

AN EXPLORATION OF DIFFIE-HELLMAN PERMUTATIONS

A Thesis
Submitted to
the Temple University Graduate Board

in Partial Fulfillment
of the Requirements for the Degree of
MASTER OF SCIENCE

by
Tommy Pham
2026

Examining Committee Members:

Dr. Vasily Dolgushev, Department of Mathematics, Temple University
Dr. Yury Grabovsky, Department of Mathematics, Temple University
Mr. Oleg Naumenko, Hideez Group Inc.
Dr. Joseph Picone, College of Engineering, Temple University
Dr. Gillian Queisser, Department of Mathematics, Temple University
Dr. Ari Shnidman, Department of Mathematics, Temple University

©
Copyright
2026

by

Tommy Pham

All Rights Reserved

ABSTRACT

For a prime integer $p \geq 7$, one can construct various families of permutations of degree $p - 1$. We use tools from group theory and number theory, together with experimental results, to study these families of permutations. In this thesis, we investigate whether an unknown exponent can be recovered from an output of a permutation called τ_p when a formula for computing τ_p is known. Our approach explores whether precomposing a suitable composition of permutations with τ_p can help recover the exponent. We prove that two particular permutations on $\{1, \dots, p - 1\}$ generate the symmetric group S_{p-1} . This result may help us in identifying a particular composition of permutations that aids in recovering the desired exponent. We also propose an authentication protocol based on a specific family of permutations, derive an upper bound on the maximum number of fixed points, and prove that the ratio of this upper bound to $p - 1$ tends to 0 as $p \rightarrow \infty$. Consequently, the probability that an adversary succeeds by choosing an exponent uniformly at random tends to 0. Our computational results further indicate that the observed fixed-point counts are substantially smaller than the theoretical upper bound.

To my parents, I would like to sincerely thank you for your support. I wholeheartedly appreciate your sacrifices and unwavering belief in me.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to my thesis advisor, Professor Vasily Dolgushev. His insights and suggestions shaped the development of this project.

I would also like to thank him for his patience, guidance, and encouragement.

I would like to thank Sophia V. Breslavets, Nathan Graff and Igor Shparlinski for useful conversations.

I am grateful to the members of my committee, who devoted their time to carefully reading my thesis and providing valuable feedback.

I am grateful to all my good friends, who have always created an environment that motivates me to broaden my horizons.

TABLE OF CONTENTS

	Page
ABSTRACT	iii
DEDICATION	iv
ACKNOWLEDGEMENTS	v
LIST OF TABLES	vii
LIST OF FIGURES	viii
CHAPTER	
1 INTRODUCTION	1
1.1 Notational Conventions	4
2 THE DISCRETE LOG PROBLEM	7
2.1 The Diffie–Hellman Key Exchange Protocol	9
2.2 Reducing The Discrete Logarithm Problem	11
2.3 Baby Steps, Giant Steps	19
2.4 The Index-Calculus Method	22
3 DIFFIE-HELLMAN PERMUTATIONS	27
3.1 What Are Diffie-Hellman Permutations?	27
3.2 Unlocking Permutations	36
4 AN AUTHENTICATION PROTOCOL BASED ON DH PERMUTATIONS . . .	41
5 CONCLUSION	52
REFERENCES CITED	55

LIST OF TABLES

LIST OF FIGURES

Figure	Page
3.1 Normalized maximizing pairs $\left(\frac{j}{p-1}, \frac{k}{p-1}\right)$ obtained for the first 30 safe primes satisfying $p \geq 7$	39
4.2 Values of $\frac{100\nu(p)}{p-1}$ for 10,000 primes in the range $55,001 \leq p \leq 171,049$	44
4.3 Values of $\frac{100\nu(p)}{p-1}$ for 6,000 safe primes in the range $74,027 \leq p \leq 1,629,899$	44
4.4 Values of $\nu(p)$ for primes $11 \leq p \leq 224,771$	51

CHAPTER 1

INTRODUCTION

The Discrete Logarithm Problem (DLP) (see Chapter 2) is one of the challenging problems related to primitive roots modulo a prime integer p .

For a prime p , we let

$$X := \{1, 2, \dots, p-1\}$$

and we call elements of X *exponents*.

We denote by τ_p the following permutation of degree $p-1$:

$$\tau_p(x) := y,$$

where y is the unique integer in X satisfying the congruence

$$y \equiv g_p^x \pmod{p}, \tag{1.1}$$

and g_p is the smallest primitive root modulo p .

It is clear that, when p , g_p , and y are known, solving the equation $y = \tau_p(x)$ for x is equivalent to solving the DLP for the multiplicative group $(\mathbb{Z}/p\mathbb{Z})^\times$. We call τ_p the *key Diffie-Hellman (DH) permutation*.

By Theorem 3.2, for any prime $p \geq 7$, the permutations μ_2 and $\rho := (1, 2, \dots, p-1)$ generate the symmetric group S_{p-1} . Here

$$\mu_2(x) := y \in \{1, 2, \dots, p-1\},$$

where y satisfies the congruence

$$y \equiv 2x \pmod{p}.$$

We construct a family of permutations of degree $p-1$ by precomposing and postcomposing the key Diffie-Hellman permutation τ_p with permutations in the group generated by μ_2 and ρ . We call this family of permutations the *Diffie-Hellman (DH) permutations* [3].

Thanks to Theorem 3.2 we know that there exist permutations $h \in \langle \mu_2, \rho \rangle$ such that the equation

$$h \circ \tau_p(x) = x$$

is satisfied for a relatively large number of exponents x .

In this thesis, we present evidence from computational experiments that the pairs (j, k) for which the permutation

$$\rho^j \circ \mu_2 \circ \rho^k \circ \tau_p$$

has the greatest number of fixed points satisfying the congruence

$$j + 2k \equiv c \pmod{p-1}$$

for some $c \in \mathbb{Z}/(p-1)\mathbb{Z}$.

The authentication protocol proposed in this thesis is based on the family of DH permutations

$$\{\rho^c \circ \tau_p \mid 0 \leq c \leq p-2\}.$$

We will see that the presence of fixed points in the permutations $\rho^c \circ \tau_p$ is a potential vulnerability in the proposed authentication protocol. We derive an upper bound on the maximum number of fixed points in the permutations $\rho^c \circ \tau_p$, $1 \leq c \leq p-2$, and prove that the ratio of this upper bound to $p-1$ tends to 0 as $p \rightarrow \infty$. The proof of this result is independent of the choice of the primitive root b modulo p . Therefore, we may replace the primitive root g_p in the definition of τ_p with any primitive root b modulo p .

We present computational results that indicate that the observed fixed-point counts are substantially smaller than the theoretical upper bound.

The tools that we use in this thesis are from group theory and number theory. We also use computational experiments to support our results. The computations were performed using SageMath.

1.1 Notational Conventions

For a finite set X , we let $\#X$ be the size of X . We denote by S_X the group of bijections from X to itself. For instance, if d is a positive integer, then $S_d := S_{\{1,2,\dots,d\}}$ is the symmetric group on d elements. Let G be a group and g be an element of G , we denote by $\text{ord}(g)$ to be the order of g in G .

We denote by nfp the function returning the number of fixed points of a given permutation. For example, consider a permutation of $\{1, 2, \dots, 13\}$,

$$\pi = (2, 3, \dots, 11, 12),$$

then

$$\text{nfp}(\pi) = 2.$$

For a positive integer n and a , we denote by $\text{rem}(a, n)$ the non-negative remainder of a when divided by n . Further, we reserve the notation $\underline{\text{rem}}(a, n)$ for the modified remainder of a when divided by n , which is defined as follows:

$$\underline{\text{rem}}(a, n) = \begin{cases} \text{rem}(a, n), & \text{if } n \nmid a, \\ n, & \text{if } n \mid a. \end{cases} \quad (1.2)$$

Throughout this thesis, we shall denote by $\bar{a}$ the equivalence class of a in $\mathbb{Z}/n\mathbb{Z}$ when the modulus n is clear from the context. We also use Greek letters for residue classes and Latin letters for integers.

For an integer $n > 1$ and an integer a coprime to n , we denote by $\text{ord}_n(a)$ the order of the equivalence class $\bar{a}$ in the multiplicative group $(\mathbb{Z}/n\mathbb{Z})^\times$. For example, let b be a primitive root modulo p for a prime p , then $\text{ord}_p(b) = p - 1$.

We also denote by ϕ the Euler's phi function, which is defined as follows:

$$\phi(n) = \#\{a \in \mathbb{Z} : 1 \leq a \leq n, \gcd(a, n) = 1\}.$$

For a commutative ring R with identity, we denote by $\text{Aff}(R)$ the semi-direct product of $R \rtimes R^\times$, where $R^\times$ is the multiplicative group of units in R and R is the additive group of R . Elements in $\text{Aff}(R)$ are of the form (c, u) , where $c \in R$ and $u \in R^\times$ and they act on R as follows:

$$(c, u) \cdot x = u \cdot x + c,$$

for all $x \in R$. For two pairs $(c_1, u_1), (c_2, u_2) \in \text{Aff}(R)$, we have

$$(c_1, u_1) \cdot (c_2, u_2) = (c_1 + u_1 c_2, u_1 u_2).$$

It is clear that the group $\text{Aff}(R)$ is isomorphic to the subgroup of $\text{GL}_2(R)$ consisting of matrices of the form

$$\begin{pmatrix} u & c \\ 0 & 1 \end{pmatrix},$$

where $c \in R$ and $u \in R^\times$.

CHAPTER 2

THE DISCRETE LOG PROBLEM

In this section, we introduce the discrete logarithm problem (DLP), which motivates our study of cycle structures and fixed points of the permutations discussed in this thesis.

The DLP is not uniformly difficult: there are natural families of instances that can be solved efficiently. A basic example arises in the additive cyclic group

$$G = \mathbb{Z}/n\mathbb{Z}.$$

An element of G is a residue class modulo n , which we may represent by its least nonnegative residue.

In the additive setting, the DLP is the following. Given $G = \mathbb{Z}/n\mathbb{Z}$, let b be a positive integer so that $\bar{b}$ is a generator of G , and let y be an integer, determine an integer x such that

$$x b \equiv y \pmod{n},$$

where $x b$ denotes repeated addition of b to itself. In $\mathbb{Z}/n\mathbb{Z}$, the element $\bar{b}$ is a generator if and only if $\gcd(b, n) = 1$. When $\gcd(b, n) = 1$, the multiplicative inverse $\bar{b}^{-1}$ exists, and the solution is immediate:

$$\bar{x} = \overline{b^{-1}y}.$$

Moreover, $\overline{b^{-1}}$ can be computed quickly using the extended Euclidean algorithm. For example, take $G = \mathbb{Z}/(10^9 + 7)\mathbb{Z}$, $b = 5$, and $y = 23$. Since $5^{-1} \equiv (4 \times 10^8 + 3) \pmod{10^9 + 7}$, we obtain

$$x \equiv (4 \times 10^8 + 3) \cdot 23 \equiv (2 \times 10^8 + 6) \pmod{10^9 + 7},$$

so the discrete logarithm of 23 to the base 5 in G is $x = (2 \times 10^8 + 6)$.

Let G be a cyclic group written multiplicatively. Given a generator $b \in G$ and an element $y \in G$, the (multiplicative) *discrete logarithm problem (DLP)* is to compute an integer x such that

$$b^x = y.$$

As a concrete example, consider the cyclic group of order $10^9 + 6$, namely the multiplicative group

$$(\mathbb{Z}/(10^9 + 7)\mathbb{Z})^\times.$$

Using SageMath, one verifies that 5 is a primitive root modulo $(10^9 + 7)$. We then consider the congruence

$$5^x \equiv 23 \pmod{(10^9 + 7)}.$$

Solving this congruence by brute force on a MacBook Pro equipped with an Apple M1 Pro chip and 16 GB of RAM required approximately 14.15 minutes of computation. Unlike the additive case, this computation is not immediate. Moreover, we expect that solving the discrete logarithm problem for primes with significantly more digits would require days or longer using brute-force methods. In practice, cryptographic applications typically employ primes with at least 100 digits, rendering such approaches entirely infeasible.

This example highlights an important distinction: although all cyclic groups of a given order are isomorphic as abstract groups (for instance, $\mathbb{Z}/(p-1)\mathbb{Z} \cong (\mathbb{Z}/p\mathbb{Z})^\times$ for prime p), their computational behavior can differ dramatically depending on how the group is presented. Consequently, while cyclic groups may be algebraically identical, their suitability for cryptographic applications depends crucially on the computational difficulty of problems such as the discrete logarithm problem.

Note that the **DLP** is not always difficult, as we can see shortly, some algorithms can break the DLP into computable pieces.

2.1 The Diffie–Hellman Key Exchange Protocol

Imagine Alice and Bob wish to share a secret key for use in a symmetric cipher, but their only means of communication is over an insecure public channel. Every message they exchange can be observed by an eavesdropper, traditionally called Eve. Under these circumstances, how can Alice and Bob establish a shared secret? The Diffie–Hellman key exchange protocol provides a solution to this problem.

Let p be a large prime (in practice, we are interested in primes that have at least 100 digits) and let b be a primitive root modulo p . Alice and Bob publicly agree on the pair (p, b) , for instance by posting it on a public website. Alice then chooses a secret exponent $1 < x_A < p - 1$, and Bob chooses a secret exponent $1 < x_B < p - 1$. The pair (x_A, x_B) constitutes the private part of the key and is never revealed over the public channel.

Alice computes

$$A \equiv b^{x_A} \pmod{p},$$

while Bob computes

$$B \equiv b^{x_B} \pmod{p}.$$

Alice sends A to Bob, and Bob sends B to Alice. Although Eve can observe the values of A and B , she does not know the secret exponents x_A and x_B .

Using the received value, Alice computes

$$A' \equiv B^{x_A} \equiv (b^{x_B})^{x_A} \pmod{p},$$

and Bob computes

$$B' \equiv A^{x_B} \equiv (b^{x_A})^{x_B} \pmod{p}.$$

Since $A' \equiv B' \pmod{p}$, Alice and Bob obtain the same value, which serves as their shared secret key.

If Eve attempts to recover this secret key, she must determine either x_A from

$$y \equiv b^{x_A} \pmod{p}$$

or x_B from

$$y' \equiv b^{x_B} \pmod{p}.$$

Both problems are instances of the discrete logarithm problem (DLP), which is believed to be computationally infeasible for sufficiently large primes p . Consequently, the security of the Diffie–Hellman key exchange protocol relies on the assumed hardness of the discrete logarithm problem.

2.2 Reducing The Discrete Logarithm Problem

We begin by introducing a straightforward method for reducing the **DLP** in a cyclic group G to instances of the **DLP** in its subgroups. Before doing so, however, we first establish several assumptions that clarify what we mean by an *efficient* computation in this context.

We assume that multiplication and inversion of elements in G can be performed efficiently. In some settings, however, determining whether two group elements are equal is not immediately obvious. For example, consider the class group of a quadratic order $\text{Cl}(\Delta)$, whose elements may be represented by binary quadratic forms in

$$f(x, y) = ax^2 + bxy + cy^2, \quad \text{where } a, b, c \in \mathbb{Z}$$

with discriminant $\Delta = b^2 - 4ac$.

Two such forms f and g are considered to represent the same element in the (proper) class group if there exists a matrix

$$\gamma = \begin{pmatrix} p & q \\ r & s \end{pmatrix} \in \text{SL}_2(\mathbb{Z})$$

such that

$$g(x, y) = f(px + qy, rx + sy).$$

Consequently, two forms with different coefficients may nonetheless correspond to the same group element. Equality testing in such groups therefore requires additional computation, typically involving reduction to a canonical representative.

For the purposes of our analysis, we shall assume that the cost of determining whether two elements of G are equal is of the same order of magnitude as performing a group operation. Finally, we also assume that group elements admit representations that allow them to be compared and sorted.

Now let G be a cyclic group of order n , let g be a generator of G , and let $t \in G$. Thus, $G = \langle g \rangle$. We are interested in finding an integer x , determined modulo n , such that

$$g^x = t.$$

Suppose that n admits a factorization $n = ab$ with $\gcd(a, b) = 1$. Then the subgroup $\langle g^a \rangle$ has order b , while the subgroup $\langle g^b \rangle$ has order a . We are interested in solving the following system of equations:

$$(g^a)^{h_a} = t^a, \quad (g^b)^{h_b} = t^b.$$

The powers g^a , t^a , g^b , and t^b can be computed efficiently via binary exponentiation. Let us provide an example of computing

$$x^m,$$

where $x \in G$ and m is a positive integer, using binary exponentiation.

Let $x \in G$ and let m be a positive integer. Write the binary expansion of m as

$$m = \sum_{i=0}^t \epsilon_i 2^i, \quad \epsilon_i \in \{0, 1\}, \quad t := \lfloor \log_2 m \rfloor.$$

Beginning with x , we successively square to get

$$x, \quad x^2, \quad x^4, \quad \dots, \quad x^{2^t}.$$

We then multiply only those powers for which $\epsilon_i = 1$. Hence,

$$x^m = \prod_{\substack{0 \leq i \leq t \\ \epsilon_i = 1}} x^{2^i}.$$

This procedure requires at most $\lfloor \log_2 m \rfloor$ squarings and at most $\lfloor \log_2 m \rfloor$ additional group operations. Thus, it takes approximately $O(\log m)$ group operations to compute x^m . Therefore, computing the values of

$$g^a, t^a, g^b, \text{ and } t^b$$

takes roughly

$$O(2 \log a + 2 \log b) = O(\log ab) = O(\log n)$$

group operations.

After computing these powers and solving the two discrete logarithm problems above, we combine the resulting exponents using Bézout's theorem. Since $\gcd(a, b) = 1$, there exist integers s and u such that

$$sa + ub = 1.$$

It follows that

$$t = t^{sa+ub} = (t^a)^s (t^b)^u = g^{ah_a s} g^{bh_b u} = g^{ah_a s + bh_b u},$$

and therefore the solution to the **DL** of t is given by $ah_a s + bh_b u$.

For the next reduction, we consider the case in which the order of G is a prime power p^e , where $e > 1$. In fact, the argument presented below does not rely on the primality of p ; however, we may assume that p is prime without loss of generality due to the first reduction, since any positive integer can be factored into prime powers. We will see that this reduction reduces the discrete logarithm problem in G to a sequence of discrete logarithm problems in a cyclic subgroup of order p .

Consider a group G of order p^e , where $e > 1$, and suppose we wish to solve the equation

$$g^x = y. \tag{2.1}$$

If x denotes the least nonnegative integer satisfying (2.1), then x admits a base- p expansion

$$x = b_0 + b_1p + \cdots + b_{e-1}p^{e-1},$$

where each digit $b_i \in \{0, 1, \dots, p-1\}$. Our goal is to determine the coefficients b_i , which allows us to reduce the original discrete logarithm problem to a sequence of problems in smaller groups, potentially making the computation more tractable.

We begin by raising both sides of (2.1) to the power p^{e-1} :

$$y^{p^{e-1}} = (g^x)^{p^{e-1}} = g^{b_0p^{e-1} + b_1p^e + \cdots + b_{e-1}p^{2(e-1)}}.$$

Since $|G| = p^e$, all terms involving powers of p^e vanish, and we obtain

$$y^{p^{e-1}} = g^{b_0 p^{e-1}} = (g^{p^{e-1}})^{b_0}.$$

Thus, determining b_0 reduces to solving a **DLP** in the cyclic subgroup generated by $g^{p^{e-1}}$.

Now suppose that we have already computed $b_0, b_1, \dots, b_{i-1}$. To determine b_i , define

$$y_i = y g^{-b_0 - b_1 p - \dots - b_{i-1} p^{i-1}}.$$

Substituting $y = g^x$, we obtain

$$y_i = g^{x - (b_0 + b_1 p + \dots + b_{i-1} p^{i-1})} = g^{b_i p^i + \dots + b_{e-1} p^{e-1}}.$$

Raising y_i to the power p^{e-i-1} yields

$$(y_i)^{p^{e-i-1}} = g^{b_i p^{e-1} + \dots + b_{e-1} p^{2(e-1)-i}} = g^{b_i p^{e-1}} = (g^{p^{e-1}})^{b_i},$$

again using the fact that $|G| = p^e$. Consequently, $(y_i)^{p^{e-i-1}}$ lies in the cyclic subgroup generated by $g^{p^{e-1}}$, and solving the resulting discrete logarithm problem recovers the coefficient b_i .

Let us provide a numerical illustration of the reduction methods described above.

Example. We consider the congruence

$$5^x \equiv 23 \pmod{97}.$$

Note that the order of the multiplicative group $(\mathbb{Z}/97\mathbb{Z})^\times$ is 96. Since

$$96 = 3 \cdot 32,$$

we may reduce the **DLP** to subgroups of order 3 and 32. Thus, the original problem reduces to solving the two congruences

$$(5^3)^{x_1} \equiv 42, \quad (5^{32})^{x_2} \equiv 61 \pmod{97},$$

where $42 \equiv 23^3 \pmod{97}$ and $61 \equiv 23^{32} \pmod{97}$.

For b_1 , Our goal is to determine x_1 and x_2 . We first consider x_1 . Since 5^3 generates a subgroup of order $32 = 2^5$, we apply the prime-power reduction described above. Writing

$$x_1 = b_0 + 2b_1 + 4b_2 + 8b_3 + 16b_4,$$

we raise both sides of $(5^3)^{x_1} \equiv 42$ to the power $16 = 2^{5-1}$ to obtain

$$42^{16} \equiv ((5^3)^{16})^{b_0} \pmod{97}.$$

A direct computation shows that

$$42^{16} \equiv 96 \pmod{97} \quad \text{and} \quad (5^3)^{16} \equiv 96 \pmod{97}.$$

Since $96 \equiv -1 \pmod{97}$ and the subgroup generated by $(5^3)^{16}$ has order 2, it follows immediately that $b_0 = 1$.

Next, to determine b_1 , we consider

$$1 \equiv ((5^3)^{16})^{b_1} \pmod{97}.$$

Solving this congruence yields $b_1 = 0$.

Proceeding in the same manner, we obtain $b_2 = 1$, $b_3 = 1$, and $b_4 = 0$. Hence,

$$x_1 = b_0 + 2b_1 + 4b_2 + 8b_3 + 16b_4 = 1 + 4 + 8 = 13.$$

Indeed, $(5^3)^{13} \equiv 42 \pmod{97}$.

For x_2 , note that 5^{32} generates a subgroup of order 3. In particular, determining x_2 requires checking at most three possibilities. By direct computation, we find $x_2 = 2$.

Finally, applying the extended Euclidean algorithm, we obtain integers u, v such that

$$u \cdot 3 + v \cdot 32 = 1, \quad \text{for instance } u = 11, v = -1.$$

Therefore,

$$x \equiv u \cdot 3 \cdot x_1 + v \cdot 32 \cdot x_2 = 11 \cdot 3 \cdot 13 - 32 \cdot 2 \equiv 77 \pmod{96}.$$

Thus, we have reduced the **DLP** above to discrete logarithm problems in smaller subgroups, which can often be solved more efficiently. The key assumption, however, is that the order of the group is known and admits a useful factorization. In many practical settings, the group is chosen to have (nearly) prime order, so the method described above offers little or no advantage.

2.3 Baby Steps, Giant Steps

The reduction method discussed above exploits the fact that the order of the group is composite, allowing for a nontrivial factorization and a corresponding decomposition of the discrete logarithm problem into smaller instances. However, when the group has prime order, such reductions offer no advantage. In this setting, one must instead rely on generic algorithms whose complexity depends essentially on the size of the group. A classical example of such an algorithm is the **Baby-Step Giant-Step** method.

Let $G = \langle g \rangle$ be a cyclic group of order t , and consider the discrete logarithm problem

$$g^x = y.$$

Set $s = \lceil \sqrt{t} \rceil$. Since $0 \leq x \leq t - 1$, we may write

$$x = a_0 + a_1 s,$$

where $0 \leq a_0 < s$ and $0 \leq a_1 < s$.

Our goal is therefore reduced to determining the integers a_0 and a_1 .

The *baby steps* consist of computing and storing the list

$$1, g, g^2, \dots, g^{s-1}.$$

The *giant steps* consist of computing the elements

$$y, yg^{-s}, yg^{-2s}, \dots, yg^{-(s-1)s}.$$

If $x = a_0 + a_1s$, then

$$g^x = g^{a_0+a_1s} = g^{a_0}(g^s)^{a_1}.$$

Since $g^x = y$, we obtain

$$g^{a_0} = yg^{-a_1s}.$$

Thus, there must exist indices p and q such that

$$g^p = yg^{-qs},$$

with $0 \leq p, q < s$. Comparing exponents, we conclude

$$x \equiv p + qs \pmod{t}.$$

Therefore, once a collision between the baby-step and giant-step lists is found, the discrete logarithm x can be recovered as $x = p + qs \pmod{t}$.

Normally, one first computes the baby-step list, namely

$$g, g^2, \dots, g^{s-1},$$

which can be done in $s - 1$ group operations. Next, to begin the giant-step phase, one performs two additional operations to form yg^{-s} . Each time a new giant-step element is computed, we simply look it up (and compare) against the precomputed baby-step list, stopping as soon as a match is found. On average, this occurs after checking about half of the giant-step list.

One may wonder whether a collision can be found efficiently. The answer is affirmative. Since we have assumed that the elements of the group can be labeled and sorted, we may first sort the baby-step list. In practice, a more efficient approach is to store the baby-step values in a dictionary (or hash table). After constructing this data structure, each lookup corresponding to a giant step can be performed in expected time $O(1)$. Building the dictionary requires $O(m)$ time, and we perform at most m lookups. Hence, the total running time to detect a collision is $O(m)$, which is considered highly efficient in comparison with the naive $O(m^2)$ approach.

2.4 The Index-Calculus Method

The exponential algorithms we discussed in 2.2 and in 2.3 have runtimes on the order of the square root of the group order. However, in some specific groups, we have additional information that could be useful for DL computations. Specifically, this requires the concept of smooth numbers, which we will introduce shortly. When smooth representatives of group elements can often be found efficiently, it may be possible to use a subexponential algorithm to perform DLs known as the index-calculus method.

Definition 2.1. *A positive integer is called B -smooth if all of its prime factors are less than or equal to the chosen bound B .*

For example, let $B = 10$. Then

$$25 = 5 \times 5$$

is B -smooth, since every prime factor of 25 is less than or equal to B . On the other hand,

$$26 = 13 \times 2$$

is not B -smooth, since it has a prime factor, namely 13, that is greater than B .

Now, we discuss the index calculus method.

Let $p = 2q + 1$, where q is a prime. Let $\mathbb{F}_p^\star$ be the multiplicative group of the finite field $\mathbb{F}_p$, and let g be a primitive root modulo p . Thus, g has order $p - 1$. Given p , g , and y , the discrete logarithm problem in $\mathbb{F}_p^\star$ is to solve the equation

$$g^x = y. \tag{2.2}$$

Since $\bar{g}$ has order $p - 1$, the map

$$\begin{aligned} \varphi_g : \mathbb{Z}/(p-1)\mathbb{Z} &\xrightarrow{\sim} (\mathbb{Z}/p\mathbb{Z})^\times, \\ \bar{x} &\longmapsto \overline{g^x} \end{aligned}$$

is a group isomorphism. Its inverse is the map

$$\begin{aligned} \mathcal{L}_g : (\mathbb{Z}/p\mathbb{Z})^\times &\mapsto \mathbb{Z}/(p-1)\mathbb{Z}, \\ \bar{y} &\longmapsto \bar{x}, \end{aligned}$$

where x is determined modulo $p - 1$ by the congruence

$$y \equiv g^x \pmod{p}.$$

We first choose a smoothness bound B . This bound determines the factor base

$$\mathcal{B} = \{p_1, p_2, \dots, p_k\},$$

where the p_i are the primes less than or equal to B . If B is too small, then the factor base $\mathcal{B}$ is also small, which makes the linear algebra part easier. However, it becomes much harder to find relations, since B -smooth numbers may be rare. On the other hand, if B is too large, then finding relations becomes easier because there are more B -smooth numbers, but the factor base becomes larger. This means that we need more relations, and the linear algebra becomes more expensive.

In the asymptotic analysis of the index calculus method described in [?, Section 3.3], a common rough choice for B is

$$B \approx \exp \left(\frac{1}{\sqrt{2}} \sqrt{\log p \log \log p} \right).$$

There are two main stages in the index calculus algorithm. The first stage is to gather approximately $\ell = \frac{B}{\ln(B)}$ relations with respect to the chosen bound B . We choose random integers s in the range $1 \leq s \leq p - 2$ until we find enough cases where $g^s \pmod{p}$ is B -smooth. For each such value of s , we can write

$$g^s \equiv p_1^{s_1} p_2^{s_2} \cdots p_k^{s_k} \pmod{p}.$$

Equivalently,

$$s_i \equiv s_{1,i} \mathcal{L}_g(p_1) + s_{2,i} \mathcal{L}_g(p_2) + \cdots + s_{k,i} \mathcal{L}_g(p_k) \pmod{p-1}.$$

After collecting enough independent relations, we can use "linear algebra" to solve for the unknowns

$$\mathcal{L}_g(p_1), \mathcal{L}_g(p_2), \dots, \mathcal{L}_g(p_k)$$

and here is how we proceed. We can write the relations in matrix form as

$$\begin{pmatrix} s_{1,1} & s_{1,2} & \cdots & s_{1,k} \\ s_{2,1} & s_{2,2} & \cdots & s_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ s_{\ell,1} & s_{\ell,2} & \cdots & s_{\ell,k} \end{pmatrix} \begin{pmatrix} \mathcal{L}_g(p_1) \\ \mathcal{L}_g(p_2) \\ \vdots \\ \mathcal{L}_g(p_k) \end{pmatrix} = \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_\ell \end{pmatrix} \pmod{2q}.$$

By the Chinese Remainder Theorem, the group $\mathbb{Z}/(2q)\mathbb{Z}$ is isomorphic to the group $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/q\mathbb{Z}$. Therefore, we can solve the above system of equations separately modulo 2 and modulo q . The values of $\mathcal{L}_g(p_i)$ modulo 2 can be computed directly from the rules:

$$\mathcal{L}_g(p_i) \equiv \begin{cases} 0 \pmod{2}, & \text{if } p_i^q \equiv 1 \pmod{p}, \\ 1 \pmod{2}, & \text{if } p_i^q \equiv -1 \pmod{p}. \end{cases}$$

The system of equations modulo q can be solved using Gaussian elimination since q is prime and so $\mathbb{Z}/q\mathbb{Z}$ is a field. Now, we reconstruct the values of $\mathcal{L}_g(p_i)$ modulo $2q$ using the Chinese Remainder Theorem. Suppose that

$$\mathcal{L}_g(p_i) \equiv a_i \pmod{2} \quad \text{and} \quad \mathcal{L}_g(p_i) \equiv b_i \pmod{q}.$$

Then,

$$\mathcal{L}_g(p_i) \equiv a_i + b_i \pmod{2q}.$$

This precomputation step addresses the hardest part of the problem.

Once these values $\mathcal{L}_g(p_i)$, $1 \leq i \leq k$, are computed, we then try to express a multiple of y in terms of the factor base. More precisely, we choose a random integer S until

$$g^S y \equiv p_1^{r_1} p_2^{r_2} \cdots p_k^{r_k} \pmod{p}$$

is B -smooth. Applying $\mathcal{L}_g$ to both sides,

$$S + x \equiv r_1 \mathcal{L}_g(p_1) + r_2 \mathcal{L}_g(p_2) + \cdots + r_k \mathcal{L}_g(p_k) \pmod{p-1}.$$

Therefore,

$$x \equiv r_1 \mathcal{L}_g(p_1) + r_2 \mathcal{L}_g(p_2) + \cdots + r_k \mathcal{L}_g(p_k) - S \pmod{p-1},$$

where the values $\mathcal{L}_g(p_i)$ were computed in the previous step.

CHAPTER 3

DIFFIE-HELLMAN PERMUTATIONS

3.1 What Are Diffie-Hellman Permutations?

Let $p \geq 7$ be prime, and let

$$X := \{1, 2, \dots, p-1\}$$

be the set of exponents. Let g_p denote the least positive primitive root modulo p ; that is, g_p is the smallest positive integer less than p such that

$$\langle g_p + p\mathbb{Z} \rangle = (\mathbb{Z}/p\mathbb{Z})^\times.$$

We identify X with both the ring $\mathbb{Z}/(p-1)\mathbb{Z}$ and the multiplicative group $(\mathbb{Z}/p\mathbb{Z})^\times$ via the natural bijections

$$x \mapsto x + (p-1)\mathbb{Z} \quad \text{and} \quad x \mapsto x + p\mathbb{Z},$$

respectively. We use $\star$ and $\cdot$ to denote the binary operations on X that correspond to the addition and multiplication in $\mathbb{Z}/(p-1)\mathbb{Z}$. Moreover, we use $*$ to denote the binary operation on X that corresponds to the multiplication in $(\mathbb{Z}/p\mathbb{Z})^\times$. Thus, for $x, y \in X$,

$$x \star y := \underline{\text{rem}}(x + y, p-1), \quad x \cdot y := \underline{\text{rem}}(xy, p-1),$$

and

$$x * y := \text{rem}(xy, p),$$

where $\text{rem}(\cdot, n)$ denotes the modified remainder of division by n as in (1.2).

We denote by τ_p the permutation of X defined by

$$\tau_p(x) := \text{rem}(g_p^x, p) \tag{3.1}$$

for all $x \in X$ and we call τ_p the *key DH permutation*.

For example,

$$\tau_7 = (1, 3, 6), \quad \tau_{11} = (1, 2, 4, 5, 10)(6, 9),$$

and

$$\tau_{13} = (1, 2, 4, 3, 8, 9, 5, 6, 12)(7, 11).$$

It is clear that for a given prime p and an integer $1 \leq y \leq p-1$, solving the equation

$$\tau_p(x) = y$$

is equivalent to finding the discrete logarithm of y with respect to the base g_p modulo p .

Remark 3.1. Given a prime p , we denote by $L(p)$ the length of the longest cycle in τ_p .

Further, we define

$$\lambda(p) := \frac{L(p)}{p-1}.$$

Our experiments found that 33 primes in the range $101 \leq p \leq 10009$ satisfy $\lambda(p) \leq 0.3$. This represents approximately 2.74% of all primes in this range. Of these 33 primes, only one, $p = 347$, is a safe prime; that is, $p = 2q + 1$ for some prime q . For such primes, the Diffie–Hellman key exchange protocol discussed in Section 2.1 may be vulnerable to a *cycle attack*, in which an eavesdropper raises the permutation to the power $L(p)$ to recover the secret exponent x .

The set X is equipped with the group action of the group $\text{Aff}_{p-1} := \text{Aff}(\mathbb{Z}/(p-1)\mathbb{Z})$, where an element in Aff_{p-1} is an affine map of the form

$$f_{a,b}(x) := ax + b \pmod{p-1}$$

where $a, b \in \mathbb{Z}/(p-1)\mathbb{Z}$, $\gcd(a, p-1) = 1$ and the residue $0 \pmod{p-1}$ is represented by $p-1$. Moreover, we also have another group action coming from the cyclic group $\mathcal{Z}_{p-1} = \langle \mu_{g_p} \rangle$, where

$$\mu_{g_p}(x) := g_p * x. \tag{3.2}$$

The action of Aff_{p-1} on X induces a homomorphism

$$\Phi: \text{Aff}_{p-1} \longrightarrow S_X,$$

$$f_{a,b} \longmapsto \Phi(f_{a,b}) : x \mapsto f_{a,b}(x) = ax + b \pmod{p-1}, \forall x \in X.$$

We show that Φ is injective by determining its kernel. Suppose that $f_{a,b} \in \ker(\Phi)$. Then $f_{a,b}$ acts as the identity on X , so

$$f_{a,b}(x) = ax + b \equiv x \pmod{p-1} \quad \text{for every } x \in X.$$

Taking $x = 0$, where 0 is represented by $p-1$ in X , gives

$$f_{a,b}(0) = b \equiv 0 \pmod{p-1}.$$

Thus, $b = 0$ in $\mathbb{Z}/(p-1)\mathbb{Z}$. The previous congruence therefore reduces to

$$f_{a,0}(x) = ax \equiv x \pmod{p-1} \quad \text{for every } x \in X.$$

Taking $x = 1$ gives $a \equiv 1 \pmod{p-1}$, and hence $a = 1$ in $\mathbb{Z}/(p-1)\mathbb{Z}$. Therefore,

$$\ker(\Phi) = \{f_{1,0}\}.$$

It follows that Φ is injective, so the action of Aff_{p-1} on X is faithful. Similarly, the action of $\mathcal{Z}_{p-1}$ on X induces a homomorphism

$$\Psi : \mathcal{Z}_{p-1} \longrightarrow S_X,$$

$$\mu_{g_p}^k \longmapsto \Psi(\mu_{g_p}^k) : x \mapsto \mu_{g_p}^k(x), \forall x \in X.$$

We show that

$$\ker(\Psi) = \{id_X\}$$

Trivially,

$$\{id_X\} \subseteq \ker(\Psi).$$

For the reverse inclusion, suppose $\mu_{g_p}^k \in \ker(\Psi)$. Then $\mu_{g_p}^k$ acts as the identity on X . Since g_p is a primitive root modulo p , it has order $p - 1$. Therefore,

$$k \equiv 0 \pmod{p-1},$$

which implies that

$$\mu_{g_p}^k = \mu_{g_p}^0 = id_X.$$

Hence,

$$\ker(\Psi) = \{id_X\},$$

and consequently, Ψ is injective and the action of $\mathcal{Z}_{p-1}$ on X is faithful. Therefore, we may safely identify Aff_{p-1} and $\mathcal{Z}_{p-1}$ with their images under Φ and Ψ , respectively. It is reasonable to consider a subgroup $R_p \leq S_X$ generated by elements of Aff_{p-1} and the permutation μ_{g_p} . Due to the following theorem, we will not need a special notation for the subgroup R_p .

Theorem 3.2. *Let $p \geq 7$ be a prime number. Then the cycle μ_{g_p} and elements of Aff_{p-1} generate the symmetric group S_{p-1} .*

We will need the following lemma and theorem to prove Theorem 3.2:

Lemma 3.3. *Let $p \geq 7$ be a prime number,*

$$\rho := (1, 2, \dots, p-1)$$

and μ_2 is the permutation in S_{p-1} defined by the formula

$$\mu_2(x) := 2 * x.$$

Then,

$$\mu_2 \circ \rho \circ \mu_2^{-1} = (1, 2) \circ \rho^2. \quad (3.3)$$

Proof. Since p is an odd prime, we can write

$$p = 2q + 1$$

for some $q \in \mathbb{Z}_{\geq 1}$. A direct computation shows that

$$\mu_2 = \begin{pmatrix} 1 & 2 & \cdots & q-1 & q & q+1 & q+2 & \cdots & 2q \\ 2 & 4 & \cdots & 2q-2 & 2q & 1 & 3 & \cdots & 2q-1 \end{pmatrix}.$$

Thus,

$$\mu_2 \circ \rho \circ \mu_2^{-1} = \begin{pmatrix} 1 & 2 & \cdots & 2q-2 & 2q-1 & 2q \\ 3 & 4 & \cdots & 2q & 2 & 1 \end{pmatrix}.$$

Further,

$$\rho^2 = (1, 3, \dots, 2q-1)(2, 4, \dots, 2q-2, 2q).$$

Therefore, we have

$$(1, 2) \circ \rho^2 = (1, 3, \dots, 2q-1, 2, 4, \dots, 2q-1, 2q).$$

Hence, the identity (3.3) holds. □

The following classical results can be found in [4, Chapter 4]. We provide our own proof.

Lemma 3.4. *S_d is generated by $(1, 2), (2, 3), \dots, (d-1, d)$.*

Proof. Let G be the subgroup of S_d generated by

$$(1, 2), (2, 3), \dots, (d-1, d).$$

Clearly, $G \subseteq S_d$. We show the reverse inclusion. Let $\pi \in S_d$ be arbitrary and write π as a product of disjoint cycles. Each cycle can be written as a product of transpositions:

$$(i_1, i_2, \dots, i_k) = (i_1, i_2)(i_2, i_3) \cdots (i_{k-1}, i_k).$$

Thus, π can be written as a product of transpositions.

Now consider a transposition (i, j) , where $1 \leq i < j \leq d$. We have

$$(i, j) = (i, i+1)(i+1, i+2) \cdots (j-2, j-1)(j-1, j) \\ \cdot (j-2, j-1) \cdots (i+1, i+2)(i, i+1).$$

Every transposition on the right-hand side is one of the generators of G . Therefore, $(i, j) \in G$ and π is a product of transpositions; it follows that $\pi \in G$.

Since $\pi \in S_d$ was arbitrary, we conclude that $S_d \subseteq G$. Therefore, $G = S_d$. $\square$

We are now ready to prove Theorem 3.2.

Proof of Theorem 3.2. From a simple observation, we see that

$$\mu_{x_1} \circ \mu_{x_2} = \mu_{x_1 * x_2}, \quad x_1, x_2 \in X. \tag{3.4}$$

Since g_p is a primitive root modulo p , there exists a unique integer $0 < t \leq p-2$ such that

$$g_p^t \equiv 2 \pmod{p}.$$

Combining this with the identity (3.4), we obtain

$$\mu_{g_p}^t = \mu_2.$$

Thus, $\mu_2 \in R_p$. Furthermore, $\rho = f_{1,1} \in \text{Aff}_{p-1} \subset R_p$. Hence,

$$(1, 2) = \mu_2 \circ \rho \circ \mu_2^{-1} \circ \rho^{-2} \in R_p.$$

Notice that,

$$(i, i+1) = \rho^k \circ (1, 2) \circ \rho^{-k}, \quad \text{where } k = |i-1|.$$

Due to lemma 3.4, it follows that $\langle \mu_{g_p}, \rho \rangle = S_{p-1}$. □

We now give a formal definition of Diffie-Hellman permutations.

Definition 3.5. Let $p \geq 7$ be prime, τ_p be the key DH permutation defined in (3.1), and R_p be the subgroup of S_{p-1} generated by elements of the groups Aff_{p-1} and $\mathcal{Z}_{p-1} := \langle \mu_{g_p} \rangle$. Furthermore, let L and R be non-empty subsets of R_p . Then, the elements of the set

$$\{a \circ \tau_p \circ b \mid a \in L, b \in R\}$$

are called **Diffie-Hellman permutations** (or **DH permutations**).

For example, take

$$L := \{\sigma_c \mid 1 \leq c \leq p-1\}, \quad R := \{\delta_u \mid 1 \leq u \leq p-1, \gcd(u, p-1) = 1\}, \quad (3.5)$$

where

$$\sigma_c(x) := x \star c, \quad \delta_u(x) := u \cdot x, \quad x \in X := \{1, 2, \dots, p-1\}. \quad (3.6)$$

By using the sets L and R defined above (3.5), we can construct the family of DH permutations as follows:

$$\{\sigma_c \circ \tau_p \circ \delta_u \mid 1 \leq c \leq p-1, 1 \leq u \leq p-1, \gcd(u, p-1) = 1\}. \quad (3.7)$$

Since $g_p + p\mathbb{Z}$ is a generator of the cyclic group $(\mathbb{Z}/p\mathbb{Z})^\times$, the set of generators of $(\mathbb{Z}/p\mathbb{Z})^\times$ is given by

$$\{g_p^u + p\mathbb{Z} \mid 1 \leq u \leq p-2, \gcd(u, p-1) = 1\}.$$

Therefore, the set of DH permutations in (3.7) is precisely the set

$$\{\tau_{p,b,c} \mid b \text{ is a primitive root modulo } p, 1 \leq c \leq p-1\},$$

where

$$\tau_{p,b,c}(x) := y$$

and y is a unique integer satisfying $1 \leq y \leq p-1$ and

$$y \equiv \text{rem}(b^x, p) + c \pmod{p-1}.$$

3.2 Unlocking Permutations

The following corollary is an immediate consequence of Theorem 3.2 and perhaps a "simple solution" to the DLP for the multiplicative group $(\mathbb{Z}/p\mathbb{Z})^\times$:

Corollary 3.6. *Let $p \geq 7$ be a prime, τ_p be the key DH permutation, $\rho := (1, 2, \dots, p-1)$ and μ_{g_p} be the permutation defined in (3.2). Then there exists a unique permutation $h_{unlock} \in \langle \rho, \mu_{g_p} \rangle$ such that*

$$h_{unlock} \circ \tau_p = id_X.$$

We call the permutation h_{unlock} the *perfect unlocking permutation*. Based on our experiments, we conjecture that finding h_{unlock} in practice is extremely difficult.

When looking for a more practical *unlocking* (or *rectifying*) permutation $h \in R_p$, we consider the following three standards:

- The permutation $h \circ \tau_p$ has a relatively large number of fixed points.
- The cycle type of the permutation $h \circ \tau_p$ is relatively long.
- The longest cycle of the permutation $h \circ \tau_p$ is relatively short.

If $h \in R_p$ satisfies one of the above three standards, we call h an *unlocking permutation*.

For a fixed prime p , we introduce the following three success functions ε_{fp} , ε_{lct} , and ε_{llc} :

$$\varepsilon_{fp}, \varepsilon_{lct}, \varepsilon_{llc} : R_p \mapsto \mathbb{Z}_{\geq 0}$$

$$\varepsilon_{fp}(h) := \#\{x \in X \mid h \circ \tau_p(x) = x\}, \quad (3.8)$$

$$\varepsilon_{lct}(h) := \text{number of cycles of the cycle type of } h \circ \tau_p, \quad (3.9)$$

$$\varepsilon_{llc}(h) := \text{length of the longest cycle of } h \circ \tau_p. \quad (3.10)$$

We have been exploring various families of permutations and determining specific subsets for which the success functions ε_{fp} and ε_{lct} have relatively large values, while ε_{uc} has relatively small values. So far, only one of our experiments has shown a clear pattern. We describe this experiment below.

Fix a safe prime $p \geq 7$, and let ε be one of the success functions defined above. In this section, we take $\varepsilon := \varepsilon_{fp}$. For $j, k \in \{1, \dots, p-1\}$, define

$$h_{j,k} := \rho^j \circ \mu_2 \circ \rho^k \circ \tau_p,$$

where ρ and μ_2 are defined as in Lemma 3.3. Our goal is to identify the pairs (j, k) for which $\varepsilon(h_{j,k})$ is as large as possible.

Our numerical experiments with the first 30 safe primes satisfying $p \geq 7$ show an interesting pattern. As shown in Figure 3.1, the normalized pairs

$$\left(\frac{j}{p-1}, \frac{k}{p-1} \right)$$

that maximize $\varepsilon(h_{j,k})$ appear to lie along diagonal lines.

The observed diagonal lines can be described by congruences of the form

$$j + 2k \equiv c \pmod{p-1}.$$

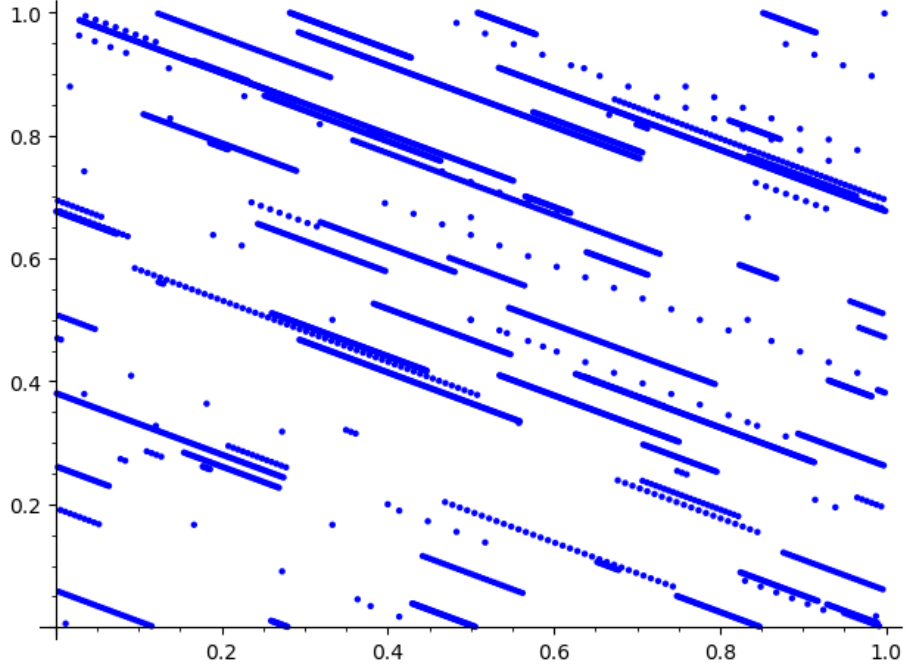


Figure 3.1: Normalized maximizing pairs $\left(\frac{j}{p-1}, \frac{k}{p-1}\right)$ obtained for the first 30 safe primes satisfying $p \geq 7$.

For each fixed prime p , let C_p denote the set of residue classes c arising from the maximizing pairs. For some values of p , the set C_p contains only one residue class, whereas for others it contains several. Our experiments suggest that the number of such residue classes is relatively small.

Although these pairs maximize $\varepsilon(h_{j,k})$, the resulting values are still relatively small compared with p . If

$$M_p := \max_{1 \leq j, k \leq p-1} \varepsilon(h_{j,k}), \quad (3.11)$$

then the largest observed value of

$$\frac{M_p}{p-1}$$

among the primes considered was $\frac{7}{382} \approx 0.0183$.

CHAPTER 4

AN AUTHENTICATION PROTOCOL BASED ON DH PERMUTATIONS

We now present a new authentication protocol based on the DH permutations introduced in Chapter 3 and provide evidence supporting its security.

Alice and Bob agree on a prime p and a primitive root b modulo p . For example, take $b := g_p$. Alice then calculates the following integers:

$$r_x := \text{rem}(b^x, p), \quad c := \text{rem}(x - r_x, p - 1).$$

Alice keeps x secret and publishes the triple

$$(p, b, c).$$

When it is time, Alice sends an important message to Bob. However, Bob does not know whether the message was sent by Alice or by an imposter, say, Eve. To verify the authenticity of the message, Alice sends Bob the integer x (if Alice uses a public channel, she can send x in an encrypted form). Bob then calculates the integer $r_x := \text{rem}(b^x, p)$ and checks whether the following congruence holds:

$$r_x \equiv x - c \pmod{p - 1}. \tag{4.1}$$

For a prime p and a primitive root b modulo p , define

$$C_{p,b}(x) := c, \quad x \in \{1, \dots, p-1\},$$

where

$$c := \text{rem}(x - r_x, p-1).$$

The security of this protocol relies on the assumption that the function $C_{p,b}$ is *pre-image resistant*.

If Eve intercepts the message and attempts to impersonate Alice, she must find an exponent x' such that

$$r_{x'} \equiv x' - c \pmod{p-1}. \quad (4.2)$$

Equivalently, she must solve

$$\text{rem}(b^{x'}, p) \equiv x' - c \pmod{p-1}. \quad (4.3)$$

We believe that neither the method for reducing the DLP discussed in Section 2.2 nor the index calculus method discussed in Section 2.4 helps Eve here, because the right-hand side of the congruence in (4.3) is unknown: it depends on the unknown exponent x' itself. We have also been exploring whether quantum algorithms, such as Shor's algorithm [6], can be used to solve (4.3). So far, we believe that the answer is no. This is because all the methods that we investigated are suitable for solving the congruence

$$b^x \equiv y \pmod{p},$$

where the prime p , the primitive root b modulo p and the integer y are given. In our authentication protocol, the integer y is c , which is defined by

$$c = \text{rem}(x - r_x, p - 1),$$

where x is unknown and so r_x is unknown as well.

For a prime p , we set

$$\nu(p) := \max(\{\text{nfp}(\rho^c \circ \tau_p) \mid 1 \leq c \leq p - 1\}), \quad (4.4)$$

where $\text{nfp}(h)$ denotes the number of fixed points of a permutation h .

Therefore, if Eve chooses an integer x' uniformly at random from $\{1, 2, \dots, p - 1\}$ and sets

$$r_{x'} := \text{rem}(b^{x'}, p),$$

then the probability that

$$r_{x'} \equiv x' - c \pmod{p - 1}$$

is at most

$$\frac{\nu(p)}{p - 1}.$$

Figures 4.2 and 4.3 plot $\frac{100\nu(p)}{p-1}$ on the vertical axis against p on the horizontal axis. Figure 4.2 uses 10,000 primes in the range $55,001 \leq p \leq 171,049$, whereas Figure 4.3 uses 6,000 safe primes in the range $74,027 \leq p \leq 1,629,899$.

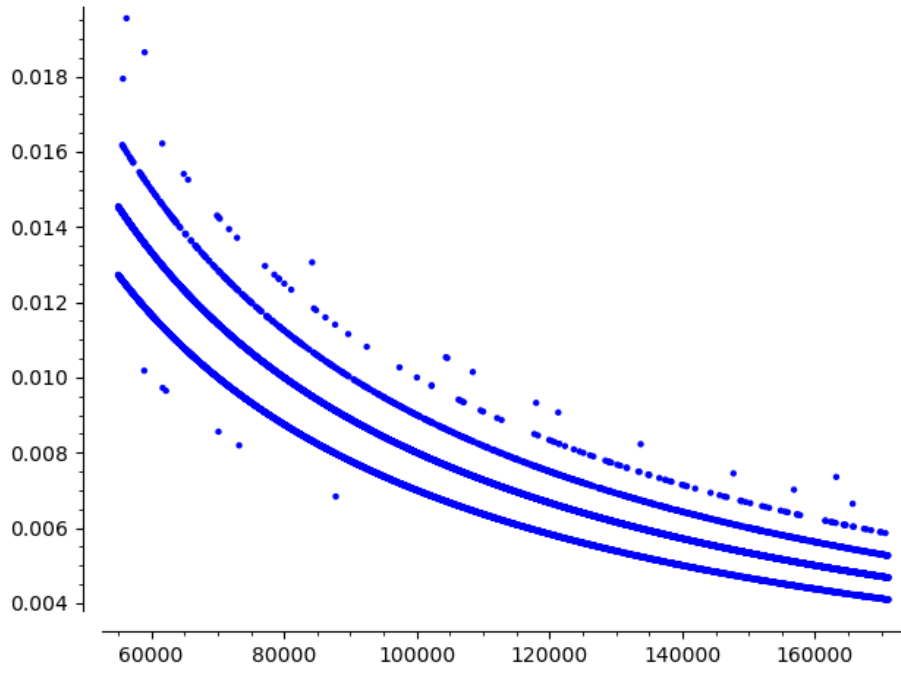


Figure 4.2: Values of $\frac{100\nu(p)}{p-1}$ for 10,000 primes in the range $55,001 \leq p \leq 171,049$.

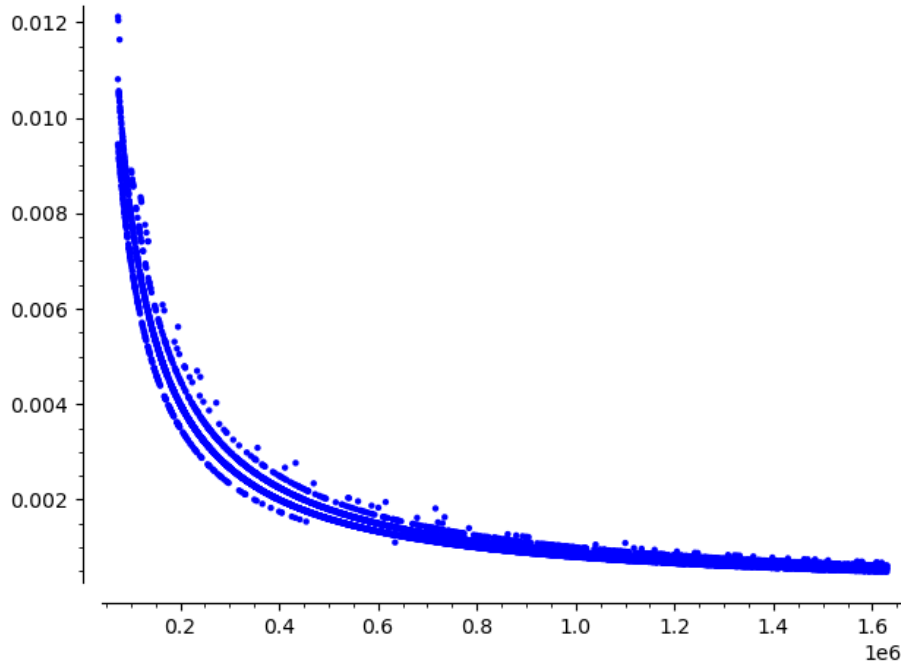


Figure 4.3: Values of $\frac{100\nu(p)}{p-1}$ for 6,000 safe primes in the range $74,027 \leq p \leq 1,629,899$.

Those figures motivate the following theorem:

Theorem 4.1. *Let $p \geq 7$ be a prime integer. Then*

$$\#\nu(p) \leq (p-1)^{1/2} + (p-1)^{1/4} + \frac{1}{2}.$$

Before proving Theorem 4.1, we need to introduce some preliminary definitions and results.

Definition 4.2 (Cilleruelo [11, Section 1]). *A Sidon set is a subset of a semigroup G having the property that all sums of two elements are distinct. Equivalently, if $a, b, c, d \in G$ and $a - b = c - d$, then $\{a, b\} = \{c, d\}$.*

Corollary 4.3 (Cilleruelo [11, Section 2, Corollary 2]). *Let $F \subseteq \{1, \dots, n\}$ be a Sidon set. Then*

$$\#F \leq n^{1/2} + n^{1/4} + \frac{1}{2}.$$

Proof. See [11, Section 2, Corollary 2]. □

Proof of Theorem 4.1. Let p be prime. Let b be a primitive root modulo p , and fix $c \in \{1, 2, \dots, p-1\}$. Define

$$A_c := \{x \in X \mid r_x \equiv x - c \pmod{p-1}\}, \tag{4.5}$$

and

$$S_{p,b,c} := \{r \in \{1, 2, \dots, p-1\} \mid b^{r+c} \equiv r \pmod{p}\}.$$

By definition, A_c is precisely the set of fixed points of $\rho^c \circ \tau_p$.

We first establish a bijection between A_c and $S_{p,b,c}$. Consider the map

$$\begin{aligned}\phi: A_c &\longrightarrow S_{p,b,c}, \\ x &\longmapsto r_x := \text{rem}(b^x, p).\end{aligned}$$

To see that ϕ is well-defined, let $x \in A_c$. By the definition of A_c ,

$$r_x \equiv x - c \pmod{p-1}.$$

Therefore,

$$b^{r_x+c} \equiv b^x \equiv r_x \pmod{p},$$

so $r_x \in S_{p,b,c}$.

We next prove that ϕ is injective. Suppose that $x_1, x_2 \in A_c$ and $\phi(x_1) = \phi(x_2)$. Then

$r_{x_1} = r_{x_2}$, and hence

$$b^{x_1} \equiv b^{x_2} \pmod{p}.$$

Because b has order $p-1$ modulo p , it follows that

$$x_1 \equiv x_2 \pmod{p-1}.$$

Since x_1 and x_2 are both in $\{1, 2, \dots, p-1\}$, we conclude that $x_1 = x_2$.

To prove surjectivity, let $r \in S_{p,b,c}$. Then

$$g_p^{r+c} \equiv r \pmod{p}.$$

Since b is a primitive root modulo p , there exists $x \in X$ such that

$$b^x \equiv r \pmod{p}.$$

Thus, $r_x = r$. Moreover,

$$b^{r+c} \equiv b^x \pmod{p},$$

and therefore

$$r + c \equiv x \pmod{p-1}.$$

Consequently,

$$r_x = r \equiv x - c \pmod{p-1},$$

which shows that $x \in A_c$ and $\phi(x) = r$. Hence, ϕ is surjective. Therefore, ϕ is bijective

and

$$\#A_c = \#S_{p,b,c}.$$

We now show that $S_{p,b,c}$ is a Sidon set. Let $r_1, r_2 \in S_{p,b,c}$ with $r_1 < r_2$, and set

$$d := r_2 - r_1.$$

Since $r_1, r_2 \in S_{p,b,c}$, we have

$$\begin{aligned} b^d &= b^{(r_2+c)-(r_1+c)} \\ &\equiv r_2 r_1^{-1} \pmod{p}. \end{aligned}$$

Multiplying through by r_1 gives

$$b^d r_1 \equiv r_2 \pmod{p}.$$

Since $r_1 = r_2 - d$, this becomes

$$b^d(r_2 - d) \equiv r_2 \pmod{p},$$

and hence

$$r_2(b^d - 1) \equiv b^d d \pmod{p}.$$

Because $1 \leq d \leq p - 2$ and b has order $p - 1$ modulo p ,

$$b^d \not\equiv 1 \pmod{p}.$$

Therefore, $b^d - 1$ is invertible modulo p , and

$$r_2 \equiv b^d d (b^d - 1)^{-1} \pmod{p}.$$

Thus, d uniquely determines r_2 in $\{1, 2, \dots, p-1\}$, and it then uniquely determines $r_1 = r_2 - d$.

Consequently, if $r_3, r_4 \in S_{p,b,c}$, with $r_3 < r_4$, satisfy

$$r_2 - r_1 = r_4 - r_3,$$

then $r_2 = r_4$ and $r_1 = r_3$. Hence, every positive difference of two elements of $S_{p,b,c}$ is distinct, so $S_{p,b,c}$ is a Sidon set.

Applying Corollary 4.3, we obtain

$$\#S_{p,b,c} \leq (p-1)^{1/2} + (p-1)^{1/4} + \frac{1}{2}.$$

□

Corollary 4.4.

$$\lim_{\substack{p \rightarrow \infty \\ p \text{ prime}}} \frac{\nu(p)}{p-1} = 0.$$

Proof. From Theorem 4.1, we have

$$\#S_{p,g_p,c} \leq (p-1)^{1/2} + (p-1)^{1/4} + \frac{1}{2}.$$

This bound is independent of c . Therefore, by the definition of $\nu(p)$,

$$\begin{aligned} 0 \leq \frac{\nu(p)}{p-1} &\leq \frac{(p-1)^{1/2} + (p-1)^{1/4} + \frac{1}{2}}{p-1} \\ &= (p-1)^{-1/2} + (p-1)^{-3/4} + \frac{1}{2(p-1)}. \end{aligned}$$

The right-hand side tends to 0 as $p \rightarrow \infty$. Thus,

$$\lim_{\substack{p \rightarrow \infty \\ p \text{ prime}}} \frac{\nu(p)}{p-1} = 0.$$

□

By Theorem 4.1, if Eve chooses an exponent x uniformly at random from X , then

$$\Pr(x \in A_c) = \frac{\#S_{p,g_p,c}}{p-1} \leq \frac{(p-1)^{1/2} + (p-1)^{1/4} + \frac{1}{2}}{p-1}. \quad (4.6)$$

To investigate how closely the theoretical bound in (4.6) reflects the actual values of $\nu(p)$, we computed the values of $\nu(p)$ for the first 20,000 primes $11 \leq p \leq 224,771$ and produced Figure 4.4.

The numerical results suggest that the theoretical upper bound in (4.6) is far from sharp. As we have seen in Figure 4.4 above, the values of $\nu(p)$ is at most 12 for primes $11 \leq p \leq 224,771$, while the theoretical upper bound grows on the order of $p^{\frac{1}{2}}$. Hence, this suggests there may be a substantially sharper upper bound for $\nu(p)$.

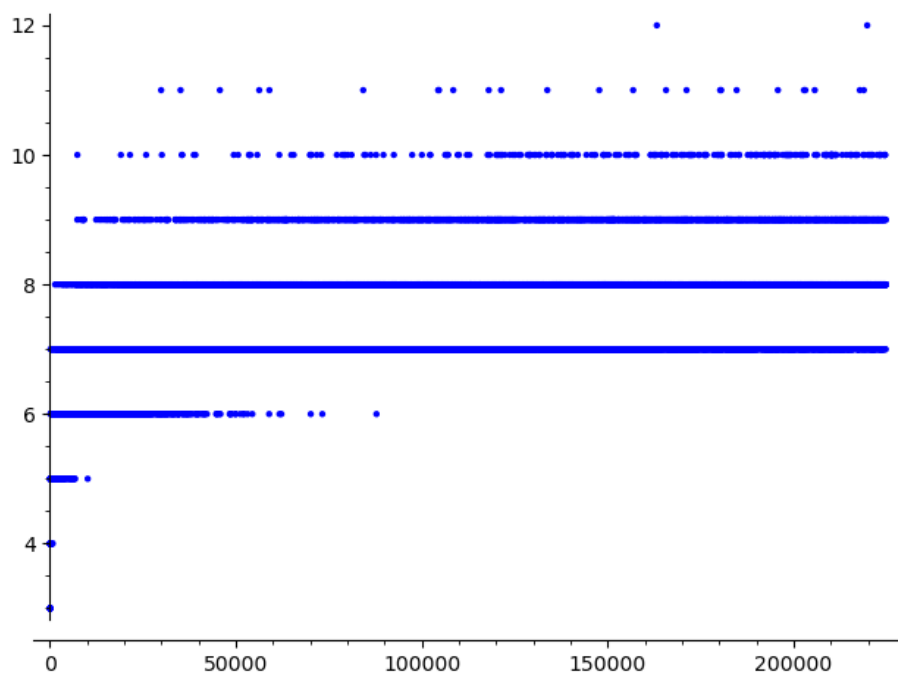


Figure 4.4: Values of $\nu(p)$ for primes $11 \leq p \leq 224,771$.

CHAPTER 5

CONCLUSION

Recall that for a prime $p \geq 7$, the set of DH permutations is defined as

$$\{a \circ \tau_p \circ b \mid a \in L, b \in R\},$$

where L and R are subsets of R_p , and τ_p is the key DH permutation.

We proved that for every prime $p \geq 7$,

$$R_p = \langle \mu_{g_p}, \rho \rangle = S_{p-1},$$

where μ_{g_p} and ρ are defined in Chapter 3. This result implies that the unlocking permutation h we seek can be expressed as a composition of μ_{g_p} and ρ . Moreover, both ρ and μ_{g_p} are relatively easy to compute.

We also conducted several computational experiments, one of which revealed a clear pattern. In particular, we plotted the normalized pairs

$$\left(\frac{j}{p-1}, \frac{k}{p-1} \right)$$

for which

$$\varepsilon(\rho^j \circ \mu_2 \circ \rho^k \circ \tau_p) = M_p.$$

The resulting points appear to lie along diagonal lines. Although M_p is the maximum value attained, it remains small relative to $p - 1$.

We also proposed an authentication protocol based on the following family of DH permutations for a given prime $p \geq 7$:

$$\{\rho^c \circ \tau_p \mid 1 \leq c \leq p - 1\}.$$

We proved that for all primes $p \geq 7$,

$$\#\nu(p) \leq (p - 1)^{1/2} + (p - 1)^{1/4} + \frac{1}{2}.$$

As a result,

$$\frac{\nu(p)}{p - 1} \longrightarrow 0$$

as $p \rightarrow \infty$ through prime values. Consequently, for every fixed c , the probability that an adversary succeeds by choosing an exponent uniformly at random is at most

$$\frac{\nu(p)}{p - 1},$$

which tends to 0 as $p \rightarrow \infty$ through prime values. Our numerical experiments also suggest that a sharper upper bound on $\nu(p)$ may hold.

We further investigated whether standard methods for solving the DLP, including reduction techniques, the index calculus method, and Shor's algorithm, could be applied to the fixed-point equation arising in our authentication protocol. Although these methods are effective for the ordinary DLP, our investigation did not reveal a direct way to apply them in this setting.

Overall, our computational and theoretical results provide a stepping stone for further study of the straightening of DH permutations. The observed diagonal patterns and the gap between our theoretical upper bound and the numerical results suggest several directions for future research. In particular, it would be interesting to prove the existence of the diagonal pattern, obtain a sharper upper bound on $\nu(p)$, and analyze the security of the authentication protocol against attacks beyond uniform random guessing. These questions demonstrate the potential of applications of DH permutations and deserve further investigation.

REFERENCES CITED

- [1] S. V. Breslavets, Fixed Points of Diffie-Hellman Permutations, Research Paper, Yulia's Dream Program, August 2025, <https://math.mit.edu/research/highschool/primes/materials/2025/YD/Breslavets.pdf>
- [2] W. Diffie and M. Hellman, New Directions in Cryptography, *IEEE Trans. Info. Th.* **22** (1976) 644-654.
- [3] P. Landrock, Power map permutations and the discrete log problem, *Des. Codes Cryptogr.* **77**, 2-3 (2015) 713–724.
- [4] J.S. Milne, Group Theory, <https://www.jmilne.org/math/CourseNotes/gt.html>
- [5] K.H. Rosen, Elementary Number Theory and Its Applications, Addison-Wesley Publishing Company, 1984.
- [6] P.W. Shor, Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer, <https://arxiv.org/abs/quant-ph/9508027>
- [7] The On-Line Encyclopedia of Integer Sequences, <https://oeis.org/>
- [8] C. Pomerance, Elementary thoughts on discrete logarithms, *Algorithmic Number Theory*, *MSRI Publications* **44** (2008)
- [9] W. Trappe and L. Washington, Introduction to Cryptography with Coding Theory, Second Edition, Pearson, 2006.
- [10] Wolfram MathWorld, Cayley Graph, <https://mathworld.wolfram.com/CayleyGraph.html>
- [11] J. Cilleruelo, Sidon Sets in $\mathbb{N}^d$, <https://www.sciencedirect.com/science/article/pii/S0097316509001848>