = sk$ and $w = t\ell$. Then, $M = g^{-1} A^x B^y A^v B^w g$ and hence $gMg^{-1} = A^x B^y A^v B^w$. Let $M_1 = gMg^{-1}$. Obviously $M_1 \in G$ and $M_1 = A^x B^y A^v B^w$. Thus, we have transformed the generalized discrete logarithm problem of $PSL(2, p)$

with respect to C and D to the generalized discrete logarithm problem of $PSL(2, p)$ with respect to A and B which we were able to solve in the previous section. Therefore, since every nonzero element in \mathbb{Z}_p has an inverse, we are able to compute integers i, j, k and ℓ from $i = xs^{-1}, j = yt^{-1}, k = vs^{-1}, \ell = wt^{-1}$ where all operations are performed modulo p .

Note that it can not happen that s or t is equal to zero, due to the fact that $C = g^{-1}A^s g$ and $D = g^{-1}B^t g$. If, say, $s = 0$, then A^s is the identity matrix and therefore C would also be the identity matrix, which leads to the contradiction that C is matrix of order p . Similarly, $t \neq 0$.

The following example illustrates the algorithm we just described. Computations are performed using the Magma algebra system [3].

Example 3.2. Suppose that group $G = PSL(2, 7)$ is represented by matrices in $SL(2, 7)$ up to a factor of $\pm I$. Non-commuting matrices C, D of order $p = 7$ in G are given, as well as $M \in G$:

$$M = \begin{pmatrix} 3 & 5 \\ 2 & 6 \end{pmatrix} \quad C = \begin{pmatrix} 5 & 1 \\ 5 & 4 \end{pmatrix} \quad D = \begin{pmatrix} 2 & 5 \\ 4 & 0 \end{pmatrix}.$$

Our goal is to compute the generalized discrete logarithm of M with respect to (C, D) i.e., to find no-negative integers i, j, k, ℓ such that $M = C^i D^j C^k D^\ell$.

We use the matrices $A, B \in G$ which were defined in the previous section. First we find element $g_1 \in G$ such that $\langle A \rangle^{g_1} = \langle C \rangle$. Note that there are in all $p(p-1)/2 = 21$ elements $g_1 \in G$ such that $\langle A \rangle^{g_1} = \langle C \rangle$. These are the elements for which holds: $N_G(\langle A \rangle)g_1 = g_1 N_G(\langle C \rangle)$. One of them is $g_1 = \begin{pmatrix} 6 & 1 \\ 2 & 4 \end{pmatrix}$. Next, we compute

$B_1 = g_1^{-1} B g_1 = \begin{pmatrix} 2 & 6 \\ 1 & 0 \end{pmatrix}$. Element $g_2 = \begin{pmatrix} 2 & 5 \\ 5 & 6 \end{pmatrix}$ from $N_G(\langle C \rangle)$ is such that $\langle B_1 \rangle^{g_2} = \langle D \rangle$. Then, for $g = g_1 g_2 = \begin{pmatrix} 3 & 1 \\ 3 & 6 \end{pmatrix}$ the following holds: $\langle A \rangle^g = \langle C \rangle$ and

$\langle B \rangle^g = \langle D \rangle$. Integer s corresponds to the (1,2) entry in matrix $gCg^{-1} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$ while integer t corresponds to the (2,1) entry in the matrix $gDg^{-1} = \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix}$. Therefore, $s = 1$ and $t = 2$. So $s^{-1} = 1$ and $t^{-1} = 4$ in \mathbb{Z}_7 . Finally, $M_1 = gMg^{-1} = \begin{pmatrix} 3 & 3 \\ 1 & 6 \end{pmatrix}$.

We have transformed the given generalized discrete logarithm problem to the GDLP problem with respect to the canonical generators A and B for which

$$PSL(2, 7) = \langle A \rangle \langle B \rangle \langle A \rangle \langle B \rangle.$$

Namely, now we look for integers x, y, v, w such that $M_1 = A^x B^y A^v B^w$. By using Proposition 3.1 we obtain $(x, y, v, w) \in \{(0, 4, 3, 3), (1, 3, 4, 2), (2, 1, 5, 0), (3, 2, 6, 1), (5, 5, 1, 4), (6, 6, 2, 5)\}$ and the corresponding generalized discrete logarithms of M with respect to (C, D) are elements of the set $\{(0, 2, 3, 5), (1, 5, 4, 1), (2, 4, 5, 0), (3, 1, 6, 4), (5, 6, 1, 2), (6, 3, 2, 6)\}$.

An element $g \in G = PSL(2, p)$ such that $\langle C \rangle = \langle A \rangle^g$ and $\langle D \rangle = \langle B \rangle^g$ can also be computed by another method. We look for an element $g \in G$ which satisfies $C = g^{-1}A^s g$ and $D = g^{-1}B^t g$, for some non-negative integers $s, t < p$. Equivalently, we require that $g \in G$ satisfies the following equations: $gC = A^s g$ and $gD = B^t g$, for some non-negative integers $s, t < p$. Since $g = \begin{pmatrix} g_1 & g_2 \\ g_3 & g_4 \end{pmatrix} \in G$, we obtain a system of equations in g_1, \dots, g_4 and s and t from which an element g is determined.

3.3 Weak GDLP in $PSL(2, p)$ with respect to two generators one of which is of order p

Suppose now that $G = PSL(2, p) = \langle A, B \rangle$ where $|A| = p$ and we wish to write a given element $M \in G$ as a word in the generators A and B . Before we describe the method, we observe the following.

Proposition 3.3. *If $G = PSL(2, p) = \langle A, B \rangle$ where $|A| = p$, then $PSL(2, p) = \langle A, A^B \rangle$, where $A^B = B^{-1}AB$.*

Proof. Every two non-commuting elements of order p from $PSL(2, p)$ generate the whole group. So we prove that elements A and A^B are non-commuting of order p . Conjugate elements have the same order, so $|A^B| = |A| = p$. Now, suppose that elements A and A^B commute. Then, A^B is in the centralizer of element A , i.e., $A^B \in C_G(A) = \langle A \rangle$. So, $A^B = A^i$ for some $i \in \{0, \dots, p-1\}$. But then, B normalizes $\langle A \rangle$, i.e., $B \in N_G(\langle A \rangle)$, hence, $\langle A \rangle$ is a proper normal subgroup of $\langle A, B \rangle$. But $PSL(2, p)$ is simple, thus $\langle A, B \rangle$ can not be all of $PSL(2, p)$, a contradiction to the fact that A and B generate G . \square

We are now able to state the following proposition:

Proposition 3.4. *Suppose that $G = PSL(2, p) = \langle A, B \rangle$, where $|A| = p$, with no further assumptions on $|B| = m$. Then, the depth of G with respect to the generating tuple (A, B) is less than or equal to four.*

Proof. Let $C = A^B = B^{-1}AB$. By the Proposition 3.3 the group $PSL(2, p)$ is generated by elements A and C , both of order p . The GDLP can be solved efficiently in $PSL(2, p)$ represented by matrices, with respect to two generators of order p . By the method described earlier, the generalized discrete logarithm (i, j, k, ℓ) can be found

such that $M = A^i C^j A^k C^\ell$. To represent the element M in terms of the generators A and B we write the following sequence of equalities.

$$\begin{aligned}
M &= A^i C^j A^k C^\ell \\
&= A^i (B^{-1} A B)^j A^k (B^{-1} A B)^\ell \\
&= A^i B^{-1} A^j B A^k B^{-1} A^\ell B \\
&= A^i B^{m-1} A^j B A^k B^{m-1} A^\ell B
\end{aligned}$$

Therefore, the GDL of $M \in PSL(2, p)$ with respect to generating tuple (A, B) , where $|A| = p$ and $|B| = m$ is $(i, m-1, j, 1, k, m-1, \ell, 1)$. It follows that every element M from $PSL(2, p) = \langle A, B \rangle$, where $|A| = p$ and $|B| = m$ can be represented as $M = A^{x_1} B^{y_1} A^{x_2} B^{y_2} A^{x_3} B^{y_3} A^{x_4} B^{y_4}$ for some integers $x_1, x_2, x_3, x_4 \in \{0, \dots, p-1\}$ and $y_1, y_2, y_3, y_4 \in \{0, \dots, m-1\}$. The proposition follows. \square

The described method for writing an element M as a word in the generators A and B does not assure obtaining the shortest possible word that represent the given element as a word in the generators.

We illustrate the described procedure for solving the GDLP in $PSL(2, p)$ by means of the following example.

Example 3.3. Given $M = \begin{pmatrix} 9 & 1 \\ 1 & 10 \end{pmatrix} \in SL(2, 11) = \langle C, T \rangle$ where $C = \begin{pmatrix} 8 & 3 \\ 2 & 5 \end{pmatrix}$, $T = \begin{pmatrix} 2 & 9 \\ 0 & 6 \end{pmatrix}$. Write M as a word in C and T .

Note that $|C| = 11$ and $|T| = 5$, i.e., $|C| = p$ and $|T| = (p-1)/2$, where $p = 11$. Denote $D = C^T = T^{-1} C T = \begin{pmatrix} 5 & 9 \\ 8 & 8 \end{pmatrix}$, $|D| = 11$. Group $PSL(2, p)$ is generated by elements C and D , both of order p , i.e., $PSL(2, 11) = \langle C, D \rangle$. We have transfered GDLP from $PSL(2, p) = \langle C, T \rangle$ to $PSL(2, p) = \langle C, D \rangle$. Since C

and D are both of order p and generate whole group, we find non-negative integers (i, j, k, ℓ) such that $M = C^i D^j C^k D^\ell$. Denote by $A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$ and $B = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$. Element $g = \begin{pmatrix} 1 & 5 \\ 7 & 13 \end{pmatrix} \in SL(2, 11)$ is such that $(\langle A \rangle^g, \langle B \rangle^g) = (\langle C \rangle, \langle D \rangle)$. Then, $C = g^{-1} A^4 g$, $D = g^{-1} B^8 g$. $M_1 = g^{-1} M g = \begin{pmatrix} 4 & 3 \\ 5 & 4 \end{pmatrix} \in SL(2, 11)$. We have transfered GDLP from $PSL(2, p) = \langle C, D \rangle$ to $PSL(2, p) = \langle A, B \rangle$. Generalized discrete logarithms in $SL(2, 11) = \langle A, B \rangle$ with respect to (A, B) for a given M_1 are: $(0, 1, 3, 1)$, $(1, 8, 10, 2)$, $(2, 6, 6, 8)$, $(3, 7, 2, 5)$, $(4, 4, 9, 3)$, $(5, 5, 5, 0)$, $(6, 3, 1, 6)$, $(7, 10, 8, 7)$, $(8, 9, 4, 10)$, $(10, 2, 7, 9)$. Corresponding generalized discrete logarithms of M in $SL(2, 11) = \langle C, D \rangle$ with respect to (C, D) are: $(0, 7, 9, 7)$, $(3, 1, 8, 3)$, $(6, 9, 7, 1)$, $(9, 5, 6, 2)$, $(1, 6, 5, 10)$, $(4, 2, 4, 0)$, $(7, 10, 3, 9)$, $(10, 4, 2, 5)$, $(2, 8, 1, 4)$, $(8, 3, 10, 8)$. In particular,

$$M = \begin{pmatrix} 9 & 1 \\ 1 & 10 \end{pmatrix} = C^9 D^5 C^6 D^2.$$

We use the fact that $D = T^{-1} C T$ to obtain

$$M = C^9 (T^{-1} C T)^5 C^6 (T^{-1} C T)^2 = C^9 T^{-1} C^5 T C^6 T^{-1} C^2 T.$$

Written with non-negative exponents: $M = C^9 T^4 C^5 T C^6 T^4 C^2 T$. Thus, the GDL of M with respect to (C, T) is $(9, 4, 5, 1, 6, 4, 2, 1)$.

3.4 GDLP in $PSL(2, p)$ with respect to two generators none of which is of order p

We have seen that the GDLP for the groups $PSL(2, p)$, p prime number, in their matrix representation, with two generators, is easily solvable if at least one of the

generators is of order p . We examine the case when none of the generators is of order p . The problem is still quite open as the number of G -orbits by conjugation on generating pairs is very large. The hope is that one can eventually show that strong generating pairs do exist, or that the contrary is true, and there exist strategies for solving the GDLP for all cases of generating pairs. In what follows we examine some strategies which lead to a solution of the GDLP in a limited number of cases, with significant but rather small probabilities.

3.4.1 A strategy for attacking GDLP in $PSL(2, p)$

Unlike the case where at least one of the two generators of $G = PSL(2, p)$ is an element of order p , solving the GDLP is much more complex when neither of the two generators is of order p . The following proposition allows a cryptanalyst to reduce the GDL problem for $PSL(2, p)$ with any two generators to the earlier cases examined where at least one generator is of order p .

Proposition 3.5. *Suppose that $G = PSL(2, p) = \langle A, B \rangle$ where the orders of A and B are relatively prime to p , and suppose that $P = w_p(A, B)$, is a word in A and B which has order p as an element of G . Then, $G = \langle A, P \rangle$ or $G = \langle B, P \rangle$.*

Proof. Let $N = N_G(\langle P \rangle)$. Then, at least one of the elements A, B is not in N . Otherwise if A, B were both in N , then $\langle A, B \rangle$ would be a subgroup of N . Without loss of generality, suppose that $A \notin N$. Then $\langle A, P \rangle = G$, because the only proper subgroups of $PSL(2, p)$ containing $\langle P \rangle$ are subgroups of the normalizer of $\langle P \rangle$. Similarly, if $B \notin N$, it follows that $G = \langle B, P \rangle$. \square

Suppose that $G = \langle A, B \rangle$ with neither A, B of order p . Using Proposition 3.5, we may proceed to solve the GDLP for $M \in G = \langle A, B \rangle$, as follows. First construct an element $P \in G$ of order p as a word in the generators A and B , i.e., $P = w_p(A, B)$.

By Proposition 3.5, $G = \langle A, P \rangle$ or $G = \langle B, P \rangle$. Without loss of generality, assume that $G = \langle A, P \rangle$. Since $|P| = p$, we can solve the GDLP of M with respect to (A, P) . $M = w(A, P)$, i.e., M is written as a word in A and P . Then,

$$M = w(A, P) = w(A, w_p(A, B)) = w(A, B).$$

Note that the word $w(A, B)$ is not necessarily the shortest one that represents M in terms of the generators A and B . However, it does reveal the generalized discrete logarithm of M with respect to (A, B) .

3.4.2 The p -attack and its analysis

Successful cryptanalysis of the GDLP in $G = PSL(2, p) = \langle A, B \rangle$, when at least one of the two generators is of order p , rests on **reduction** to the basic case where the generators are the canonical elements $\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$, $\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$. We call a cryptanalytic attack of this type a **basic p -attack**.

Fundamentally, a basic p -attack is possible because i) G acts 2-transitively by conjugation on its p -Sylow subgroups, and ii) the depth with respect to the canonical generators is 2. It turns out that for an arbitrary pair (A, B) which generates G , a direct approach, such as the one used for the canonical generators of order p , will not be feasible because the number of orbits of G on $G \times G$ (by conjugation) is very large. Additionally, for each arbitrary generating pair (A, B) , the GDLP is of much higher complexity than for the canonical generators of order p . We show that even in the case where G acts by conjugation on $X \times X$, where X is the collection of all elements of order $(p-1)/2$ in G , the number of orbits is large, and reduction to a basic p -attack by means of Proposition 3.5 is generally not feasible. For arbitrary generators (A, B) , an attack based on the use of Proposition 3.5, will be called a **p -attack**.

Definition 3.1. Suppose that $G = PSL(2, p) = \langle A, B \rangle$. The p -depth of (A, B) , denoted by $pd(A, B)$, is the length of the shortest possible word in A and B which is of order p as an element of G .

For the rest of this section, let $d = (p-1)/2$ and $X = \{x \in G : |x| = d\}$. Since X is the union of $\phi(d)/2$ conjugacy classes of elements of order d , and for $x \in X$ the centralizer of x is $\langle x \rangle$, we have that

$$|X| = \frac{\phi(d)p(p+1)}{2}$$

Now, the action $G|X$ is intransitive and the number of orbits is $\phi(d)/2$. Consider now the induced action of G by conjugation on $X \times X$. That is, for $(x, y) \in X \times X$ and $g \in G$, $(x, y)^g = (x^g, y^g)$. Initially, we are interested in the number of orbits of $G|(X \times X)$.

Proposition 3.6. For G , X , and $G|(X \times X)$ as defined above, the number of orbits of the action $G|(X \times X)$ is

$$\eta = \frac{\phi(d)^2(p+3)}{2}$$

Proof. Let H be any particular fixed subgroup of G of order d . If $t = \phi(d)/2$ the elements of order d in H can be arranged in conjugate pairs as $\{(x_1, x_1^{-1}), \dots, \dots, (x_t, x_t^{-1}), \dots, (x_t, x_t^{-1})\}$ with x_i conjugate to x_j if and only if $i = j$. Let $Y = \{x_1, \dots, x_t\}$. Let Z be a collection of orbit representatives of H acting on X by conjugation. By elementary group theory, we can show that the pairs in $Y \times Z$ constitute a complete set of distinct representatives of the G -orbits on $X \times X$. Looking at the action $H|X$ by conjugation, we see that H commutes with the $\phi(d)$ elements of order d in H , thus these elements are fixed under conjugation by H . The remaining $|X| - \phi(d)$ elements of X fall into orbits of length d under H , because no element of

H fixes under conjugation (i.e. commutes with) any element of order d besides the elements of order d in H . This yields

$$\phi(d) + \frac{|X| - \phi(d)}{d}$$

H -orbits on X . Thus $\eta = |Y \times Z| = |Y| \cdot |Z|$ and we have

$$\begin{aligned} \eta &= \frac{\phi(d)}{2} \left(\phi(d) + \frac{|X| - \phi(d)}{d} \right) = \frac{\phi(d)}{2} \left(\phi(d) + \frac{\frac{\phi(d)p(p+1)}{2} - \phi(d)}{d} \right) = \frac{\phi(d)^2}{2} \left(1 + \frac{\frac{p(p+1)}{2} - 1}{d} \right) \\ &= \frac{\phi(d)^2}{4d} (2d + p(p+1) - 2) = \frac{\phi(d)^2}{4d} ((p-1) + p(p+1) - 2) = \frac{\phi(d)^2}{2(p-1)} (p^2 + 2p - 3) \\ &= \frac{\phi(d)^2}{2(p-1)} (p-1)(p+3) = \frac{\phi(d)^2(p+3)}{2} \quad \square \end{aligned}$$

It is clear that for $(x, y) \in X \times X$ and $g \in G$, the pairs (x, y) and $(x, y)^g = (x^g, y^g)$ have the same p -depth, that is, p -depth is an orbit invariant. For the cryptanalyst it would be highly desirable if the p -depth was bounded as a function of p , or if the distribution of p -depths over orbit representatives had a small mean.

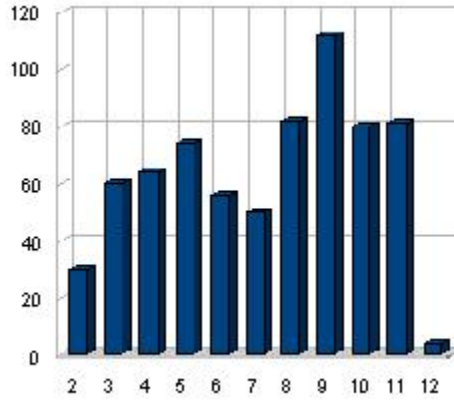


Figure 3.1: Distribution of p -depths for $p = 37$

p	η	max	mean		p	η	max	mean
		p -depth	p -depth				p -depth	p -depth
7	20	6	3.67		41	1408	15	6.80
11	112	8	4.61		43	3312	16	6.81
13	32	9	5.64		47	12100	16	6.92
17	160	12	5.92		53	4032	15	7.10
19	396	13	5.92		59	24304	18	7.22
23	1300	13	6.06		61	2048	14	7.43
29	576	13	6.37		67	14000	17	7.45
31	1088	14	6.57		71	21312	19	7.46
37	720	12	6.77		73	5472	17	7.62

Table 3.4.1

However, experimental evidence indicates this is not the case. On the Figure 3.1, the distribution of p -depths for $p = 37$ is presented. On the graph, the x -axis represents p -depth and the y -axis number of pair representatives in $PSL(2, 37)$. In the Table 3.4.1 we present experimental data for primes $p \leq 73$. The minimum p -depth is provably 2, and we investigate this case in a later section. The maximum p -depth generally increases with p , although it is not a strictly increasing function of p . The mean p -depth is indeed a slowly increasing function, for $7 \leq p \leq 73$. We further observe that the mean is always above $\log_2 p$, from which an interesting conclusion can be drawn. Indeed, for a given generating pair (A, B) , whether an oracle is available to provide $k = pd(A, B)$ or one finds $pd(A, B)$ by brute force, the time complexity for finding a shortest word $w_p(A, B)$ of order p is $O(2^k)$. Since $\text{mean } pd(A, B) > \log_2 p$, the worst case complexity of finding a $w_p(A, B)$ for an average (A, B) is at least $2^{\log_2 p} = p$. Since the system designer would choose large p , say of the order of 2^{100} , finding $w_p(A, B)$ by brute force would be infeasible, thus, a p -attack is generally infeasible if brute force method is used to construct $w_p(A, B)$.

3.4.3 Analysis of special cases

In cryptographic applications we can clearly assume that $p > 3$. Hence, one of the following conditions must hold for the prime p : (i) $p \equiv 1 \pmod{12}$, (ii) $p \equiv 5 \pmod{12}$, (iii) $p \equiv 7 \pmod{12}$, (iv) $p \equiv 11 \pmod{12}$.

As a consequence of Proposition 3.6, even if we restrict the generators A and B to have order $(p-1)/2$, the number of conjugacy classes on pairs of generators is too large to analyze. We presently restrict our attention to the case $p \equiv 1 \pmod{12}$ with $|A| = |B| = (p-1)/2$.

The following proposition holds for any odd **prime power** $q > 3$.

Proposition 3.7. *The probability that two elements generate $G = PSL(2, q)$ when selected randomly among all possible pairs of elements of order $(q-1)/2$ of G is $\frac{(q-1)(q-2)}{(q+1)q}$.*

Proof. G is doubly transitive on the $q+1$ points of the projective line $\mathcal{L} = \{1, \dots, q+1\}$, with stabilizer $G_{x,y}$ a cyclic subgroup of order $(q-1)/2$. Moreover, the ordered pairs (x, y) characterize the subgroups $G_{x,y}$. Let $G_{x,y} = \langle A \rangle$ and $G_{u,v} = \langle B \rangle$, then $\langle A, B \rangle = G$ if and only if $\{x, y\} \cap \{u, v\} = \emptyset$. Moreover, if $|\{x, y\} \cap \{u, v\}| = 1$ A and B lie in the stabilizer of the point in the intersection of $\{x, y\} \cap \{u, v\}$, thus $\langle A, B \rangle \neq G$. Further, if $\{x, y\} = \{u, v\}$ then $A \in \langle B \rangle$ and $\langle A, B \rangle = \langle B \rangle \neq G$. Since G is doubly transitive on the $q+1$ points, we can fix one pair of points, say $(1, 2)$ from $X = \mathcal{L} \times \mathcal{L}$ and count the number of pairs in X intersecting $\{1, 2\}$ in at least one point. These pairs are $Y = \{(1, 2)\} \cup \{(1, x) : x \notin \{1, 2\}\} \cup \{(x, 2) : x \notin \{1, 2\}\}$, and their symmetric flips. Thus, the total number of “short” pairs is $2(1 + 2(q-1)) = 4q - 2$, and the number of pairs corresponding to elements generating all of G is

$$q(q+1) - (4q-2) = q^2 - 3q + 2 = (q-1)(q-2)$$

Hence, the required probability is $\frac{(q-1)(q-2)}{q(q+1)}$ as claimed. \square

Remark 3.1. *As $q \rightarrow \infty$ the probability that two random elements of order $(q-1)/2$, generate G approaches 1.*

Recall now that in $PSL(2, q)$, q odd, there is a single conjugacy class of involutions and a single class of elements of order 3. Assume now that $q = p$ is a prime and $p \equiv 1 \pmod{12}$. Note that in this case 6 divides $(p-1)/2$. Suppose now that A and B are any two elements of order $(p-1)/2$, and let $z = A^{(p-1)/4}$ and $t = B^{(p-1)/6}$. Then, $|z| = 2$ and $|t| = 3$. We have already discussed that if p is large, with high probability A and B will generate G . The extremely interesting experimental fact is that if $G = \langle A, B \rangle$ then with probability at least 0.90 we will also have that $G = \langle z, t \rangle$. Although we have not established yet precisely this probability in terms of the parameters of the group, it can be easily shown that if $|zt| = p$ then we certainly have that $G = \langle z, t \rangle$. We have the following

Proposition 3.8. *Suppose that $p \equiv 1 \pmod{12}$, $G = PSL(2, p)$, A and B elements of order $(p-1)/2$ such that $FixA \cap FixB = \emptyset$. Let $z = A^{(p-1)/4}$ and $t = B^{(p-1)/6}$. Then, the order of zt is p with probability $\frac{2(p-1)}{p(p+1)}$, and in this case $G = \langle z, t \rangle$.*

Proof. Since there is a single conjugacy class of involutions, it suffices to count the pairs (z, t) where z is a particular fixed involution, t is an element of order 3, and zt is of order p . The centralizer $C_G(z)$ has order $(p-1)$ and acts by conjugation on the G -conjugacy class of $p(p+1)$ elements of order 3 in G . The class calculus shows that there are exactly two orbits of $C_G(z)$, each of length $|C_G(z)| = p-1$ of elements t such that zt has order p . Thus the required probability is $\frac{2(p-1)}{p(p+1)}$. \square

3.5 Relations in the context of cryptography

By solving the generalized discrete logarithm problem for a finite group with respect to a given set of generators we are factorizing group elements in terms of the generators. By equating two different factorizations of the same group element, we obtain a *relation*. This observation holds in any finite group.

Let G be a finite group generated by $\alpha_1, \dots, \alpha_t$, i.e., $G = \langle \alpha_1, \dots, \alpha_t \rangle$. Denote by $\alpha = (\alpha_1, \dots, \alpha_t)$ the ordered tuple of generators of the group G . For a given $\beta \in G$, assume that

$$\beta = \prod_{i=1}^k (\alpha_1^{x_{i1}} \dots \alpha_t^{x_{it}})$$

i.e., $\beta = \alpha^x$, where $x = (x_{11}, \dots, x_{1t}, \dots, x_{k1}, \dots, x_{kt})$.

Recall that $x = (x_{11}, \dots, x_{1t}, \dots, x_{k1}, \dots, x_{kt})$, the generalized discrete logarithm with respect to the generators $\alpha = (\alpha_1, \dots, \alpha_t)$, is not unique. In fact there will exist infinitely many distinct $y = (y_{11}, \dots, y_{1t}, \dots, y_{s1}, \dots, y_{st})$ such that $\beta = \alpha^y = \prod_{i=1}^s \alpha_1^{y_{i1}} \dots \alpha_t^{y_{it}}$. For any such y we have:

$$\prod_{i=1}^k \alpha_1^{x_{i1}} \dots \alpha_t^{x_{it}} = \prod_{i=1}^s \alpha_1^{y_{i1}} \dots \alpha_t^{y_{it}}.$$

In this way we obtain non-trivial relations among the generators. Further, by collecting different relations we may obtain a *presentation* of the group: $G = \langle X | R \rangle$, where X is the set of generators, and R a set of relations of the above type, sufficiently many to completely determine the group.

Relations of particular interest in cryptography are those which represent the identity element of the group, that is of the form $1_G = a$ *word in the generators*. Moreover, in a finite group G we can always convert a presentation of the form $G = \langle X | R \rangle$, into one of the form $G = \langle X | R' \rangle$, where R' is a set of relations of the

type: $\prod_{i=1}^k \alpha_1^{x_{i1}} \dots \alpha_t^{x_{it}} = 1_G$ with non-negative x_{ij} .

The *length* of word $w = \prod_{i=1}^k \alpha_1^{x_{i1}} \dots \alpha_t^{x_{it}}$ in the symbols $\alpha_1, \dots, \alpha_t$, where the x_{ij} are non-negative integers, is defined to be the integer $|w| = \sum_{i=1}^k \sum_{j=1}^t x_{ij}$. Moreover, if w_1 and w_2 are words in the symbols $\alpha_1, \dots, \alpha_t$ and $\rho : w_1 = w_2$ is a relation, the *length* of the relation is defined to be the integer $|\rho| := |w_1| + |w_2|$.

If G is a finite group generated by $\alpha_1, \dots, \alpha_t$, a relation ρ in the $\alpha_1, \dots, \alpha_t$ is said to be *short* if $|\rho| = O(\log(|G|))$, otherwise ρ is said to be *long*. Relations of importance to cryptographic hash functions designed in finite groups are those which are short.

We turn to our group of interest, $PSL(2, p)$, and examine the length of some relations there.

Let $G = PSL(2, p)$, and consider the elements $A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$, $B = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$ in G . The matrices A and B are both of order p , non-commuting and thus generate $PSL(2, p)$. As we have seen earlier, the depth of $PSL(2, p)$ with respect to the generating tuple (A, B) is two. Therefore, the identity matrix $I \in PSL(2, p)$ can be written as $I = A^i B^j A^k B^\ell$ for some non-negative integers i, j, k and ℓ . In the next proposition we establish that for any large prime p , any relation of the form $I = A^i B^j A^k B^\ell$ in $PSL(2, p)$ is long.

Proposition 3.9. *Let A, B and I be matrices in $PSL(2, p)$ as above. Then, a solution (i, j, k, ℓ) to the generalized discrete logarithm problem $I = A^i B^j A^k B^\ell$ is such that either $i + j + k + \ell \geq p$ or $i = j = k = \ell = 0$.*

Proof.

$$A^i B^j A^k B^\ell = \begin{pmatrix} 1 & i \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ j & 1 \end{pmatrix} \begin{pmatrix} 1 & k \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ \ell & 1 \end{pmatrix}.$$

Therefore,

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 + ij + \ell((1 + ij)k + i) & (1 + ij)k + i \\ j + \ell(jk + 1) & jk + 1 \end{pmatrix}.$$

Then, $jk + 1 = 1 \pmod{p}$ and hence $jk = 0 \pmod{p}$. By using $jk = 0 \pmod{p}$, we obtain

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 + ij + \ell k + \ell i & k + i \\ j + \ell & 1 \end{pmatrix}$$

So, $j + \ell = 0 \pmod{p}$ and $k + i = 0 \pmod{p}$, i.e., $j + \ell = s_1 p$, $s_1 \in \mathbb{Z}^0$ and $k + i = s_2 p$, $s_2 \in \mathbb{Z}^0$. If $s_1 \geq 1$, then $j + \ell \geq p$. Hence, $i + j + k + \ell \geq p$. If $s_1 = 0$, i.e., $j + \ell = 0$, then $j = \ell = 0$. Similarly, $s_2 \geq 1$ leads to $i + j + k + \ell \geq p$, and $s_2 = 0$ leads to $k = i = 0$. The length of the word $1_G = A^i B^j A^k B^\ell$, is $i + j + k + \ell \geq p$ or $i = j = k = \ell = 0$. Thus, $i + j + k + \ell \geq p > 3 \log p > \log(|PSL(2, p)|)$

□

Although relations of the form $I = A^i B^j A^k B^\ell$ in $PSL(2, p)$ are long for a large prime p , short relations of a different form do exist inside the group.

In the proposition that follows we prove the existence of short relations in any finite group G generated by two elements.

Proposition 3.10. *Let G be a finite group generated by two elements A and B . Then, there exist a relation $\rho : w_1 = w_2$ where w_1 and w_2 are two different words in A and B , such that $|w_1| + |w_2| \leq O(\log_2 |G|)$.*

Proof. We construct the blocks of all words of successive lengths in A and B . Let $B_0 = \{I\}$, where I is the identity of the group G . Let B_k be the collection of all words in A and B of length k . Then $|B_k| = 2^k$.

Let n be the positive integer such that $\sum_{k=0}^{n+1} |B_k| > |G|$ and such that $\sum_{k=0}^n |B_k| \leq |G|$. Since $\sum_{k=0}^n |B_k| = 2^{n+1} - 1$ we can write $2^{n+1} - 1 \leq |G|$, i.e., $2^{n+1} \leq |G| + 1$. By taking logarithms of both sides of the inequality, we obtain that $n + 1 \leq \log_2(|G| + 1)$.

By the pigeon-hole principle, two distinct words, say w_1 and w_2 belonging to $\{B_0 \cup B_1 \cup \dots \cup B_{n+1}\}$ must correspond to the same element of G . Then, $|w_1| + |w_2| \leq 2(n + 1) \leq 2\log_2(|G| + 1) = O(\log_2(|G|))$.

□

Of course the proof can be generalized to any finite group G generated by k generators. A direct consequence of Proposition 3.10 is that short relations in two generators do exist in $SL(2, q)$. In particular, for $G = SL_2(\mathbb{F}_{2^n})$, $|G| = 2^n(2^{2n} - 1)$, and there are short relations of length at most $6n$.

Chapter 4

Cryptographic primitives based on the generalized discrete logarithm problem in non-abelian groups

In this chapter we discuss possible cryptographic applications of the generalized discrete logarithm problem in finite non-abelian groups. Assuming the hardness of the GDLP in the underlying finite non-abelian group, we give a possible generalization of the Diffie-Hellman key exchange protocol and the ElGamal encryption scheme.

Earlier proposals have been made involving generalizations of the Diffie-Hellman key exchange protocol to finite non-abelian groups, see for example [49]. These schemes, however, have been cryptanalyzed in [46, 48]. Proposals for the construction of ElGamal-like schemes are given in [18, 24]. Ways of generalizing the DLP, the Diffie-Hellman key exchange protocol and the ElGamal cryptosystem to non-abelian groups are given in [25, 26, 27]. As related work we also mention the cryptosystem described in [37] which was later cryptanalyzed. Combinatorial group theory problems and, as a special case, the conjugacy search problem have been investigated in [45].

4.1 Algebra on the exponents

For the purpose of constructing cryptographic primitives and cryptosystems we present a way of defining exponentiation of a generating tuple of a finite group to a given integer.

If G is a multiplicative group, $\alpha \in G$ and x a non-negative integer, the meaning of α^x is well established. Presently, we wish to extend the notion of α^x to the case where α is an ordered set of elements $(\alpha_1, \dots, \alpha_t)$ of group G , where α_i has finite order n_i .

Definition 4.1. *Let x be a non-negative integer, and $\alpha = (\alpha_1, \dots, \alpha_t)$ an ordered set of elements of group G , where, for $i \in \{1, \dots, t\}$, n_i is the order of element α_i . Let $n = n_1 n_2 \dots n_t$. Denote by (x_1, x_2, \dots, x_k) the digits of x with respect to radix n , i.e.,*

$$x = x_1 n^{k-1} + x_2 n^{k-2} + \dots + x_{k-1} n + x_k .$$

For each x_i , let $(x_{i1}, x_{i2}, \dots, x_{it})$ be the digits of x_i with respect to the mixed radix (n_1, n_2, \dots, n_t) , thus, $0 \leq x_{ij} < n_j$. Then, by α^x we mean the group element

$$\alpha^x = \prod_{i=1}^k (\alpha_1^{x_{i1}} \dots \alpha_t^{x_{it}})$$

We also use the notation of exponentiation and product to write

$$\alpha^x = \prod_{i=1}^k [\alpha_1, \dots, \alpha_t]^{x_i},$$

where $[\alpha_1, \dots, \alpha_t]^{x_i} = \alpha_1^{x_{i1}} \dots \alpha_t^{x_{it}}$ and $(x_{i1}, x_{i2}, \dots, x_{it})$ are the digits of x_i with respect to the mixed radix (n_1, n_2, \dots, n_t) .

Example 4.1. Let α be a root of generating polynomial $f(x) = x^5 + x^2 + 1$ of the field \mathbb{F}_{2^5} . Consider special linear group $SL_2(\mathbb{F}_{2^5})$ generated by elements

$$A = \begin{pmatrix} \alpha & 1 \\ 1 & 0 \end{pmatrix} \quad B = \begin{pmatrix} \alpha & \alpha + 1 \\ 1 & 1 \end{pmatrix}$$

We compute $[A, B]^{1000}$. The order n_A of matrix A in $SL_2(\mathbb{F}_{2^5})$ is 31 and the order n_B of matrix B in the same group is 11. First, we write 1000 in base $n_A n_B = 341$: $1000 = 2 \times 341 + 318$. Therefore,

$$(A, B)^{1000} = [A, B]^2 [A, B]^{318} = A^0 B^2 A^{28} B^{10}$$

When computed, $(A, B)^{1000} = \begin{pmatrix} \alpha^{29} & \alpha^8 \\ \alpha^{17} & \alpha^{23} \end{pmatrix}$

Next, we define the operation of exponentiating an ordered t -tuple to a negative integer in a finite (non-abelian) group.

Definition 4.2. Suppose x is non-negative integer. If

$$(\alpha_1, \dots, \alpha_t)^x = \prod_{i=1}^k (\alpha_1^{x_{i1}} \dots \alpha_t^{x_{it}}) \quad (4.1.1)$$

is given as in definition 4.1, then

$$(\alpha_1, \dots, \alpha_t)^{-x} = \left(\prod_{i=1}^k \alpha_1^{x_{i1}} \dots \alpha_t^{x_{it}} \right)^{-1}$$

For applications it is useful to write $(\alpha_1, \dots, \alpha_t)^{-x}$ as a t -tuple of generators raised to a non-negative exponent. This can be done by applying the following facts. Every α_j , $j \in \{1, \dots, t\}$ generate a cyclic subgroup of G of order n_j . If $y \leq n_j$ and y is a non-negative integer, then, $\alpha_j^{-y} = \alpha_j^{n_j - y}$ and $n_j - y \geq 0$. Using the fact that

$(ab)^{-1} = b^{-1}a^{-1}$, we see that α^{-x} can be written in the form 4.1.1 as β^z where β is the reversal of tuple α and integer z is non-negative.

Definition 4.3. *Let G be a finite group generated by elements $\alpha_1, \dots, \alpha_t$. For $x, y \in \mathbb{Z}$*

$$(\alpha_1, \dots, \alpha_t)^{x \oplus y} \stackrel{\text{def}}{=} (\alpha_1, \dots, \alpha_t)^x (\alpha_1, \dots, \alpha_t)^y.$$

We define the relation \sim among the elements of the set of integers \mathbb{Z} as follows:

Definition 4.4. *Let G be the group generated by elements $\alpha_1, \dots, \alpha_t$. For $x, y \in \mathbb{Z}$ $x \sim y$ if and only if $(\alpha_1, \dots, \alpha_t)^x = (\alpha_1, \dots, \alpha_t)^y$ in G .*

Since \sim is an equivalence relation, it partitions the set \mathbb{Z} into equivalence classes. Let S be a complete set of representatives of equivalence classes of \sim . Then,

Theorem 4.1. *(S, \oplus) is a group isomorphic to the (G, \cdot) .*

As an immediate consequence of theorem 4.1, it follows that if the underlying group G is non-abelian, then the group (S, \oplus) is non-abelian.

The next example shows two elements in \mathbb{Z} which do not commute with respect to the operation \oplus when the underlying group is $SL_2(\mathbb{F}_{2^n})$.

Example 4.2. *Let A and B be as in example 4.1. Recall that $n = n_A n_B = 341$.*

$$353 = 1 \cdot 341^1 + 12 \cdot 341^0$$

$$695 = 2 \cdot 341^1 + 13 \cdot 341^0$$

$$\begin{aligned}
(A, B)^{353}(A, B)^{695} &= [A, B]^1[A, B]^{12}[A, B]^2[A, B]^{13} \\
&= A^0 B^1 A^1 B^1 A^0 B^2 A^1 B^2 \\
&= BAB^3 AB^2 \\
&= \begin{pmatrix} \alpha^4 & \alpha^8 \\ \alpha^{30} & \alpha^{18} \end{pmatrix}
\end{aligned}$$

On the other side,

$$\begin{aligned}
(A, B)^{695}(A, B)^{353} &= [A, B]^2[A, B]^{13}[A, B]^1[A, B]^{12} \\
&= A^0 B^2 A^1 B^2 A^0 B^1 A^1 B^1 \\
&= B^2 AB^3 AB \\
&= \begin{pmatrix} \alpha^{29} & \alpha^2 \\ \alpha^{20} & \alpha^9 \end{pmatrix}
\end{aligned}$$

So, $(A, B)^{353 \oplus 695} = (A, B)^{353}(A, B)^{695} \neq (A, B)^{695}(A, B)^{353} = (A, B)^{695 \oplus 353}$.

Remark 4.1. *There are some difficulties with our definition of α^x for $\alpha = (\alpha_1, \dots, \alpha_t)$ and $x \in \mathbb{Z}$. We enumerate some of these here:*

1. *The representation does not reduce to the usual definition for α^x when $t = 1$, i.e. when $\alpha = (\alpha_1)$. Suppose for example that $t = 1$, $n = n_1 = 10$ and $\alpha = (\alpha_1)$. Then for $x = 73 = 7 \cdot 10^1 + 3 \cdot 1$ our definition would yield $\alpha^{73} = \alpha_1^7 \cdot \alpha_1^3 = 1$ which is not compatible with the normal representation for which $\alpha^{73} = \alpha_1^3 \neq 1$.*
2. *A second incompatibility is that for integers x and y , $(\alpha^x)^y \neq \alpha^{(xy)}$. When $z = \alpha^x \in G$ is computed using our definition, then $(\alpha^x)^y$ is computed as z^y in the natural meaning of raising group element z to the integer power y .*
3. *It follows that $(\alpha^x)^y \neq (\alpha^y)^x$, for $\alpha = (\alpha_1, \dots, \alpha_t)$, $t > 1$ and $x, y \in \mathbb{Z}$. Thus a straight-forward application of a Diffie-Hellman protocol is not possible (using*

products of exponents in \mathbb{Z} .)

4. A consequence of 2. is that for $x, y, z \in \mathbb{Z}$,

$(x \oplus y)z \neq xz \oplus yz$. Thus the new operation \oplus is not compatible with ordinary multiplication in \mathbb{Z} , and certainly $(\mathbb{Z}, \oplus, \cdot)$ is not a ring.

4.2 Two commuting operations

The authors of [20] notice that the operation of conjugation by group elements commutes with exponentiation by integers as given in the definition of the generalized discrete logarithm. Consequently, the operation of conjugation commutes also with the exponentiation by an integer as defined in the previous section. Before we use this fact in building cryptographic primitives, we give a formal proof of the claim.

Theorem 4.2. *Let $G = \langle \alpha_1, \dots, \alpha_n \rangle$ be a finite non-abelian group. Let $(\alpha_1, \dots, \alpha_n)^x$ denote the operation of exponentiation by integer x and for $g \in G$ let $(\alpha_1, \dots, \alpha_n)^g$ denote the operation of conjugation:*

$$(\alpha_1, \dots, \alpha_n)^g = (\alpha_1^g, \dots, \alpha_n^g) = (g^{-1}\alpha_1g, \dots, g^{-1}\alpha_ng).$$

Then,

$$((\alpha_1, \dots, \alpha_n)^x)^g = ((\alpha_1, \dots, \alpha_n)^g)^x.$$

Proof.

$$\begin{aligned}
((\alpha_1, \dots, \alpha_n)^g)^x &= (\alpha_1^g, \dots, \alpha_n^g)^x \\
&= \prod_{i=1}^k ((\alpha_1^g)^{x_{i1}} \dots (\alpha_n^g)^{x_{in}}) \\
&= \prod_{i=1}^k ((g^{-1}\alpha_1 g)^{x_{i1}} \dots (g^{-1}\alpha_n g)^{x_{in}}) \\
&= \prod_{i=1}^k (g^{-1}\alpha_1^{x_{i1}} \dots \alpha_n^{x_{in}} g) \\
&= g^{-1} \left(\prod_{i=1}^k (\alpha_1^{x_{i1}} \dots \alpha_n^{x_{in}}) \right) g \\
&= \left(\prod_{i=1}^k (\alpha_1^{x_{i1}} \dots \alpha_n^{x_{in}}) \right)^g \\
&= ((\alpha_1, \dots, \alpha_n)^x)^g
\end{aligned}$$

□

4.3 Diffie-Hellman problems based on the GDLP in non-abelian groups

A possible direct generalization of the computational and decision Diffie-Hellman problems to finite non-abelian groups is as follows.

Computational Diffie-Hellman Problem

Given a finite non-abelian group $G = \langle \alpha_1, \dots, \alpha_n \rangle$ and $(\alpha_1, \dots, \alpha_n)^x$ and $(\alpha_1, \dots, \alpha_n)^y$ where x, y are integers, find $((\alpha_1, \dots, \alpha_n)^x)^y$.

Decision Diffie-Hellman Problem

Given a finite non-abelian group $G = \langle \alpha_1, \dots, \alpha_n \rangle$ and $(\alpha_1, \dots, \alpha_n)^x, (\alpha_1, \dots, \alpha_n)^y$ and $(\alpha_1, \dots, \alpha_n)^z$ where x, y, z are integers, determine whether $((\alpha_1, \dots, \alpha_n)^x)^y =$

$$(\alpha_1, \dots, \alpha_n)^z.$$

As we have mentioned in Remark 4.1, in finite non-abelian groups, the equality $((\alpha_1, \dots, \alpha_n)^x)^y = ((\alpha_1, \dots, \alpha_n)^y)^x$ does not generally hold (actually, almost never holds). However, in the traditional Diffie-Hellman key exchange protocol, integer exponents commute. This commutativity property of exponents is the fundamental reason why two parties can obtain a common secret key in the key exchange protocol. In order to have commutativity of the exponents, we generalize the computational and decision Diffie-Hellman problems as follows.

Computational Diffie-Hellman Problem

Given a finite non-abelian group $G = \langle \alpha_1, \dots, \alpha_n \rangle$, $(\alpha_1, \dots, \alpha_n)^x$ and $(\alpha_1, \dots, \alpha_n)^g$ where x is an integer and $g \in G$, find $((\alpha_1, \dots, \alpha_n)^g)^x$.

Decision Diffie-Hellman Problem

Given a finite non-abelian group $G = \langle \alpha_1, \dots, \alpha_n \rangle$ and $(\alpha_1, \dots, \alpha_n)^x$, $(\alpha_1, \dots, \alpha_n)^g$ and $(\alpha_1, \dots, \alpha_n)^z$ where x, z are integers and $g \in G$, determine whether $((\alpha_1, \dots, \alpha_n)^g)^x = (\alpha_1, \dots, \alpha_n)^z$.

Based on the Theorem 4.2, $((\alpha_1, \dots, \alpha_n)^x)^g = ((\alpha_1, \dots, \alpha_n)^g)^x$, $x \in \mathbb{Z}$ and $g \in G$.

4.4 Diffie-Hellman key exchange based on the GDLP in non-abelian groups

Two parties, Alice and Bob, want to establish a secret key over an insecure channel. Alice and Bob have not exchanged any secret information in the past, and possibly have no prior knowledge of each other. They agree on the group $G = \langle \alpha_1, \dots, \alpha_n \rangle$ they will use. Therefore, the group and the generators are publicly known. To begin the protocol, Alice selects a random positive **integer** x , computes $g_a = (\alpha_1, \dots, \alpha_n)^x$

and sends it to the Bob. Bob selects a random **group element** $g \in G$, computes $g_b = (\alpha_1, \dots, \alpha_n)^g$ and sends it to Alice. Both parties are now able to compute the common secret key. Alice receives g_b from Bob, she takes her secret integer x and computes:

$$g_b^x = ((\alpha_1, \dots, \alpha_n)^g)^x = k_A.$$

Bob receives g_a from Alice and computes

$$g_a^g = ((\alpha_1, \dots, \alpha_n)^x)^g = k_B.$$

Based on the Theorem 4.2, $k_A = k_B$. The common secret key is $k = k_A = k_B$.

The security of the Diffie-Hellman key exchange protocol just described depends on the assumption of intractability of the generalized discrete logarithm problem and also on the assumption of intractability of the conjugacy problem in the underlying group.

If a third party, named Oscar, is listening to the communication through the channel, he is able to obtain g_a and g_b . Recall that $G = \langle \alpha_1, \dots, \alpha_n \rangle$ is publicly known. If Oscar is able to solve the GDLP of g_a in G with respect to $(\alpha_1, \dots, \alpha_n)$, he obtains k -tuples (x_{i1}, \dots, x_{in}) and positive integer k . Since Oscar knows $g_b = (\alpha_1^g, \dots, \alpha_n^g)$, he is able to compute $\prod_{i=1}^k ((\alpha_1^g)^{x_{i1}} \dots (\alpha_n^g)^{x_{in}})$.

$$\begin{aligned} \prod_{i=1}^k ((\alpha_1^g)^{x_{i1}} \dots (\alpha_n^g)^{x_{in}}) &= \prod_{i=1}^k (g^{-1} \alpha_1^{x_{i1}} \dots \alpha_n^{x_{in}} g) \\ &= g^{-1} \left(\prod_{i=1}^k (\alpha_1^{x_{i1}} \dots \alpha_n^{x_{in}}) \right) g \\ &= \left(\prod_{i=1}^k (\alpha_1^{x_{i1}} \dots \alpha_n^{x_{in}}) \right)^g \\ &= ((\alpha_1, \dots, \alpha_n)^x)^g \end{aligned}$$

Note that there are more than one solutions to the generalized discrete logarithm problem. Oscar can use any of these solutions to produce the common key.

If Oscar is able to solve the simultaneous conjugacy search problems, he may use the value g_b that he strips from the insecure channel to find $g^* \in G$ such that $g_b = (\alpha_1, \dots, \alpha_n)^{g^*}$. Note that $g^* \in G$ must satisfy $(g^*)^{-1}\alpha_i g^* = g^{-1}\alpha_i g$ for all $i \in \{1, \dots, n\}$. There could be more than one element in the group G that satisfy this property. Oscar does not have to know the exact secret g that Bob used to construct g_b , to be able to discover the common secret key of Bob and Alice. Once he knows g^* , he computes the secret key $g_a^{g^*}$. Next, we show that $g_a^{g^*} = g_a^g$:

$$\begin{aligned}
g_a^{g^*} = ((\alpha_1, \dots, \alpha_n)^x)^{g^*} &= ((\alpha_1, \dots, \alpha_n)^{g^*})^x \\
&= (\alpha_1^{g^*}, \dots, \alpha_n^{g^*})^x \\
&= (\alpha_1^g, \dots, \alpha_n^g)^x \\
&= ((\alpha_1, \dots, \alpha_n)^g)^x \\
&= ((\alpha_1, \dots, \alpha_n)^x)^g \\
&= g_a^g
\end{aligned}$$

4.5 ElGamal encryption scheme based on the GDLP in non-abelian groups

By using the fact that the operation of exponentiating a generating tuple by an integer in finite non-abelian groups commutes with conjugation, the ElGamal encryption scheme can be generalized to non-abelian groups.

The security of the aforementioned generalized ElGamal encryption scheme depends on the assumption that the GDLP and conjugacy search problems are intractable in the underlying group.

Key generation

Each entity ε creates a private key and the corresponding public key pair as follows. Entity ε selects non-abelian group $G = \langle \alpha_1, \dots, \alpha_n \rangle$, a randomly selects positive integer x_ε , computes $g_\varepsilon = (\alpha_1, \dots, \alpha_n)^{x_\varepsilon}$ and publishes g_ε and $(\alpha_1, \dots, \alpha_n)$, but keeps x_ε secret. In particular Alice' secret key is x_a , and public key $(g_a, (\alpha_1, \dots, \alpha_n))$, where $g_a = (\alpha_1, \dots, \alpha_n)^{x_a}$.

Encryption

To send a message to Alice, Bob obtains Alice's public key pair $(g_a, (\alpha_1, \dots, \alpha_n))$, and writes the message m as an element of the group $G = \langle \alpha_1, \dots, \alpha_n \rangle$. Then, he selects his secret key, a random element $g \in G$, computes $(\beta_1, \dots, \beta_n) = (\alpha_1, \dots, \alpha_n)^g$, and sends $((\beta_1, \dots, \beta_n), mg_a^g)$ to Alice.

Decryption

To decrypt the message from Bob, Alice uses her secret key x_a to compute $((\beta_1, \dots, \beta_n)^{x_a})^{-1}$ and multiplies on the right the mg_a^g she received from Bob.

$$\begin{aligned} mg_a^g((\beta_1, \dots, \beta_n)^{x_a})^{-1} &= \\ mg_a^g(((\alpha_1, \dots, \alpha_n)^g)^{x_a})^{-1} &= \\ m((\alpha_1, \dots, \alpha_n)^{x_a})^g(((\alpha_1, \dots, \alpha_n)^{x_a})^g)^{-1} &= m. \end{aligned}$$

If Oscar, who listens to the communications channel, could solve the simultaneous conjugacy search problem, he could find an element $g^* \in G$ such that $\alpha_i^{g^*} = \beta_i$, for all $i \in \{1, \dots, n\}$, and he would be able to compute $(g_a^{g^*})^{-1}$ which would enable him

to decrypt the message.

If Oscar is able to solve GDLP for g_a with respect to $(\alpha_1, \dots, \alpha_n)$, he can find x such that $g_a = (\alpha_1, \dots, \alpha_n)^x$ and can compute $((\alpha_1, \dots, \alpha_n)^g)^x)^{-1}$ and therefore decrypt the message.

Therefore, potential implementation of this generalization of the ElGamal encryption scheme would require hard generalized discrete logarithm problem and hard conjugacy search problem in the underlying group.

Chapter 5

Cryptanalysis of the Tillich-Zémor hash function

5.1 Tillich-Zémor hashing scheme

At CRYPTO'94, Jean-Pierre Tillich and Gilles Zémor, [55] proposed a new family of cryptographic hash functions. Their work attracted significant cryptanalytic interest [10, 4, 1, 50, 40], but, for carefully chosen parameters, the hashing scheme remained unbroken for more than fifteen years.

In this dissertation we show that the Tillich-Zémor hash function is not collision resistant by showing how to construct collisions efficiently for any choice of the input parameters. More specifically, we construct collisions of palindromic bit strings of length $2n + 2$, where n is the degree of the irreducible polynomial used to define the field \mathbb{F}_{2^n} . For each irreducible polynomial of degree n , we construct two palindromic collisions from which we deduce two more collisions of non-palindromic bit strings.

5.1.1 Description of the Tillich-Zémor hashing scheme

Denote by V collection of all bitstrings of arbitrary finite length, i. e., $V := \{0, 1\}^*$. If $v = b_1 \dots b_m \in V$ is a bitstring, then $v^r := b_m \dots b_1$ denotes the **reversal** of bitstring v . A **palindrome** is a bitstring $v \in V$ satisfying $v = v^r$. We denote by $|v|$ the *length* of $v \in V$.

Assume that the finite field \mathbb{F}_{2^n} is represented as $\mathbb{F}_{2^n} := \mathbb{F}_2[x]/(q(x))$ where $q(x)$ is a given irreducible polynomial of degree n . Let α be a root of $q(x)$ and denote by G the group $SL_2(\mathbb{F}_{2^n})$ of 2×2 matrices of determinant 1 over \mathbb{F}_{2^n} . Define matrices s_0 and s_1 of group G by:

$$s_0 := \begin{pmatrix} \alpha & 1 \\ 1 & 0 \end{pmatrix}, \quad s_1 := \begin{pmatrix} \alpha & \alpha + 1 \\ 1 & 1 \end{pmatrix} \in G.$$

According to the Tillich-Zémor proposal in [55], a bitstring $v = b_1 \dots b_m \in V$ is hashed by applying the function $\check{h}: V \longrightarrow G$:

$$\check{h}(b_1 \dots b_m) := s_{b_1} \cdots s_{b_m} \in G$$

Remark 5.1. *At ICECS '08 [41] and CT-RSA '09 [40] vectorial and projective variants of the Tillich-Zémor hash function were proposed and in [30] these ideas were combined. By construction, any collision for the original Tillich-Zémor proposal also yields a collision for these more recent proposals. Hence, throughout, we restrict our attention to constructing collisions for the original proposal from CRYPTO '94.*

5.1.2 Challenge parameters

The Tillich and Zémor hash function uses an irreducible polynomial of degree n as an input parameter. Once the irreducible polynomial is fixed, the function is completely determined. Originally, Tillich and Zémor suggested values for $n \in \{130, \dots, 170\}$.

After cryptanalytic attack [50], which is effective for composite n , the suggested input parameters were reduced to n being a prime number from the set $\{131, \dots, 167\}$. A prime number n close to 1024 was suggested in [39].

The most recent proposal for input parameters is given in [41], where the following choices of the irreducible polynomial $q(x)$ are proposed to define the underlying $\mathbb{F}_{2^n} = \mathbb{F}_2[x]/(q(x))$:

$$\begin{aligned} & x^{127} + x + 1 \\ & x^{251} + x^7 + x^4 + x^2 + 1 \\ & x^{509} + x^8 + x^7 + x^3 + 1 \\ & x^{1021} + x^5 + x^2 + x + 1 \\ & x^{2039} + x^{10} + x^9 + x^8 + x^7 + x^5 + x^4 + x^2 + 1 \end{aligned}$$

In [30] de Meulenaer et al. suggest to use polynomial $x^{127} + x + 1$ in order to provide collision resistance comparable to SHA-1, and to use polynomial $x^{251} + x^7 + x^4 + x^2 + 1$ to provide collision resistance comparable to SHA-256. According to [30] all of the above five polynomials are safe with respect to the attacks in [10, 1, 50].

To demonstrate the applicability of our cryptanalytic attack, we construct collisions for all of the aforementioned polynomials.

5.1.3 Short relations

We note that the problem is specific to the representation of \mathbb{F} as well as to the generators. The respective orders k, ℓ of s_0 and s_1 could be very large, for example any divisors of $2^n + 1$ or $2^n - 1$, and can be efficiently calculated. If k or ℓ is small, then the system can be effectively attacked because one can write a short relation, such as $s_0^k = I$, or $s_1^\ell = I$, where I is the identity of G . Thus a successful attack must assume nothing about the orders k and ℓ . We have proved in Proposition 3.10 the existence of short relations in a finite group generated by two elements and concluded

the existence of short relations in two generators in $SL(\mathbb{F}_{2^n})$.

5.2 Experimental results

Our early experiments were restricted to cases in which the defining irreducible polynomial $q(x)$ is of degree n , small enough to enable us to perform a brute force attack in searching for collisions. Data analysis of experimental results showed that for every n we obtain collisions of words of length $2n+2$. We observed that among those collisions there exist colliding palindromes. We performed all computation on a standard PC, using the computer algebra system Magma [3].

Example 5.1. When irreducible polynomial $q(x) = x^5 + x^4 + x^3 + x + 1$ is used to define the field $\mathbb{F}_{2^5} = \mathbb{F}_2[x]/(q(x))$ and with Tillich-Zémor generators s_0, s_1 , the following collisions of palindromes of length $2 \cdot 5 + 2 = 12$ occur:

$$\begin{aligned} \check{h}(0 \overbrace{00110}^{v_1} \overbrace{01100}^{v_1^r} 0) &= \check{h}(1 \overbrace{00110}^{v_1} \overbrace{01100}^{v_1^r} 1) \\ \check{h}(0 \overbrace{11101}^{v_2} \overbrace{10111}^{v_2^r} 0) &= \check{h}(1 \overbrace{11101}^{v_2} \overbrace{10111}^{v_2^r} 1) \end{aligned}$$

We observe that the bitstring v_2 can be obtained by reversing bitstring v_1 followed by inverting the first and the last bit.

For n small enough to be able to list all irreducible polynomials of degree n , we checked for the existence of the collisions of palindromes of length $2n + 2$ for each irreducible polynomial of the degree n . The next example shows the list of all irreducible polynomials over \mathbb{F}_2 of degree 5 and corresponding collisions of palindromes of length 12.

Example 5.2. For each irreducible polynomial $q(x)$ of degree 5 used to define the field $\mathbb{F}_{2^5} = \mathbb{F}_2[x]/(q(x))$ and with Tillich-Zémor generators s_0, s_1 , there are exactly two collisions of palindromes of length 12.

Irreducible polynomial $q(x)$	Collisions of palindromes of length $2n + 2$
$x^5 + x^2 + 1$	$\check{h}(001110011100) = \check{h}(101110011101)$ $\check{h}(011111111110) = \check{h}(111111111111)$
$x^5 + x^4 + x^2 + x + 1$	$\check{h}(000000000000) = \check{h}(100000000001)$ $\check{h}(010001100010) = \check{h}(110001100011)$
$x^5 + x^3 + x^2 + x + 1$	$\check{h}(000010010000) = \check{h}(100010010001)$ $\check{h}(011001100110) = \check{h}(111001100111)$
$x^5 + x^3 + 1$	$\check{h}(000111111000) = \check{h}(100111111001)$ $\check{h}(001101101100) = \check{h}(101101101101)$
$x^5 + x^4 + x^3 + x + 1$	$\check{h}(011101101110) = \check{h}(111101101111)$ $\check{h}(000110011000) = \check{h}(100110011001)$
$x^5 + x^4 + x^3 + x^2 + 1$	$\check{h}(011000000110) = \check{h}(111000000111)$ $\check{h}(010010010010) = \check{h}(110010010011)$

Experimental results showed that for each tested choice of $\mathbb{F}_{2^n} = \mathbb{F}_2[x]/(q(x))$ there are exactly two bitstrings $v_1, v_2 \in \{0, 1\}^n$, $|v_1| = n$, $|v_2| = n$, such that

$$h(0v_i v_i^r 0) = h(1v_i v_i^r 1) \quad (i = 1, 2).$$

5.3 Finding short palindrome collisions

Based on our experimental results we restrict our search for collisions to a search for palindromic collisions of bitstrings of length $2n + 2$. We proceed in three steps. First,

we change the original generators, while preserving collisions, secondly, we develop results characteristic for palindromic collisions but work inside the group $SL_2(\mathbb{F}_2[x])$ of unimodular matrices over the polynomial ring $\mathbb{F}_2[x]$ rather than over a field \mathbb{F}_{2^n} . Thirdly, we establish a connection between the Tillich-Zémor proposal and maximal length chains in the Euclidean algorithm for polynomials over the field of two elements and connect the underlying problem to a crucial result of Mesirov and Sweet [29].

5.3.1 Collision preserving change of generators

Recall that the original Tillich-Zémor generators are elements of $SL_2(\mathbb{F}_{2^n})$:

$$s_0 := \begin{pmatrix} \alpha & 1 \\ 1 & 0 \end{pmatrix}, \quad s_1 := \begin{pmatrix} \alpha & \alpha + 1 \\ 1 & 1 \end{pmatrix},$$

where α is a root of the irreducible polynomial used to define the field \mathbb{F}_{2^n} .

We define new generators c_0 and c_1 and show that the search for collisions for the Tillich-Zémor hashing scheme can be translated to a search for collisions when the new generators are used for hashing.

Set $c_0 = s_0$ and obtain c_1 by conjugating s_1 by s_0 :

$$c_1 := s_0^{-1} s_1 s_0.$$

After performing the computations, the new generators are:

$$c_0 := \begin{pmatrix} \alpha & 1 \\ 1 & 0 \end{pmatrix}, \quad c_1 = \begin{pmatrix} \alpha + 1 & 1 \\ 1 & 0 \end{pmatrix}.$$

The new generators differ in exactly one entry and both matrices are symmetric. We now define a new hash function: $h: V \longrightarrow G$ by:

$$h(b_1 b_2 \dots b_m) := c_{b_1} \cdots c_{b_m} \in G.$$

The next proposition shows that the collisions in the original Tillich-Zémor generators occur for exactly the same bitstrings as collisions in the generators c_0 and c_1 .

Proposition 5.1. *Let $v_1, v_2 \in V$. Then, $\check{h}(v_1) = \check{h}(v_2)$ if and only if $h(v_1) = h(v_2)$.*

Proof. Let $v_1 = b_1 b_2 \dots b_n$ and $v_2 = b'_1 b'_2 \dots b'_m$ be two bitstrings from V . Then $\check{h}(v_1) = \check{h}(v_2)$, written in terms of bitstrings $\check{h}(b_1 b_2 \dots b_n) = \check{h}(b'_1 b'_2 \dots b'_m)$, is equivalent to $s_{b_1} \dots s_{b_n} = s_{b'_1} \dots s_{b'_m}$. Conjugate both sides of the equality by s_0 to obtain:

$$s_0^{-1}(s_{b_1} \dots s_{b_n})s_0 = s_0^{-1}(s_{b'_1} \dots s_{b'_m})s_0.$$

The last equality is equivalent to

$$\prod_{i=1}^n (s_0^{-1} s_{b_i} s_0) = \prod_{i=1}^m (s_0^{-1} s_{b'_i} s_0).$$

Since, $s_0^{-1} s_0 s_0 = s_0 = c_0$ and $s_0^{-1} s_1 s_0 = c_1$, the last equality is equivalent to $\prod_{i=1}^n c_{b_i} = \prod_{i=1}^m c_{b'_i}$, or equivalently $h(v_1) = h(v_2)$. \square

5.3.2 Palindromic collisions

We define matrices $C_0, C_1 \in SL_2(\mathbb{F}_2[x])$, with polynomial entries, as follows:

$$C_0 := \begin{pmatrix} x & 1 \\ 1 & 0 \end{pmatrix}, \quad C_1 := \begin{pmatrix} x+1 & 1 \\ 1 & 0 \end{pmatrix} \in SL_2(\mathbb{F}_2[x]),$$

and define $H: V \longrightarrow SL_2(\mathbb{F}_2[x])$ by:

$$H(b_1 b_2 \dots b_m) := C_{b_1} \dots C_{b_m} \in SL_2(\mathbb{F}_2[x])$$

Notice that the function H is defined in an analogous way to function h except that when H is applied to a bitstring, it gives a product of matrices in $SL_2(\mathbb{F}_2[x])$ and the result in $SL_2(\mathbb{F}_2[x])$ rather than in the field \mathbb{F}_{2^n} .

When applied to palindromes, function H has some interesting properties which we list in the next lemma.

Lemma 5.1. *Let $v \in V$ be a palindrome, and write $H(v) = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$. Then, $b = c$, i. e., $H(v)$ is symmetric. Moreover, a has degree $\deg(a) = |v|$, and we have $\max(\deg(b), \deg(d)) \leq |v|$.*

Proof. We prove the lemma by induction on the length $|v|$ of v .

If $|v| \leq 1$, then v is either empty word or the one bit word 1 or 0. Then $H(v)$ is either the identity matrix, or else C_0 or C_1 , all of which satisfy the claimed properties.

Assume that palindromes of length $n - 2$ satisfy the claimed properties and let ω be a palindrome of length n . Then, $H(\omega) = C_0 H(v) C_0$ or $H(\omega) = C_1 H(v) C_1$, where v is a palindrome of length $n - 2$ and where $H(v) = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$. In the first case,

$$C_0 H(v) C_0 = \begin{pmatrix} ax^2 + (b + c)x + d & ax + c \\ ax + b & a \end{pmatrix}.$$

In the second case,

$$C_1 H(v) C_1 = \begin{pmatrix} ax^2 + (b + c)x + a + b + c + d & a(x + 1) + c \\ a(x + 1) + b & a \end{pmatrix}.$$

By the induction hypothesis, $b = c$, and the first part of the statement follows directly.

The statement about the degrees can be verified directly.

□

The following proposition states that if $v \in V$ is palindrome of even length, then the diagonal entries of $H(v)$ are squares.

Proposition 5.2. *If $v \in V$ is a palindrome of even length, then $H(v) = \begin{pmatrix} a^2 & b \\ b & d^2 \end{pmatrix}$ for some $a, b, d \in \mathbb{F}_2[x]$.*

Proof. Suppose that $v \in V$ is a palindrome of even length. Then, we can write $v = ww^r$ for some $w \in V$. The proof is by induction on $|w|$.

If $|w| = 0$, $H(v)$ is identity matrix and the statement follows.

Suppose that w is extended by a single bit β , $\beta \in \{0, 1\}$. Then the form of the palindrome is $\beta v \beta = (\beta w)(w^r \beta)$. By the induction hypothesis,

$$H(v) = H(ww^r) = \begin{pmatrix} a^2 & b \\ b & d^2 \end{pmatrix},$$

and hence

$$H(\beta v \beta) = C_\beta \begin{pmatrix} a^2 & b \\ b & d^2 \end{pmatrix} C_\beta = \begin{pmatrix} (x + \beta)^2 a^2 + d^2 & (x + \beta)a^2 + b \\ (x + \beta)a^2 + b & a^2 \end{pmatrix}$$

□

Our goal is to construct collisions in $SL_2(\mathbb{F}_2[x]/(q(x)))$ of the form $h(0v0) = h(1v1)$, where v is a palindrome of length $2n$. Now, $h(0v0) = h(1v1)$ if and only if $h(0v0) + h(1v1)$ is the zero matrix in $SL_2(\mathbb{F}_2[x]/(q(x)))$. This motivates us to define the function $\rho: V \longrightarrow \mathbb{F}_2[x]^{2 \times 2}$ by

$$\rho(v) := H(0v0) + H(1v1).$$

$h(0v0) = h(1v1)$ is a collision in $SL_2(\mathbb{F}_2[x]/(q(x)))$ if and only if $\rho(v)$ is equal to zero matrix modulo irreducible polynomial $q(x)$, i.e., $\rho(v) \equiv \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \pmod{q(x)}$.

The next proposition concerns the form of the matrix entries in $\rho(v)$, when v is a palindrome.

Proposition 5.3. *Let $v \in V$ be a palindrome of length $|v|$. Then, $\rho(v) = \begin{pmatrix} a & a \\ a & 0 \end{pmatrix}$ where $a \in \mathbb{F}_2[x]$ has degree $|v|$ and a is the upper left entry of $H(v)$.*

Proof. If $v \in V$ is a palindrome, then by the Lemma 5.1, $H(v)$ is matrix with entries (1,2) and (2,1) equal. We can write: $H(v) := \begin{pmatrix} a & b \\ b & d \end{pmatrix}$. With $C_0 = \begin{pmatrix} x & 1 \\ 1 & 0 \end{pmatrix}$ and $C_1 = \begin{pmatrix} x+1 & 1 \\ 1 & 0 \end{pmatrix}$, direct computation yields:

$$\rho(v) = H(0v0) + H(1v1) = C_0 H(v) C_0 + C_1 H(v) C_1 = \begin{pmatrix} a & a \\ a & 0 \end{pmatrix}$$

From Lemma 5.1, the degree of upper left entry of $H(v)$, for $v \in V$ being a palindrome, is equal to $|v|$. Therefore, $\deg(a) = |v|$. \square

A direct consequence of Propositions 5.3 and 5.2 is the following corollary:

Corollary 5.1. *Let $v \in V$ be a palindrome of even length. Then $\rho(v) = \begin{pmatrix} a^2 & a^2 \\ a^2 & 0 \end{pmatrix}$ for some $a \in \mathbb{F}_2[x]$ with $\deg(a) = |v|/2$ and where a^2 is the upper left entry of $H(v)$.*

Example 5.3. *With C_0 and C_1 as above and with palindrome $v = 0011001100$, we have $\rho(v) = H(000110011000) + H(100110011001)$. When computed:*

$$\rho(v) = \begin{pmatrix} x^{10} + x^8 + x^6 + x^2 + 1 & x^{10} + x^8 + x^6 + x^2 + 1 \\ x^{10} + x^8 + x^6 + x^2 + 1 & 0 \end{pmatrix}$$

We notice that the nonzero polynomial entries in matrix $\rho(v)$ are squares of the polynomial $a(x) = x^5 + x^4 + x^3 + x + 1$. Moreover, $\deg(a) = 5$, i.e., exactly half of the length of the given palindrome and a^2 is the upper left entry of $H(v)$:

$$H(v) = \begin{pmatrix} x^{10} + x^8 + x^6 + x^2 + 1 & x^9 + x^4 + x^3 + x^2 + x + 1 \\ x^9 + x^4 + x^3 + x^2 + x + 1 & x^8 + x^6 + x^2 \end{pmatrix}$$

Further, from the proof of Proposition 5.2 we are able to deduce the following recurrence relation:

Corollary 5.2. *Let $b_n \dots b_1 b_1 \dots b_n \in V$ be a palindrome of length $2n$. Then, for $0 \leq i \leq n$, the square root p_i of the upper left entry of $H(b_i \dots b_1 b_1 \dots b_i)$ is given by*

$$p_i = \begin{cases} 1, & \text{if } i = 0; \\ x + b_1 + 1, & \text{if } i = 1; \\ (x + b_i)p_{i-1} + p_{i-2}, & \text{if } 1 < i \leq n. \end{cases}$$

Based on our experimental results, we observed that besides palindromic collisions, for every input parameter $q(x)$, we obtain also non-palindromic collisions of bitstrings of the same length. The following proposition confirms the observation.

Proposition 5.4. *Let $v \in V$ be a palindrome. Then $h(0v0) = h(1v1)$ if and only if $h(0v1) = h(1v0)$.*

Proof. From Lemma 5.1, if $v \in V$ is a palindrome, then $H(v) = \begin{pmatrix} a & b \\ b & d \end{pmatrix}$ for $a, b, d \in \mathbb{F}[x]$, i.e., $H(v)$ is symmetric. From Proposition 5.3, it follows that

$$H(0v0) + H(1v1) = C_0 H(v) C_0 + C_1 H(v) C_1 = \begin{pmatrix} a & a \\ a & a \end{pmatrix} = a \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}.$$

Similar computation yields:

$$H(0v1) + H(1v0) = C_0 H(v) C_1 + C_1 H(v) C_0 = \begin{pmatrix} 0 & a \\ a & 0 \end{pmatrix} = a \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

Therefore,

$$h(0v0) = h(1v1) \iff a = 0 \pmod{q} \iff h(0v1) = h(1v0)$$

where q is input irreducible polynomial. □

5.3.3 Maximal length chains in the Euclidean algorithm

A palindrome v of length $2n$ for which $\rho(v) = 0$ modulo the irreducible polynomial $q(x)$, produces a collision $h(0v0) = h(1v1)$ in $SL_2(\mathbb{F}_2[x]/(q(x)))$. From Corollary 5.1, it follows that $\rho(v) = \begin{pmatrix} a^2 & a^2 \\ a^2 & 0 \end{pmatrix}$ for some $a \in \mathbb{F}_2[x]$. However, since we require that $\rho(v) = 0$ modulo $q(x)$, it must be that the polynomial $q(x)$ is a divisor of polynomial $a(x)$. Based on the same Corollary, $\deg(a) = |v|/2 = n$. Since $q(x)$ is irreducible polynomial which divides $a(x)$ and since $\deg(a) = \deg(q) = n$, it must be that $a(x) = q(x)$. Finally, we are able to conclude that for a given irreducible polynomial as an input parameter, $\rho(v) = \begin{pmatrix} q^2 & q^2 \\ q^2 & 0 \end{pmatrix}$ and that q^2 is the upper left entry of $H(v)$.

Example 5.4. *Palindrome $v = 0011001100$ from Example 5.3, has $\rho(v) = \begin{pmatrix} q^2 & q^2 \\ q^2 & 0 \end{pmatrix}$ where $q(x) = x^5 + x^4 + x^3 + x + 1$, i.e., $\rho(v) = 0 \pmod{q(x)}$, and produces collision: $h(0v0) = h(1v1)$, i.e., $h(000110011000) = h(100110011001)$.*

Corollary 5.2 gives the recurrence relation which connects the square root p_i of the upper left entry of $H(v)$, i.e., $q(x)$ and bits of the palindrome $v = b_i \dots b_1 b_1 \dots b_i \in V$. To be able to deduce bits $b_i \dots b_1 b_1 \dots b_i$, it is sufficient to know the second polynomial, namely p_{i-1} . The second polynomial $p_{i-1}(x) \in \mathbb{F}_2[x]$ must be of degree $n - 1$ and such that $\gcd(q(x), p_{i-1}(x)) = 1$ and such that in the Euclidean algorithm with input $(q(x), p_{i-1}(x))$, the successive quotients are all of degree 1, and the degree of each remainder is one less than the degree of the respective divisor. The second polynomial produces a "Euclidean chain" of maximal length. The existence of polynomial p_{i-1} is assured by the following Proposition by Mesirov and Sweet:

Proposition 5.5 (Mesirov and Sweet [29]). *Given any irreducible polynomial q of degree n over \mathbb{F}_2 , there is a sequence of polynomials p_n, p_{n-1}, \dots, p_0 with $p_n = q$ and $p_0 = 1$, and additionally, the degree of p_i is equal to i and $p_i \equiv p_{i-2} \pmod{p_{i-1}}$.*

Once the second polynomial is known, the Euclidean algorithm will uniquely complete the sequence $p_n = q, p_{n-1}, \dots, p_1, p_0 = 1$ and also provide the linear quotients $x + \beta_i$ ($i = 1, \dots, n$) which allow us to derive the bits b_i of the palindrome in Corollary 5.2. The bits can be computed as $b_1 = \beta_1 + 1$, since $p_1 = x + b_1 + 1$ and as $b_i = \beta_i$ for $i > 1$, i.e., the bit β_1 has to be inverted. This yields the collision

$$h(0\beta_n \dots \overline{\beta_1} \overline{\beta_1} \dots \beta_n 0) = h(1\beta_n \dots \overline{\beta_1} \overline{\beta_1} \dots \beta_n 1),$$

where $\overline{\beta_1}$ indicates the inversion of β_1 .

Mesirov and Sweet [29], prove the Proposition 5.5 and the proof contains an actual algorithm for constructing the second polynomial p_{i-1} . The algorithm produces exactly two polynomials with the above described properties and they induce two collisions.

We give the algorithm as it is described in [29]:

- 1) Construct a matrix $A \in \mathbb{F}_2^{(n+1) \times n}$ from the $n + 1$ polynomials

$$\begin{aligned} g_0 &= x^0 \bmod q(x), \\ g_i &= x^{i-1} + x^{2i-1} + x^{2i} \bmod q(x), \quad \text{for } i = 1, 2, \dots, n, \end{aligned}$$

placing in the i^{th} row of A the coefficients $a_{i,0}, a_{i,1}, \dots, a_{i,n-1}$ of the polynomial

$$g_i = a_{i,0} + a_{i,1}x + \dots + a_{i,n-1}x^{n-1}.$$

- 2) Solve the linear system $Au^t = (1, 0, \dots, 0, 1)^t$ where $u = (u_1, u_2, \dots, u_n)$.
- 3) Compute $p(x)$ by multiplying $q(x)$ by $\sum_{i=1}^n u_i x^{-i}$ and taking only the non-negative powers of x .

Before giving collisions for the challenge parameters, we illustrate the algorithm

for an input irreducible polynomial of small degree n . The algorithm will provide palindrome collisions of length $2n + 2$ for the Tillich-Zémor hash function. Small n will result in a system of small number of equations and short Euclidean maximal chains. Note that when n is small, we could also preform a brute force to discover collisions.

Example 5.5. *With irreducible polynomial $q(x) = x^5 + x^3 + x^2 + x + 1$ used to define the field $\mathbb{F}_{2^5} = \mathbb{F}_2[x]/(q(x))$ and with Tillich-Zémor generators s_0, s_1 we discover palindromic collisions of words of length $2n+2$ by applying above described procedures. Given a polynomial $q(x)$ we first form polynomials $g_i, i \in \{0, \dots, 5\}$:*

$$\begin{aligned} g_0 &= 1 \bmod q(x) = 1 \\ g_1 &= 1 + x + x^2 \bmod q(x) = 1 + x + x^2, \\ g_2 &= x + x^3 + x^4 \bmod q(x) = x + x^3 + x^4, \\ g_3 &= x^2 + x^5 + x^6 \bmod q(x) = x + x^2 + x^4, \\ g_4 &= x^3 + x^7 + x^8 \bmod q(x) = x + x^4, \\ g_5 &= x^4 + x^9 + x^{10} \bmod q(x) = 1 + x^3. \end{aligned}$$

We place in the i^{th} row of matrix A the coefficients $a_{i,0}, a_{i,1}, a_{i,2}, a_{i,3}, a_{i,4}$ of the polynomial $g_i = a_{i,0} + a_{i,1}x + a_{i,2}x^2 + a_{i,3}x^3 + a_{i,4}x^4$. The linear system of equations $Au^t = (1, 0, \dots, 0, 1)^t$ is given by:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}.$$

By solving the above system of linear equations in u_1, \dots, u_5 , we obtain two solutions: $(u_1, \dots, u_5) \in \{(1, 0, 1, 0, 0), (1, 1, 0, 0, 1)\}$.

First, assume that $(u_1, \dots, u_5) = (1, 0, 1, 0, 0)$ and compute $p(x)$ by multiplying $q(x)$ by $\sum_{i=1}^5 u_i x^{-i}$ and taking only the non-negative powers of x :

$$(x^{-1} + x^{-3})(x^5 + x^3 + x^2 + x + 1) = x^4 + x + x^{-2} + x^{-3}$$

Thus, $p(x) = x^4 + x$.

The Euclidean algorithm with $q(x) = x^5 + x^3 + x^2 + x + 1$ and $p(x) = x^4 + x$ produces the following sequence of equalities:

$$\begin{aligned} x^5 + x^3 + x^2 + x + 1 &= (x + \mathbf{0})(x^4 + x) + x^3 + x + 1, \\ x^4 + x &= (x + \mathbf{0})(x^3 + x + 1) + x^2, \\ x^3 + x + 1 &= (x + \mathbf{0})(x^2) + x + 1, \\ x^2 &= (x + \mathbf{1})(x + 1) + 1, \\ x + 1 &= (x + \mathbf{1})(1) + 0. \end{aligned}$$

The sequence yields the following sequence of linear quotients: $x + 0, x + 0, x + 0, x + 1, x + 1$ and reveals bits $(\beta_1, \beta_2, \beta_3, \beta_4, \beta_5) = (1, 1, 0, 0, 0)$. This yields the collision

$$h(0\beta_5\beta_4\beta_3\beta_2\bar{\beta}_1\bar{\beta}_1\beta_2\beta_3\beta_4\beta_50) = h(1\beta_5\beta_4\beta_3\beta_2\bar{\beta}_1\bar{\beta}_1\beta_2\beta_3\beta_4\beta_51),$$

i.e., $h(000010010000) = h(100010010001)$.

We derive the second palindromic collision by taking $(u_1, \dots, u_5) = (1, 1, 0, 0, 1)$.

Then, $q(x) \sum_{i=1}^5 u_i x^{-i}$ becomes:

$$(x^{-1} + x^{-2} + x^{-5})(x^5 + x^3 + x^2 + x + 1) = x^4 + x^3 + x^2 + 1 + x^{-2} + x^{-3} + x^{-4} + x^{-5}.$$

It follows that $p(x) = x^4 + x^3 + x^2 + 1$. The Euclidean algorithm with $q(x) = x^5 + x^3 + x^2 + x + 1$ and $p(x) = x^4 + x^3 + x^2 + 1$ produces the following list of equalities:

$$\begin{aligned} x^5 + x^3 + x^2 + x + 1 &= (x + \mathbf{1})(x^4 + x^3 + x^2 + 1) + x^3, \\ x^4 + x^3 + x^2 + 1 &= (x + \mathbf{1})(x^3) + x^2 + 1, \\ x^3 &= (x + \mathbf{0})(x^2 + 1) + x, \\ x^2 + 1 &= (x + \mathbf{0})(x) + 1, \\ x &= (x + \mathbf{0})(1) + 0. \end{aligned}$$

From the linear quotients we derive bits $(\beta_1, \dots, \beta_5) = (0, 0, 0, 1, 1)$ which yields the second palindromic collision: $h(011001100110) = h(111001100111)$.

From these two collisions of palindromes of length $2n + 2$ and based on Proposition 5.4, we derive two more collisions of words of length $2n + 2$ by inverting appropriate bits:

$$\begin{aligned} h(000010010001) &= h(100010010000) \\ h(011001100111) &= h(111001100110) \end{aligned}$$

5.4 Collisions for the challenge parameters

Our final task was to derive collisions for the challenge parameters. We implemented our attack in the computer algebra system Magma [3] on a standard PC. For each choice of $\mathbb{F}_{2^n} = \mathbb{F}_2[x]/(p(x))$ we obtain two bitstrings $v_1, v_2 \in \{0, 1\}^n$ with

$$h(0v_iv_i^r0) = h(1v_iv_i^r1) \quad (i = 1, 2),$$

i. e., we obtain two collisions of bitstrings of length $2n + 2$. For simplicity, below we restrict to listing one bitstring v_1 for each challenge parameter—the value v_2 can be obtained by reversing v_1 followed by inverting the first and last bits. To specify our solutions v_1 , we use hexadecimal notation where each hexadecimal digit represents 4 bits (0 – 0000, 1 – 0001, \dots , E – 1110, F – 1111). Spaces are for readability only.

A collision for $SL_2(\mathbb{F}_2[x]/(x^{127} + x + 1))$

Here we may choose

$$v_1 = \quad 8000 \quad 0000 \quad 0000 \quad 0003 \quad 0000 \quad 0000 \quad 0000 \quad 000$$

followed by the three bit sequence 000.

A collision for $SL_2(\mathbb{F}_2[x]/(x^{251} + x^7 + x^4 + x^2 + 1))$

Here we may choose

$$v_1 = \quad 4451 \quad 04E5 \quad 4DAB \quad 26EB \quad 91D3 \quad 5201 \quad 0EBD \quad E579 \quad 54F7 \quad AE10 \\ \quad 0959 \quad 713A \quad EC9A \quad B654 \quad E411 \quad 44$$

followed by the three bit sequence 011.

A collision for $SL_2(\mathbb{F}_2[x]/(x^{509} + x^8 + x^7 + x^3 + 1))$

Here we may choose

$$v_1 = \quad 10BB \quad E68D \quad B808 \quad 2B84 \quad 9A1C \quad 569C \quad 9043 \quad 7170 \quad 8D98 \quad E3EB \\ \quad C923 \quad 4CF8 \quad 44F4 \quad 552C \quad 8B49 \quad 1D45 \quad 25C4 \quad 9689 \quad A551 \quad 7910 \\ \quad F996 \quad 249E \quad BE38 \quad CD88 \quad 7476 \quad 1049 \quad CB51 \quad C2C9 \quad 0EA0 \quad 80ED \\ \quad 8B3E \quad E84$$

followed by the single bit 1.

A collision for $SL_2(\mathbb{F}_2[x]/(x^{1021} + x^5 + x^2 + x + 1))$

Here we may choose

$v_1 =$	7EDE	B9C6	F43F	3707	050D	36F7	0DA4	C665	CD36	41ED
	101D	F09A	258F	8C09	1176	82FF	42A1	6475	21B2	8901
	143D	DB01	10FE	FD61	C4A9	C498	4005	0C28	F705	C7DA
	6449	1D97	CDC4	9132	DF1D	0778	A185	0010	C91C	A91C
	35FB	F844	06DD	E144	048A	6C25	7134	2A17	FA0B	7444
	818F	8D22	C87D	C045	BC13	659D	3319	2D87	7B65	8507
	0767	E17B	1CEB	DBF						

followed by the single bit 1.

A collision for

$SL_2(\mathbb{F}_2[x]/(x^{2039} + x^{10} + x^9 + x^8 + x^7 + x^5 + x^4 + x^2 + 1))$

Here we may choose

$v_1 =$	5DB1	31E2	BFD6	5D34	A98C	7FEF	8049	6043	1918	8835
	7F23	1BEF	CF42	391A	E5AF	A211	BACE	74DF	F1B3	4B0D
	372F	1A17	4D0C	FE33	6064	292E	790A	57C7	DF43	5E17
	E424	49EA	3BE4	C978	3D58	1F53	ECDA	DE3A	6B60	06DC
	5EDD	8E80	E201	B9C8	23A7	0998	3521	A78D	8D49	1239
	8700	9071	2D47	943F	A369	C3C9	ABF7	7E05	FC66	FA4E
	607C	0D22	433E	8368	42F9	8489	607C	0CE4	BECC	7F40
	FDDF	AB27	872D	8BF8	53C5	691C	1201	C338	9125	6363
	CB09	5833	21CB	8827	3B00	8E02	E376	F476	C00D	ACB8
	F6B6	6F95	F035	783D	264F	B8AF	2448	4FD0	F585	F7C7
	D4A1	3CE9	284C	0D98	FE61	65D0	B1E9	D961	A59B	1FF6
	5CE6	BB10	8BEB	4EB1	3885	E7EF	B189	FD58	2231	3184
	0D24	03EF	FC63	2A59	74D7	FA8F	191B	7		

followed by the three bit sequence 011.

Based on the previous results we conclude that neither the Tillich-Zémor hash function from CRYPTO '94 nor its variants from ICECS '08 and CRT-RSA '09 should be used in applications where collision resistance is essential.

Conclusions

We analyzed possible applications of the generalized discrete logarithm problem in non-abelian groups in cryptography and drew several conclusions.

Special care must be taken to ensure that the GDLP is hard in the carrier group, its representation and particular generators. We have seen that for the group $PSL(2, p)$, p prime number, generated by two elements and its representation on matrices $SL(2, p)$, GDLP can be solved efficiently provided that at least one of the generators is of order p or that we are able to write efficiently a word of order p in terms of the given generators. As such, group $PSL(2, p)$ in the mentioned representation, generated by special generating pairs is not suitable for cryptographic applications whose security relies on the intractability of the GDLP.

Under the assumption that the GDLP and conjugacy search problems are hard in the underlying group, Computational Diffie-Hellman Problem and Decision Diffie-Hellman Problem can be generalized and a Diffie-Hellman like key exchange protocol and ElGamal like encryption scheme constructed.

Finally, after our cryptanalytic attack on the Tillich-Zémor hash function which is defined in non-abelian group, we conclude that it is not collision resistant and as such should not be used in the cryptographic applications where collision resistance of hash function is essential.

Bibliography

- [1] Kanat S. Abdukhalikov, and Chul Kim. On the Security of the Hashing Scheme Based on SL_2 . In S. Vaudenay, editor, *Fast Software Encryption – FSE ’98*, volume 1372 of *Lecture Notes in Computer Science*, pages 93-102. *Springer-Verlag*, 1998.
- [2] I. F. Blake, R. Fuji-Hara, R. C. Mullin, and S.A. Vanstone). Computing logarithms in finite fields of characteristic two, *SIAM J. Discrete Math. and Appl.*, 5 (1984), 276-285.
- [3] Wieb Bosma, John Cannon, and Catherine Playoust. The Magma algebra system. I. The user language. *Journal of Symbolic Computation*, 24(3-4):235-265, 1997.
- [4] Chris Charnes and Josef Pieprzyk. Attacking the SL_2 hashing scheme. In J. Pieprzyk and R. Safavi-Naini, editors, *Advances in Cryptology-ASIACRYPT ’94*, volume 917 of *Lecture Notes in Computer Science*, pages 322-330. *Springer-Verlag*, 1995.
- [5] Leo G. Chouinard II, Robert Jajcay and Spyros S. Magliveras. Finite Groups and Designs, Handbook of Combinatorial Designs, C.J. Colbourn and J. H. Dinitz editors, *Chapman & Hall / CRC* ISBN 1-58488-506-8, (2007), pp. 819-847.

- [6] Leonard Eugene Dickson with an introduction by Wilhelm Magnus. Linear Groups with an exposition of the Galois field theory. *Dover Publications, Inc.*, New York, 1958.
- [7] Whitfield Diffie, Martin Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22, 1976, pp. 644-654.
- [8] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, **31**(1985), 469-472.
- [9] Walter Feit, Characters of Finite groups, *W.A. Benjamin, Inc.*, New York, 1967.
- [10] Willi Geiselmann. A Note on the Hash Function of Tillich and Zémor. In C. Boyd, editor, *Cryptography and Coding*, volume 1025 of *Lecture Notes in Computer Science*, pages 257-263. *Springer-Verlag*, 1995.
- [11] D. Gorenstein, Finite groups. *Harper & Row*, New York, 1968.
- [12] Markus Grassl, Ivana Ilić, Spyros Magliveras, Rainer Steinwandt. Cryptanalysis of the Tillich-Zémor hash function. *Journal of Cryptology*, 2010. Cryptology ePrint Archive: Report 2009/376, 2009. Available at: <http://eprint.iacr.org/2009/376>
- [13] M. Hall, Jr.. The theory of groups. *Macmillan*, New York, 1959.
- [14] Derek Holt, Bettina Eick, Eamonn A. O'Brien. Handbook of computational group theory. *Chapman & Hall / CRC Press*, Boca Raton, 2005.
- [15] B. Huppert. Endliche Gruppen. *Springer*, 1967.

- [16] Ivana Ilić, Spyros S. Magliveras. Weak discrete logarithms in non-abelian groups. *Journal of Combinatorial Math. and Comb. Computing (JCMCC)*, **74** (2010), pp. 3-11.
- [17] Ivana Ilić, Spyros S. Magliveras. Crypto applications of combinatorial group theory. To appear in: Information security and related combinatorics. *IOS Press*, Amsterdam, 2010.
- [18] Wolfgang Lempken, Spyros S. Magliveras, Tran van Trung and Wandi Wei. A public key cryptosystem based on non-abelian finite groups. *J. Cryptology*, **22**, (2009) pp. 62-74.
- [19] D. E. Littlewood. The Theory of Group Characters. 2nd edition, *Clarendon Press*, Oxford, 1958.
- [20] Lee C. Klingler, Spyros S. Magliveras, Fred Richman, Michal Sramka. Discrete logarithms for finite groups. *Computing*, (2009) 85 pp. 3-19.
- [21] N. Koblitz. A Course in Number Theory and Cryptography. Second Edition. *Springer-Verlag*, 1994.
- [22] S. S. Magliveras and N. D. Memon. The Algebraic Properties of Cryptosystem PGM. *J. of Cryptology*, 5 (1992), pp. 167-183.
- [23] S. S. Magliveras, P. Svaba, Tran van Trung and P. Zajac. On the security of a realization of cryptosystem MST3. *Tatra Mt. Publ.* 41 (2008), pp. 1-13.
- [24] S. S. Magliveras, Tran van Trung and D.R. Stinson. New approaches to designing public key cryptosystems using one-way functions and trap-doors in finite groups. *J. Cryptology*, 15, (2002), pp. 285-297.

- [25] Ayan Mahalanobis. Diffie-Hellman key exchange protocol, its generalizations and nilpotent groups, Ph.D. dissertation, Florida Atlantic University, Boca Raton, FL, 2005.
- [26] Ayan Mahalanobis. A simple generalization of the ElGamal cryptosystem to non-abelian groups. *Comm Algebra* 36(10):3878-3889, 2008.
- [27] Ayan Mahalanobis. The Diffie-Hellman key exchange protocol, and non-abelian nilpotent groups. *Isr. Jr. Math* 165:161-187, 2008.
- [28] Alfred Menezes, Paul C. van Oorschot, Scott A. Vanstone. Handbook of Applied Cryptography. *CRC Press*, 1996.
- [29] Jill P. Mesirov and Melvin M. Sweet. Continued Fraction Expansions of Rational Expressions with Irreducible Denominators in Characteristic 2. *Journal of Number Theory*, 27:144-148, 1987.
- [30] Giacomo de Meulenaer, Christophe Petit, and Jean-Jacques Quisquater. Hardware Implementations of a Variant of the Zémor-Tillich Hash Function: Can a Provable Secure Hash Function be very efficient?, May 2009. Available at <http://eprint.iacr.org/2009/229>.
- [31] R. C. Mullin, J. L. Yucas, and G. L. Mullen. A generalized counting and factoring method for polynomials over finite fields. *Proc. Fortieth Southeastern International Conference*, Submitted March 2009.
- [32] R. C. Mullin and A. Mahalanobis. An alternative representation of finite fields. *Utilitas Math* 67 (2005), 305-318.

- [33] R. C. Mullin. A combinatorial proof of the existence of finite fields. *Amer. Math. Monthly* 71 (1964), 901-902. Reprinted in Selected Papers on Algebra Math. Ass'n of America (1977), 231-233.
- [34] Special Volume: Contemporary Mathematics 225, Finite Fields: Theory Applications, and Algorithms, *Proc. Fourth International Conference on Finite Fields*, Aug. 12- 15, 1997 Ronald C. Mullin and Gary L. Mullen eds.
- [35] Andrew Odlyzko. Discrete logarithms in finite fields and their cryptographic significance. In *Advances in Cryptology-EUROCRYPT'84*, LNCS 219, pp. 224-314, *Springer Verlag*, 1985.
- [36] Andrew Odlyzko. Discrete logarithms: The past and the future. *Designs, Codes, and Cryptography*, 19:129-145, 2000.
- [37] Seong-Hun Paeng, Kil-Chan Ha, Jae Heon Kim, Seongtaek Chee, and Choonsik Park. New public key cryptosystem using finite non-abelian groups. *Crypto 2001* (J. Kilian, ed.), LNCS, vol. 2139, *Springer-Verlag*, 2001, pp. 470-485.
- [38] D. S. Passman. *Permutation Groups*. W.A. Benjamin, Inc., New York, 1968.
- [39] Christophe Petit and Kristin Lauter and Jean-Jacques Quisquater. Cayley Hashes: A Class of Efficient Graph-based Hash Functions. Preprint, 2007. Available at: <http://www.dice.ucl.ac.be/~petit/files/Cayley.pdf>
- [40] Christophe Petit and Jean-Jacques Quisquater and Jean-Pierre Tillich and Gilles Zémor. Hard and Easy Components of Collision Search in the Zémor-Tillich Hash Function: New Attacks and Reduced Variants with Equivalent Security. In M. Fischlin, editor, *Topics in Cryptology-CT-RSA 2009*, volume 5473 of *Lecture Notes in Computer Science*, pages 182-194. *Springer-Verlag*, 2009.

- [41] Christophe Petit and Nicolas Veyrat-Charvillon and Jean-Jacques Quisquater. Efficiency and Pseudo-Randomness of a Variant of Zémor-Tillich Hash Function. In *IEEE International Conference on Electronics, Circuits, and Systems ICECS 2008*, 2008.
- [42] Joseph J. Rotman. An Introduction to the Theory of Groups. *Springer-Verlag* New York, Berlin, Heidelberg, 4th ed., 1995.
- [43] Joseph J. Rotman. Advanced Modern Algebra. *Prentice Hall*, Upper Saddle River, NJ, 2002.
- [44] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. on Computing*, **26**(5), pp. 1484-1509 (1997).
- [45] V. Shpilrain and G. Zapata. Combinatorial group theory and public key cryptography. *Applicable Algebra in Engineering, Communication and Computing* 17 (2006), 291-302.
- [46] V. Shpilrain. Cryptanalysis of Stickel’s key exchange scheme, in: Computer Science in Russia 2008, *Lecture Notes Comp. Sc.* 5010 (2008), 283288.
- [47] Michal Sramka. New Results in Group Theoretic Cryptology, Ph.D. Thesis, Florida Atlantic University, Boca Raton, FL 2006.
- [48] Michal Sramka. On the Security of Stickel’s Key Exchange Scheme. *Journal of Combinatorial Mathematics and Combinatorial Computing* 66 (2008), pp. 151-159.

- [49] E. Stickel. A New Method for Exchanging Secret Keys. In. *Proc. of the Third International Conference on Information Technology and Applications (ICITA05)* 2(2005), 426-430.
- [50] Rainer Steinwandt, Markus Grassl, Willi Geiselmann, Thomas Beth. Weaknesses in the $SL(\mathbb{F}_{2^n})$ Hashing Scheme. In Bellare, editor, *Advances in Cryptology-CRYPTO 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 287-299. *Springer-Verlag*, 2000.
- [51] Douglas R. Stinson. Cryptography: Theory and Practice, 2nd ed, *CRC Press*, New York, NY, 2002.
- [52] Michio Suzuki. Group Theory I. *Springer-Verlag*, New York, 1982.
- [53] Edlyn Teske. Square-Root Algorithms for the Discrete Logarithm Problem (A Survey). *Public-Key Cryptography and Computational Number Theory*, Walter de Gruyter, Berlin-New York, 2001, pp. 283-301.
- [54] Jean-Pierre Tillich and Gilles Zémor. Group-theoretic hash functions. In G. D. Cohen and S. Litsyn and A. Lobstein and G. Zémor, editors, *Lecture Notes in Computer Science*, pages 90-110. *Springer-Verlag*, 1994.
- [55] Jean Pierre Tillich, Gilles Zémor. Hashing with SL_2 . In Y. Desmedt, editor, *Advances in Cryptology-CRYPTO'94*, volume 839 of *Lecture Notes in Computer Science*, pages 40-49, 1994.
- [56] M. I. Gonzlez Vasco, M. Rotteler, and R. Steinwandt. On Minimal Length Factorizations of Finite Groups. *Experimental Mathematics*, 12(1): 1-12, 2003.

- [57] M. I. González Vasco and R. Steinwandt. Chosen ciphertext attacks as common vulnerability of some group- and polynomial-based encryption schemes. *Tatra Mountains Mathematical Publications*, vol. 33, pp. 149-157, 2006.
- [58] M. I. González Vasco, C. Martínez, and R. Steinwandt. Towards a Uniform Description of Several Group Based Cryptographic Primitives. *Designs, Codes and Cryptography*, vol. 33, pp. 215-226, 2004.
- [59] H. Wielandt. Finite Permutation Groups. *Academic Press*, 1964.
- [60] Gilles Zémor. Hash Functions and Graphs With Large Girths. In D. W. Davies, editor, *Advances in Cryptology – EUROCRYPT '91*, volume 547 of *Lecture Notes in Computer Science*, pages 508–511. *Springer-Verlag*, 1991.
- [61] Gilles Zémor. Hash Functions and Cayley Graphs. *Designs, Codes and Cryptography*, 4(4):381–394, October 1994.

Appendix A

Appendix on group $PSL(2, q)$ actions

Assume that q is a prime power. We consider two cases, namely $q \equiv 1 \pmod{4}$, and $q \equiv 3 \pmod{4}$. In either case, the normalizers of cyclic subgroups of order $(q-1)/2$ and $(q+1)/2$ are dihedral D_{q-1} and D_{q+1} of orders $(q-1)$ and $(q+1)$ respectively. The distribution and frequencies within conjugacy classes of the elements of D_{q-1} and D_{q+1} are easy to determine, so an application of Theorem 2.3, item (c) yields the induced characters $\theta = 1 \uparrow_{D_{q-1}}^G$ and $\phi = 1 \uparrow_{D_{q+1}}^G$.

Note that columns 4, 5 and 6 of tables in the Appendix correspond to several conjugacy classes of elements of G , and the number of such classes is given in row 5 of the tables. The first row describes canonical representatives of the classes, row 2 the orders of elements in the classes, row 3 the orders of centralizers of elements in classes and row 4 the class sizes. Rows 7 and 9 give the values of the characters θ and ϕ of the action of G on the conjugacy classes of cyclic subgroups of order $(q-1)/2$ and $(q+1)/2$ respectively. These are computed from rows 6 and 8 respectively and the formula given in Theorem 2.3, (c).

x	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & \lambda \end{pmatrix}$	$\begin{pmatrix} 0 & 1 \\ -1 & \lambda \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$
$ x $	1	2	all divisors $d \neq 2$ of $(q-1)/2$	all divisors $d \neq 2$ of $(q+1)/2$	p
σ_x	$q(q^2-1)/2$	$(q-1)$	$(q-1)/2$	$(q+1)/2$	q
κ_x	1	$q(q+1)/2$	$q(q+1)$	$q(q-1)$	$(q^2-1)/2$
$\# K_i$	1	1	$(q-5)/4$	$(q-1)/4$	2
h_x	1	$(q+1)/2$	2	0	0
θ	$q(q+1)/2$	$(q+1)/2$	1	0	0
h'_x	1	$(q+1)/2$	0	2	0
ϕ	$q(q-1)/2$	$(q-1)/2$	0	1	0

Table A.1: $PSL(2, q)$ for $q \equiv 1 \pmod{4}$

Proposition A.1. *Suppose that q is a prime power, $q \equiv 1 \pmod{4}$, and let $G = PSL(2, q)$. Let X be the collection of all cyclic subgroups of order $(q-1)/2$ of G , Y the collection of all cyclic subgroups of order $(q+1)/2$ of G , and let G act on X and Y by conjugation. Then,*

- (i) *Each of X and Y constitute of a single conjugacy class of subgroups,*
- (ii) *There are exactly $3(q+3)/4$ G -orbits on $X \times X$,*
- (iii) *There are exactly $3(q-1)/4$ G -orbits on $Y \times Y$, and*
- (iv) *There are $(3q+1)/4$ G -orbits on $X \times Y$.*

Proof. The number of G -orbits on X is $(\theta, [1])$, and the number of G -orbits on Y is $(\phi, [1])$. Since $(\theta, [1]) = (\phi, [1]) = 1$ we have an independent proof that there is just one conjugacy class of cyclic subgroups of order $(q-1)/2$ and one conjugacy class of cyclic subgroups of order $(q+1)/2$. Moreover, $(\theta \cdot \theta, [1]) = (\theta, \theta)$ gives the number of G -orbits on $X \times X$. Similarly, (ϕ, ϕ) gives the number of conjugacy classes on pairs of cyclic subgroups of order $(q+1)/2$, and (θ, ϕ) the number of G -orbits on $X \times Y$. Direct computation yields that $(\theta, \theta) = 3(q+3)/4$, $(\phi, \phi) = 3(q-1)/4$ and $(\theta, \phi) = (3q+1)/4$. Hence, the result. \square

We proceed to study the case $q \equiv 3 \pmod{4}$.

x	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & \lambda \end{pmatrix}$	$\begin{pmatrix} 0 & 1 \\ -1 & \lambda \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$
$ x $	1	2	all divisors $d \neq 2$ of $(q-1)/2$	all divisors $d \neq 2$ of $(q+1)/2$	p
σ_x	$q(q^2 - 1)/2$	$(q + 1)$	$(q - 1)/2$	$(q + 1)/2$	q
κ_x	1	$q(q - 1)/2$	$q(q + 1)$	$q(q - 1)$	$(q^2 - 1)/2$
$\# K_i$	1	1	$(q - 3)/4$	$(q - 3)/4$	2
h_x	1	$(q - 1)/2$	2	0	0
θ	$q(q + 1)/2$	$(q + 1)/2$	1	0	0
h'_x	1	$(q + 3)/2$	0	2	0
ϕ	$q(q - 1)/2$	$(q + 3)/2$	0	1	0

Table A.2: $PSL(2, q)$ for $q \equiv 3 \pmod{4}$

For $q \equiv 3 \pmod{4}$ a proposition analogous to Proposition A.1 takes the following form:

Proposition A.2. *Suppose that q is a prime power, $q \equiv 3 \pmod{4}$, and let $G = PSL(2, q)$. Let X be the collection of all cyclic subgroups of order $(q - 1)/2$ of G , Y the collection of all cyclic subgroups of order $(q + 1)/2$ of G , and let G act on X and Y by conjugation. Then,*

- (i) *Each of X and Y constitute of a single conjugacy class of subgroups,*
- (ii) *There are exactly $(3q + 7)/4$ G -orbits on $X \times X$,*
- (iii) *There are exactly $3(q + 1)/4$ G -orbits on $Y \times Y$, and*
- (iv) *There are $3(q + 1)/4$ G -orbits on $X \times Y$.*

Proof. The number of G -orbits on X is $(\theta, [1])$, and the number of G -orbits on Y is $(\phi, [1])$. Since $(\theta, [1]) = (\phi, [1]) = 1$ we again have that there is just one conjugacy class of cyclic subgroups of order $(q - 1)/2$ and one conjugacy class of cyclic subgroups

of order $(q+1)/2$. Moreover, $(\theta \cdot \theta, [1]) = (\theta, \theta)$ gives the number of G -orbits on $X \times X$. Similarly, (ϕ, ϕ) gives the number of conjugacy classes on pairs of cyclic subgroups of order $(q+1)/2$, and (θ, ϕ) the number of G -orbits on $X \times Y$. Direct computation yields that $(\theta, \theta) = (3q+7)/4$ $(\phi, \phi) = 3(q+1)/4$ and $(\theta, \phi) = 3(q+1)/4$. \square