University of Tartu
Faculty of Science and Technology
Institute of Mathematics and Statistics

Calvin Pärn

Applications of semigroup actions to the cryptanalysis of Diffie-Hellman-like schemes

Mathematics and Statistics Curriculum
Mathematics

Master's thesis (30 ECTS)

Supervisor: MSc Valeh Farzaliyev

Tartu 2025

APPLICATIONS OF SEMIGROUP ACTIONS TO THE CRYPTANALYSIS OF DIFFIE-HELLMAN-LIKE SCHEMES

Master's thesis

Calvin Pärn

Abstract

In this master's thesis we examine semigroup actions and how they can be applied in the cryptanalysis of different cryptographic schemes by generalizing the schemes and viewing the underlying computational problems as problems regarding semigroup actions. We present an attack scenario against a generalized Diffie-Hellman protocol, which can be used if the underlying semigroup is close to a matrix algebra. We show how to use the approach to break one scheme proposed by the author and also a special case of an ElGamal-like scheme, which is based on Chebyshev polynomials.

CERCS research specialisation: P120 Number theory, field theory, algebraic geometry, algebra, group theory

Key Words: Semigroup actions, Diffie-Hellman key exchange, Chebyshev polynomials, representation theory, post-quantum cryptography

POOLRÜHMA TOIMETE RAKENDUSED DIFFIE-HELLMANI-LAADSETE SKEEMIDE KRÜPTOANALÜÜSIS

Magistritöö

Calvin Pärn

Lühikokkuvõte

Selles magistritöös vaatleme poolrühma toimeid ning nende kasutusvõimalusi erinevate krüptosüsteemide ründamiseks. Selleks peame olemasolevaid skeeme üldistama ning vaatlema arvutuslikke ülesandeid, millel nad põhinevad, poolrühma toimete vaatepunktist. Me esitame rünnakustrateegia, mille abil saab murda Diffie-Hellmani võtmevahetusprotokolli üldistatud versiooni juhul, kui poolrühm, millel protokolli allolev toime põhineb, on struktuurilt sarnane maatriksalgebraga. Me näitame, kuidas seda rünnakut kasutades saab murda ühe autori väljapakutud skeemi ning teatud eeldustel ühe ElGamali-laadse skeemi, mis põhineb Tšebõsevi polünoomidel.

CERCS teaduseriala: P120 Arvuteooria, väljateooria, algebraline geomeetria, algebra, rühmateooria

Märksõnad: Poolrühma toimed, Diffie-Hellmani võtmevahetusskeem, Tšebõsevi polünoomid, esitusteooria, post-kvantkrüptograafia

Contents

Introduction **3**

1 Semigroup actions, representations and cryptography **4**

1.1 Semigroup actions 4

1.2 Hard problems based on semigroup actions 5

1.3 Representation theory 9

2 Attack against the semigroup action-based Diffie-Hellman problem using linear algebra **12**

2.1 Attack against the extended Diffie-Hellman problem 12

3 Application of the semigroup action attack against a Chebyshev polynomial-based Diffie-Hellman problem **18**

3.1 Chebyshev polynomials 18

3.2 Periods of Chebyshev polynomials 21

3.3 Attack against the Chebyshev polynomial based Diffie-Hellman problem 23

Conclusions **27**

References **28**

A SageMath implementation of the Chebyshev polynomial Diffie-Hellman attack **30**

Introduction

In this master's thesis, we are looking at different cryptographic schemes from the viewpoint of semigroup actions. Many well-known cryptosystems can be generalized by redefining them as instantiations of different problems related to semigroup actions and thus attempts can be made to break these schemes using approaches amenable to certain attacks related to semigroup actions in general.

A semigroup action is a mapping, which requires a set, a semigroup and a well-defined way of using semigroup elements to move between the elements of the set. As it happens, if the semigroup is close to being a matrix algebra, an attack in [1] can be used to efficiently break a semigroup action-based Diffie-Hellman protocol.

We use this attack to break an ElGamal-like scheme based on Chebyshev polynomials and we also show how and why the attack can or cannot be used in different scenarios where semigroup actions are involved. Although the above-mentioned scheme does not claim to be post-quantum, then many of the examples viewed in the thesis are motivated by their potential use in post-quantum cryptography, which deals with cryptographic schemes that cannot be broken by either a classical computer or a quantum computer. We note, that the main computational problem regarding semigroup actions is considered post-quantum at the time of writing.

The work consists of three chapters. In the first chapter we give the preliminary information about semigroup actions and representations. We give basic definitions regarding semigroup actions and introduce different hard computational problems that can be formulated using semigroup actions. These problems also include generalizations of more traditional problems. We also give basic definitions from representation theory, which will allow us to view semigroup actions as certain types of graphs, which can be endowed with a linear structure.

In the second chapter we present the attack that allows us to break a Diffie-Hellman-like protocol based on semigroup actions if the semigroup action can be efficiently translated into the language of matrices and vectors. We give two examples of cases, where the attack cannot be applied and one example, where it can be applied.

In the third chapter we show that the attack can be used against a special case of a scheme based on Chebyshev polynomials proposed in [2]. We introduce some theory that justifies the use of the attack strategy and we present the attack together with an implementation in SageMath.

1 Semigroup actions, representations and cryptography

In this chapter we present the preliminary definitions and results regarding semigroup actions and representation theory. We also present different hard problems related to semigroup actions and group actions, some results regarding their cryptanalysis and a framework for analysing the Diffie-Hellman key exchange protocol from the viewpoint of semigroup actions.

1.1 Semigroup actions

We begin by giving basic definitions regarding semigroup actions.

Definition 1. A **semigroup** $(S, \cdot)$ is a set S with an associative binary operation $\cdot$. If the semigroup accepts an unit element, it is called a **monoid** and if every element in a monoid has an inverse element with respect to the binary operation, it is called a **group**. If all elements of a semigroup commute, we call it a **commutative** semigroup and if all elements of a group commute, we call it an **abelian** group.

There are many constructions in mathematics and computer science, where a set itself might not contain any explicit structure, but where a well-defined way to move between different elements of the set gives the set its meaning (e.g. automata). If these transitions can be combined with each other in a meaningful way and every type of transition is possible from any element of the set, the notion of an action appears.

Definition 2 (Semigroup action [1])**.** Consider a semigroup $(S, \cdot)$ and an arbitrary set X. A (left) **semigroup action** $(S, X, *)$ of S on X is a map

$$*\colon S \times X \to X,$$

which satisfies $(g \cdot h) * x = g * (h * x)$ for all $g, h \in S$ and $x \in X$. Similarly, we can define right actions, but we will consider all actions to be left actions unless said otherwise. We say that $*$ is an S-action on the set X and that S acts on X using the action $*$.

Definition 3 (Group action [3])**.** Given a group $(G, \cdot)$ with identity e and an arbitrary set X, a (left) **group action** $(G, X, *)$ of G on X is a map

$$*\colon G \times X \to X,$$

which satisfies the following identities:

- $(g \cdot h) * x = g * (h * x)$ for all $g, h \in G$ and $x \in X$.
- $e * x = x$ for all $x \in X$.

Right group actions are defined similarly, but we will consider all actions to be left actions unless said otherwise. We say that $*$ is a G-action on the set X and that G acts on X using the action $*$.

We note that a semigroup action can be visualized as a directed multigraph, where every element of X is a vertex and semigroup elements are edges between elements of X. Also, in the case of group actions, every element of X has a loop. This notion will be formalized later.

From every element $x \in X$, we can look at all the elements which can be obtained by acting on that element with every element of the semigroup. That set of elements originating from x, is called the **orbit** of x. More formally, given a semigroup (or group) action $(S, X, *)$, the orbit of x is the set

$$\mathcal{O}_x = \{y \in X \mid \exists s \in S \quad s * x = y\}.$$

The orbit of an element can be viewed as an indicator of the injectivity of the action on a fixed element: if the orbit of an element is large, it maps to many different elements. For cryptographic purposes, we usually require actions, which produce large orbits.

1.2 Hard problems based on semigroup actions

Now we present some hard problems based on semigroup and group actions and the relevant results in their cryptanalysis.

Problem 1 (Semigroup (group) action problem [1])**.** Given a semigroup (group) S acting on a set X and elements $x \in X$ and $y \in \mathcal{O}_x$, find $s \in S$ such that $s * x = y$.

Example 1. If we take the set of all n-dimensional vectors over a field $\mathbb{F}$ as the set X (it can also be viewed as a vector space, but it is not strictly necessary here) and the set of all $n \times n$ matrices over $\mathbb{F}$ to be the semigroup $(S, \cdot)$, then the following action can be defined:

$$* : S \times X \to X, \quad (A, x) \mapsto Ax.$$

We have that $(AB) * x = (AB)x = A(Bx) = A * (B * x)$, therefore the action is well-defined. Solving the semigroup action problem is not particularly hard

here, as for the vectors x, $y = A * x = Ax$, one only needs to find a vector a_i for every coefficient y_i of y such that $\langle a_i, x \rangle = y_i$. This can be done by setting the first coefficient of a_i to be $x_1^{-1} y_i$, where x_1 is the first coefficient of x and setting all the rest of the elements of a_i to be zero.

We note that it is not likely that one can use the semigroup action problem in its general form to construct efficient cryptographic primitives. Usually a specific semigroup action is chosen and then the problem changes into a version specific to the chosen action. However, it is a good starting point to build attacks against schemes that implicitly or explicitly make use of semigroup actions as a central building block.

While the general semigroup action problem can be considered unbroken, then in the presence of many small substructures within the semigroup, Pohlig-Hellman type reductions can be used to reduce the problem into smaller instances [4]. In the case of group actions, a Pollard-rho type birthday attack can be used [5] and the collision is found by going through a sequence of approximately $\sqrt{|\mathcal{O}_x|}$ elements. The worst-case hardness for the Pohlig-Hellman type attack is also $O(\sqrt{|\mathcal{O}_x|})$, but it improves in the presence of many small substructures in the group or semigroup. We note that for some cases the reductions themselves are not efficiently computable, which means that the choice of semigroup is critical in using Pohlig-Hellman type attacks – the Pollard-rho method is more general in it's construction. All in all, the orbits of elements with regards to the semigroup/group action need to be large to prevent generic attacks against the problems.

We are also interested in quantum attacks against the problem, as the semigroup action problem in general is considered to be post-quantum, meaning that no efficient methods for solving it using a classical or quantum computer have been found yet. In [6], from the view-point of elliptic curve isogenies, the authors tie the group action problem to the hidden shift problem, which can be solved by a quantum computer in $2^{O(\sqrt{\log |G|})}$ time. As far as we know, this attack cannot be used for semigroup actions. In [7], the authors give polynomial-time reductions of some cases of the semigroup action problem on modules to the hidden subgroup problem of an abelian group, which can be efficiently solved by a quantum computer, but the broken cases are quite specific and in general, classical attacks remain the most efficient against the semigroup action problem.

We can use the above-mentioned problems to construct a Diffie-Hellman-like protocol for key exchange with the hope that if we choose a group or semigroup for which the general attacks are ineffective, then we don't create

any new possible attacks by losing generality. The classical construction of the Diffie-Hellman key exchange uses discrete logarithms:

Problem 2 (Discrete logarithm problem in a group [8])**.** Given a finite group G and an element $g \in G$, when choosing a random integer a in the range $[0, \mathrm{ord}(g) - 1]$, the **discrete logarithm problem** requires an adversary to guess a, given g and g^a.

The discrete logarithm problem is well studied and it is known to be breakable by a quantum computer using a variant of Shor's algorithm [9]. In the non-quantum scenario the hardness of the discrete logarithm is dependent on the chosen group's structure. For a general group we have a lower bound of $\Omega(\sqrt{\mathrm{ord}(g)})$ found by Shoup [10] in 1997 — other approaches depend on the choice of the underlying group. It is also possible to formulate the discrete logarithm problem in periodic semigroups.

Problem 3 (Semigroup discrete logarithm problem [11])**.** We call a semigroup S **periodic** if each element of S has a finite order. The **discrete logarithm problem** in a periodic semigroup S is the problem of finding, given an element $g \in S$ and a positive power g^a of g, the integer a.

In [11], the authors show that the semigroup discrete logarithm problem can be reduced to the group discrete logarithm in a subgroup of the semigroup S in polynomial time. Thus the semigroup version of the problem is not harder than the classical discrete logarithm problem in groups.

Example 2. The classical number-theoretical discrete logarithm problem uses the cyclic group $\mathbb{F}_p^*$, where p is a prime number, fixes an element g of the group and chooses a random integer a in the range $[0, p - 1]$. The adversary must find a knowing p, g and g^a.

Several cryptographic schemes can be built upon the discrete logarithm problem, but the most influential of them is the Diffie-Hellman key exchange protocol, first presented in the seminal paper by Diffie and Hellman [12] in 1976. We present the protocol in its modern formulation. Alice and Bob are the parties that wish to share a common secret k by the end of the protocol.

Definition 4 (Diffie-Hellman key exchange [5])**.**

1. Alice and Bob agree on a finite cyclic group G and an element $g \in G$.
2. Alice privately chooses an integer a and computes $\alpha = g^a$. She sends α to Bob.

3. Bob privately chooses an integer b and computes $\beta = g^b$. He sends β to Alice.

4. Alice and Bob both compute

$$k = g^{ab} = \beta^a = \alpha^b.$$

The associated computational Diffie-Hellman problem is stated as follows:

Problem 4 (Diffie-Hellman problem in a group [5])**.** Let G be a finite cyclic group with generator $g \in G$ and elements $\alpha = g^a, \beta = g^b \in G$. The **Diffie-Hellman problem** asks for $\gamma \in G$ such that $\gamma = g^{(\log_g \alpha)(\log_g \beta)} = g^{ab}$.

One can see that solving the discrete logarithm problem allows to solve the Diffie-Hellman problem as well. Therefore G must be chosen such that the discrete logarithm problem is hard in that group. If an adversary can solve the Diffie-Hellman problem, he can also find out the shared secret in the Diffie-Hellman key exchange by eavesdropping on the communication between Alice and Bob.

As a quantum computer can break the Diffie-Hellman protocol efficiently by breaking the discrete logarithm problem, then we wish to extend the definition in order to study the post-quantum capabilities of the protocol. We can conveniently use semigroup actions for this.

Definition 5 (Extended Diffie-Hellman key exchange [1])**.** Let X be a finite set, S be a *commutative* semigroup and let $*$ be an S-action on the set X. The extended Diffie-Hellman key exchange in $(S, X, *)$ goes as follows:

1. Alice and Bob publicly agree on an element $x \in X$.

2. Alice chooses $s \in S$ and computes $s * x$. Alice sends $s * x$ to Bob.

3. Bob chooses $t \in S$ and computes $t * x$. Bob sends $t * x$ to Alice.

4. Alice and Bob both compute

$$s * (t * x) = (st) * x = (ts) * x = t * (s * x)$$

Notice that it is crucial here that S is commutative. In the usual Diffie-Hellman setting, we choose a cyclic group, which is already an abelian group.

Example 3. The traditional Diffie-Hellman key exchange is an instance of the extended protocol. Notice that if we choose $S = (\mathbb{Z}, \cdot)$ as the semigroup and X to be a cyclic group where the discrete logarithm is hard, we get that the following is an S-action on X:

$$* : S \times X \to X, \quad (s, x) \mapsto x^s.$$

Clearly, we have that $a * (b * x) = a * (x^b) = (x^b)^a = x^{ab} = (ab) * x$ and it is therefore a correctly defined action. If we instantiate the extended Diffie-Hellman protocol using this action, we get the original Diffie-Hellman protocol.

As seen in the previous example, the semigroup action problem replaced the discrete logarithm problem as one of the underlying assumptions in the protocol. Let us lastly define the Diffie-Hellman problem in a semigroup action setting.

Problem 5 (Extended Diffie-Hellman problem [1])**.** Let S be a commutative semigroup acting on a finite set X and let $a = s * x$, $b = t * x$ be two elements of the set X, which have been found by acting with $s, t \in S$ on a common element $x \in X$. The **extended Diffie-Hellman problem** asks for $y \in X$ such that $y = (st) * x = s * (t * x)$.

1.3 Representation theory

As we intend to use an attack, which relies on representations of semigroup actions, we will also give some basic definitions from representation theory.

Definition 6 (Associative algebra [13])**.** An **associative algebra** over a field $\mathbb{F}$ is a vector space A over $\mathbb{F}$ equipped with an associative, bilinear multiplication $A \times A \to A$, $(a, b) \mapsto ab$. A **unit** in an associative algebra A is an element $1 \in A$ such that $1a = a1 = a$.

Example 4. Here are some examples of associative algebras over $\mathbb{F}$:

1. The field $\mathbb{F}$.
2. The algebra $\mathbb{F}[x_1, \ldots, x_n]$ of polynomials of n variables over the field $\mathbb{F}$.
3. The algebra $\operatorname{End} V$ of endomorphisms of a vector space V over the field $\mathbb{F}$. The bilinear multiplication is the composition of endomorphisms as maps.

Definition 7 (Commutative algebra [13])**.** An algebra over a field $\mathbb{F}$ is **commutative** if $ab = ba$ for all $a, b \in A$.

Definition 8 (Homomorphism of algebras [13])**.** A **homomorphism of algebras** $f\colon A \to B$ is a linear map such that $f(xy) = f(x)f(y)$ for all $x, y \in A$ and $f(1) = 1$.

Definition 9 (Representation [13])**.** A **representation** of an algebra A is a vector space V together with a homomorphism of algebras $\rho\colon A \to \operatorname{End} V$.

As any finite-dimensional vector space over $\mathbb{F}$ can have its endomorphism ring represented by square matrices over $\mathbb{F}$ of the same dimension, we find that the goal of this definition (at least in the scope of this thesis) is to represent some less structured algebras as matrices over a field, which are usually more comfortable to work with. In our case, we will be interested in such semigroups to which the structure of a vector space can be easily added (e.g. polynomials and matrices).

It is sometimes useful to visualize semigroup actions as directed graphs with vertices as the set elements and edges as the semigroup elements. Then, of course, from every vertex there originates as many edges as there are elements in the semigroup and vertices can have many different edges between them. This notion is formulated as a quiver.

Definition 10 (Quiver [13])**.** A **quiver** Q is a directed graph, possibly with self-loops and/or multiple edges between two vertices. We denote the set of vertices of the quiver as I and the set of edges as E, which are represented as pairs of vertices (a, b), where a is the source and b is the target of the edge.

Definition 11 (Representation of a quiver [13])**.** A **representation** of a quiver Q is an assignment to each vertex $i \in I$ of a vector space V_i and to each edge $h = (a, b) \in E$ of a linear map $x_h\colon V_a \to V_b$.

In our case, we assign a single vector to each vertex as a one-dimensional subspace of $\mathbb{F}^n$ and the linear maps are $n \times n$ square matrices over $\mathbb{F}$.

Lastly, we can define an algebra for every quiver called a path algebra. This notion allows us to view every quiver as an algebra and therefore use the tools of representation theory on every structure that can be visualized as a quiver.

Definition 12 (Path algebra [13])**.** The **path algebra** P_Q of a quiver Q is the algebra whose basis is formed by oriented paths in Q, including the trivial paths p_i $i \in I$, corresponding to the vertices of Q. Multiplication is the concatenation of paths: ab is the path obtained by first tracing b and then a. If two paths cannot be concatenated, the product is defined to be zero.

This means that we can always find representations for semigroup actions by visualizing them as graphs, finding the associated path algebras and then view the representations of the path algebras. While it is impractical in most cases, it allows us to visualize them as quivers and be confident that we are not doing anything wrong from the viewpoint of representation theory.

2 Attack against the semigroup action-based Diffie-Hellman problem using linear algebra

In this chapter we present an attack against the extended Diffie-Hellman protocol using semigroup actions. We shall elaborate on its design rationale and give different examples to show when the attack can or cannot be applied.

2.1 Attack against the extended Diffie-Hellman problem

The attack is due to Maze, Monico and Rosenthal [1] from 2007. The setup is simple. First we must have a field $\mathbb{F}$ and a semigroup action $S \times X \to X$, where S is a commutative semigroup and X is a finite set. If we can find a semigroup homomorphism $\rho\colon S \to \mathbb{F}^{n\times n}$ (with multiplication as the operation) and an embedding (injective structure-preserving mapping) $\iota\colon X \to \mathbb{F}^n$ such that for all $s \in S$, $x \in X$ one has

$$\iota(s * x) = \rho(s)\iota(x),$$

then we can apply the attack.

As one can see, $\mathbb{F}^{n\times n}$ together with ρ is a representation of the semigroup S and if we visualize the semigroup action as a quiver, we see that the embedding ι together with the homomorphism ρ allows us to view the semigroup action as a representation of a path algebra, where the images of set elements are loops and the images of semigroup elements are the non-trivial paths in the quiver.

As ι is an injective mapping, we can use the representation to analyze the semigroup action using linear algebra and after finding a solution to the extended Diffie-Hellman problem in the representation, we can find the actual solution by applying the inverse of ι. The attack goes as follows:

Theorem 1 (Reduction of the Diffie-Hellman semigroup problem [1])**.** *Let S, X, ρ, ι be as above. We denote $\rho(S)$ as the image of S under ρ, which is also a commutative subsemigroup of $\mathbb{F}^{n\times n}$. Let $\mathbb{F}[S]$ be the commutative subalgebra of $\mathbb{F}^{n\times n}$ generated by the elements of $\rho(S)$ and let $d = \dim_{\mathbb{F}} \mathbb{F}[S]$. Then there exists a polynomial time reduction, which reduces the extended Diffie-Hellman problem to a linear algebra problem over $\mathbb{F}$ which can be solved in $O(d^2n+n^3)$ field operations.*

Proof. Let x, $y = g*x$ and $z = h*x$ be three elements in X and let $u = \iota(x)$, $v = \iota(y)$ and $w = \iota(z)$. Let $\mathcal{B} = \{M_1, \ldots, M_d\}$ be a basis of $\mathbb{F}[S]$. If $d \geq n$, we can extract a sub-family of cardinality n, $M_{i_1}, \ldots, M_{i_n}$ from $M_1, \ldots, M_d$ such that

$$\mathrm{Span}_{\mathbb{F}^n}\{M_{i_1}u, \ldots, M_{i_n}u\} = \mathrm{Span}_{\mathbb{F}^n}\{M_1u, \ldots, M_du\}.$$

This procedure can be done by using Gaussian elimination with complexity $O(d^2n)$. If $d < n$, we can just add enough zero matrices to get n matrices.

Let us now consider two equations with unknowns $a = (a_1, \ldots, a_n)^T \in \mathbb{F}^n$ and $b = (b_1, \ldots, b_n)^T \in \mathbb{F}^n$:

$$(a_1M_{i_1} + \ldots + a_nM_{i_n})u = v \quad \text{and} \quad (b_1M_{i_1} + \ldots + b_nM_{i_n})u = w.$$

As $\mathcal{B}$ is a basis, then both equations possess at least one solution, because thanks to the equality of spans, the left sides of both equalities can represent any element in $\mathbb{F}[S]u$ and $v, w \in \mathbb{F}[S]u$.

Now we note that the equations are equivalent to the following system:

$$[M_{i_1}u|\ldots|M_{i_n}u]a = v \quad \text{and} \quad [M_{i_1}u|\ldots|M_{i_n}u]b = w.$$

and a solution can be found in $O(n^3)$ steps.

We form the matrices

$$\begin{aligned} M_g &= (a_1M_{i_1} + \ldots + a_nM_{i_n}) \\ \text{and} \quad M_h &= (b_1M_{i_1} + \ldots + b_nM_{i_n}), \end{aligned}$$

which satisfy $M_gM_h = M_hM_g$, $M_gu = v$ and $M_hu = w$ and we let $\sigma = M_gM_hu$. Since $M_gu = \rho(g)u$ and $M_hu = \rho(h)u$, then

$$\sigma = M_gM_hu = M_g\rho(h)u = \rho(h)M_gu = \rho(h)\rho(g)u = \rho(gh)\iota(x) = \iota((gh)*x)$$

and

$$\iota^{-1}(\sigma) = (gh)*x,$$

which is a solution to the extended Diffie-Hellman problem instance. □

There are a few important assumptions that the attack requires. First of all, we require sampling to be efficient (and possible) in $\mathbb{F}[S]$. If d and n are very large, then sampling might become an issue and affect the complexity of the algorithm and also the running time of the algorithm may become too big. If the chosen semigroup S is already close to a matrix algebra, then this

shouldn't be an issue, but if ρ is not efficiently computable and/or the basis of $\mathbb{F}[S]$ is hard to find, then this presents a problem.

Secondly, we require that ι be efficiently invertible. This, however, is not an issue in most cases, as the embedding is mostly a quite simply defined mapping.

And finally, we emphasize that the original semigroup needs to be commutative. To get the equality $M_g M_h u = \rho(g)\rho(h)u$, we rely on the fact that if S is commutative, then $\mathbb{F}[S]$ is commutative and since $\rho(S)$ lies in $\mathbb{F}[S]$, we get $M_g \rho(h) = \rho(h) M_g$. In the language of quivers, this means that either way we compose the paths, we arrive at the same element of X, when starting our walk from the same initial element.

Next we present a few examples of how this attack can or cannot be used.

Example 5. In the case of the original Diffie-Hellman, the semigroup action was defined using the semigroup $(\mathbb{Z}, \cdot)$ and the underlying set was chosen to be the multiplicative group $\mathbb{F}_p^*$, which we can represent as powers of a generator g. The action was defined as

$$* \colon \mathbb{Z} \times \mathbb{F}_p^* \to \mathbb{F}_p^*, \quad (b, g^a) \mapsto g^{ab}.$$

Now, when finding ρ and ι for the attack, we need to consider that the equality $\iota(b * g^a) = \rho(b)\iota(g^a)$ must hold. Notice that on the left hand side we have an embedding of an exponentiation and on the right hand side there is a matrix-vector product. Exponentiation cannot be presented as a matrix-vector product in a natural way, therefore the embedding must reduce exponentiation to simple multiplication. Only one option seems feasible:

$$\rho(b) = b \quad \text{and} \quad \iota(g^a) = a.$$

Notice that the equality holds, as $\iota(b * g^a) = \iota(g^{ab}) = ab = ba = \rho(b)\iota(g^a)$. However, ι is not efficiently computable, as it requires solving the discrete logarithm problem in $\mathbb{F}_p^*$.

Example 6. In this example, we exemplify the need for commutativity of the original semigroup. We state the lattice isomorphism problem in its original setup and in its alternative representation:

Problem 6 (Lattice isomorphism problem [14])**.** Given two isomorphic lattices $\mathcal{L}, \mathcal{L}' \subset \mathbb{R}^n$ (meaning that one lattice can be transformed into the other by using rotations and reflections), find an orthonormal transformation O such that $\mathcal{L}' = O \cdot \mathcal{L}$.

Equivalently, given a quadratic form $Q = B^T B$, where $B \in \mathcal{GL}_n(\mathbb{R})$ and a quadratic form $Q' = U^T Q U$, where $U \in \mathcal{GL}_n(\mathbb{Z})$, find U.

Here we consider a lattice with a basis $B \in \mathcal{GL}_n(\mathbb{R})$ to consist of all linear combinations with integer coefficients of the basis vectors, which are the columns of B.

The lattice isomorphism problem is considered post-quantum and thus is actively studied. The problem can be translated efficiently into the language of group actions (we shall be using the quadratic form representation). Let $[Q]$ denote the class of lattices isomorphic to the lattice $\mathcal{L}(B)$, where $B \in \mathcal{GL}_n(\mathbb{R})$ and $Q = B^T B$. Denoting all $n \times n$ unimodular matrices as $\mathcal{GL}_n(\mathbb{Z})$, we can start defining the lattice isomorphism group action [15].

For this, we need to define $\mathcal{GL}_n^{\pm}(\mathbb{Z})$ as $\mathcal{GL}_n(\mathbb{Z})/\simeq_{\pm}$, which is the quotient of $\mathcal{GL}_n(\mathbb{Z})$ by the equivalence relation $\simeq_{\pm}$, where $U \simeq_{\pm} V \iff U = \pm V$. This means that we have equivalence classes $[U] = \{U, -U\}$ and we can define multiplication of the equivalence classes $[U]_{\pm}, [V]_{\pm} \in \mathcal{GL}_n^{\pm}(\mathbb{Z})$ as:

$$[U]_{\pm} \cdot [V]_{\pm} = [VU]_{\pm}.$$

Notice that $\mathcal{GL}_n^{\pm}(\mathbb{Z})$ is a group, where $[E]_{\pm}$ is the unit element (E is the $n \times n$ unit matrix in $\mathcal{GL}_n(\mathbb{Z})$). By choosing from each class $[U]_{\pm}$ a representative U, we can define the lattice isomorphism group action as:

$$* : \mathcal{GL}_n^{\pm}(\mathbb{Z}) \times [Q] \to [Q], \quad ([U]_{\pm}, Q') \mapsto U^T Q' U.$$

Notice that

$$\begin{aligned}([U]_{\pm} \cdot [V]_{\pm}) * Q' = ([VU]_{\pm}) * Q' = (VU)^T Q' VU = U^T V^T Q' VU \\ = [U]_{\pm} * ([V]_{\pm} * Q')).\end{aligned}$$

Therefore the group action is correctly defined.

However, note that $\mathcal{GL}_n^{\pm}(\mathbb{Z})$ is not commutative and thus the isomorphism problem itself cannot be tackled using the linearization attack. However, other related problems can be studied in its place, namely the weak unpredictability of the lattice isomorphism group action.

Definition 13 (t-weakly unpredictability of a group action [15])**.** A group action $*\colon G \times X \to X$ is said to be **t-weakly unpredictable** if given t pairs $(x_i, y_i) \in X \times X$, where $y_i = g * x_i$ for some fixed secret $g \in G$, when given a new element x_i', which is different from all other inputs, there is no probabilistic polynomial-time algorithm to produce an element $y_i' = g * x_i'$.

Our goal would be to show that the lattice isomorphism group action is not 1-weakly unpredictable. Our goal is to produce the element $Q'' = (U^2)^T Q U^2$ by

using the semigroup action attack on a hypothetical Diffie-Hellman instance, where both Alice and Bob send each other the element $Q' = U^T QU$. Both parties possess the same secret U and the final shared key is exactly Q''.

However, the attack doesn't solve this problem either. We either need to change the acting group from $\mathcal{GL}_n^{\pm}(\mathbb{Z})$ to the group $\mathbf{U}_{\pm} = \{[U^n]_{\pm} | n \in \mathbb{Z}\}$, which is a commutative group that contains both $[U]_{\pm}$ and $[U^2]_{\pm}$ or otherwise, we need to find the linear representations of U and U^2 from $\mathbb{F}[\mathcal{GL}_n^{\pm}(\mathbb{Z})]$, but both of these variants fail. In the first case, we need to know U to sample elements from $\mathbb{F}[\mathbf{U}_{\pm}]$, but U is secret. In the second case we find that as $\mathbb{F}[\mathcal{GL}_n^{\pm}(\mathbb{Z})]$ is non-commutative, then the final equalities in the algorithm description don't generally hold. Although we have a Diffie-Hellman instance where both of the parties' secrets are the same, then by using the notation in the algorithm description, we get

$$\begin{aligned}\sigma =& M_U M_U \iota(Q) = M_U \rho(U)\iota(Q) \neq \rho(U) M_U \iota(Q) = \\ & \rho(U)\rho(U)\iota(Q) = \rho(U^2)\iota(Q) = \iota([U^2]_{\pm} * Q) = \iota\left((U^2)^T Q U^2\right)\end{aligned}$$

Therefore, applying ι^{-1} to our potential σ does not generally produce Q'' and the weak unpredictability remains unbroken.

Example 7. We now present a protocol proposed by the author, which was efficiently broken by the linearization attack. First we choose a non-invertible $n \times n$ matrix S defined over a finite field $\mathbb{F}$ and take as the semigroup all non-negative powers of this matrix. As S is non-invertible, then the semigroup generated by the powers of S is not a group as it does not possess inverse elements to any elements except $S^0 = E$. As the underlying set, we choose all $m \times n$ matrices over $\mathbb{F}$, where m is much smaller than n. Let $\mathbf{S} = \langle S \rangle$ be the semigroup consisting of all the non-negative powers of S and let $\mathbf{A}$ be the set of all $m \times n$ matrices over $\mathbb{F}$. We define the following right semigroup action:

$$*\colon \mathbf{A} \times \mathbf{S} \to \mathbf{A}, \quad (A, S^x) \mapsto AS^x$$

For the key exchange, Alice and Bob agree on a public matrix A, send each other AS^x and AS^y accordingly and the shared secret is AS^{x+y}.

Both choosing a matrix A to mask S^x and choosing S to be non-invertible are necessary choices with regards to cryptanalysis. If S is invertible and no matrix A is chosen to mask S^x, then the discrete logarithm problem over invertible matrices can be reduced to a classical discrete logarithm instance [8], but we wish to obtain post-quantum security. If S is non-invertible and no matrix A is chosen to mask S^x, then there is another quantum algorithm by Childs and Ivanyos [16], which can be used to compute the semigroup discrete logarithm. However, in the same paper, the shifted semigroup discrete

logarithm problem is discussed for which no efficient quantum algorithms are yet known and our setup relies on a variant of that problem. Nevertheless, as we can now see, as the chosen semigroup is very close to a matrix algebra, then the semigroup action linearization attack applies.

If we choose $\rho\colon \mathbf{S} \to \mathbb{F}^{mn\times mn}$ to be

$$\rho(A) = A^T \otimes I_m,$$

and $\iota\colon \mathbf{A} \to \mathbb{F}^{mn}$ to be $\iota(A) = \mathrm{vec}(A)$, then we have

$$\iota(A * S) = \mathrm{vec}(I_m AS) = (S^T \otimes I_m)\mathrm{vec}(A) = \rho(S)\iota(A).$$

ι is efficiently computable and invertible and $\mathbf{S}$ is commutative. Thus the attack applies.

3 Application of the semigroup action attack against a Chebyshev polynomial-based Diffie-Hellman problem

In this chapter we use the linearization attack introduced in the last chapter against an extended Diffie-Hellman protocol, which is based on Chebyshev polynomials over matrices. We prove different properties of the Chebyshev polynomials and show how to exactly use the attack to break a special case of a scheme proposed in [2].

3.1 Chebyshev polynomials

First, we present the traditional definition of the Chebyshev polynomial.

Definition 14 (Chebyshev polynomial [2])**.** Let $n \in \mathbb{N}_0$, and $x \in [-1, 1]$. The **Chebyshev polynomial** (of the first kind) $T_n(x)\colon [-1, 1] \to [-1, 1]$ is defined using the following recurrence relation:

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x), \quad n \geq 2$$

and $T_0(x) = 1$ and $T_1(x) = x$.

The Chebyshev polynomial (in the case where $n \geq 2$) exhibits chaotic behaviour, which means that repeated application of the polynomial function on two initial values very close to one another produces two very different chains of valuations. In essence, it acts unpredictably. Relying on this property, we define two computational problems:

Problem 7 (Chebyshev-based discrete logarithm problem [2])**.** Given x and y, it is hard to find an integer n which satisfies $T_n(x) = y$.

Problem 8 (Chebyshev-based Diffie-Hellman problem [2])**.** Given x, $T_n(x)$ and $T_m(x)$, it is hard to find $T_{mn}(x)$.

In cryptography, it is commonplace to work over finite fields, which is why an another definition of the Chebyshev polynomial is required.

Definition 15 (General Chebyshev polynomial (GCP) [2])**.** Let $n \in \mathbb{N}_0$, p a large prime, $a \in \mathbb{F}_p^*$ and $x \in \mathbb{F}_p$. The **general Chebyshev polynomial** (GCP) of degree n, $T_n\colon \mathbb{F}_p \to \mathbb{F}_p$ is recursively defined as

$$T_n(x) = axT_{n-1}(x) - T_{n-2}(x) \mod p, \quad n \geq 2,$$

where $T_0(x) = 2a^{-1} \mod p$, where a^{-1} is the multiplicative inverse of a in $\mathbb{F}_p^*$ and $T_1(x) = x$.

While this definition together with the previously presented computational problems would already lend itself well for the uses of cryptography, then there is one issue. Let us fix $x \in \mathbb{F}_p$ and observe the sequence $(T_n(x))_{n=0}^{\infty}$. In the case of finite fields, we will necessarily encounter cycles in this sequence, namely of length which is a divisor of either $p-1$ or $p+1$ [2, 17]. Notice, that this might result in very short periods.

One way to solve the issue, is to choose a very large prime p and an initial element $x \in \mathbb{F}_p$, which provides us a period of exactly $p-1$ or $p+1$. However, period finding is not an easy task and generally a quantum computer would be needed to efficiently find the period of a candidate for x. If we choose p such that $p-1$ or $p+1$ has very few divisors, we could potentially find a good x fast (checking all the divisors for every candidate), but it happens that we can actually do better. For this, we need to change the dimensions of the indeterminate of the GCP.

Definition 16 (Multi-dimensional General Chebyshev polynomial [2])**.** Let $n \in \mathbb{N}_0$, p a large prime, $a \in \mathbb{F}_p^*$ and $X \in \mathbb{F}_p^{k\times k}$. Then the **multi-dimensional general Chebyshev polynomial** (MDGCP) of degree n, $T_n(X)\colon \mathbb{F}_p^{k\times k} \to \mathbb{F}_p^{k\times k}$ is recursively defined as

$$T_n(X) = aXT_{n-1}(X) - T_{n-2}(X) \mod p, \quad n \geq 2, \tag{1}$$

where $T_1(X) = X$ and $T_0(X) = 2a^{-1}E_k \mod p$, where a^{-1} is the multiplicative inverse of a in $\mathbb{F}_p^*$ and E_k is the $k \times k$ unit matrix.

In order to view Chebyshev polynomials in the context of semigroup actions, we need to prove that the consecutive application of Chebyshev polynomials on a fixed initial element forms a commutative semigroup with regards to the composition of polynomials.

We prove this property for general Chebyshev polynomials, but as the only difference between MDGCP and GCP is the choice of rings for the indeterminate, then after noting that the matrices aX^r and bX^s commute for $a, b \in \mathbb{F}_p$, we can just assert the same property onto multi-dimensional general Chebyshev polynomials.

Theorem 2. [2] *For general Chebyshev polynomials it holds for every $n, m \in \mathbb{N}_0$ that*

$$T_n(T_m(x)) = T_{mn}(x) = T_m(T_n(x)).$$

Proof. The equality (1) can be viewed as a difference equation and thus it has a characteristic equation

$$\lambda^2 - ax\lambda + 1 = 0,$$

which has roots $\lambda_{1,2} = \frac{ax \pm \sqrt{(ax)^2-4}}{2}$. As the two roots are different, we can write the general solution for equation (1) as

$$T_n(x) = c_1\lambda_1^n + c_2\lambda_2^n$$

If we plug in $T_0(x)$, we get $c_1 + c_2 = 2a^{-1}$. When plugging in $T_1(x)$, we get

$$\begin{aligned} x &= \frac{c_1 ax + c_1\sqrt{(ax)^2-4} + c_2 ax - c_2\sqrt{(ax)^2-4}}{2} \\ &= \frac{(c_1+c_2)ax + (c_1-c_2)\sqrt{(ax)^2-4}}{2}. \end{aligned}$$

Plugging in $c_1 + c_2 = 2a^{-1}$, we get

$$x = x + \frac{(c_1-c_2)\sqrt{(ax)^2-4}}{2} \implies \frac{(c_1-c_2)\sqrt{(ax)^2-4}}{2} = 0,$$

which implies $c_1 = c_2$, because ax is not a constant. This means that $c_1 = c_2 = a^{-1}$.

Now the general expression becomes

$$T_n(x) = (\lambda_1^n + \lambda_2^n)/a$$

and for some fixed $m, n \in \mathbb{N}_0$

$$\begin{aligned} T_m(x) &= (\lambda_1^m + \lambda_2^m)/a \quad \text{and} \\ T_n(T_m(x)) &= (k_1^n + k_2^n)/a \end{aligned}$$

for some k_1 and k_2. Taking $y = T_m(x)$, we get

$$k_1 + k_2 = \frac{ay + \sqrt{(ay)^2-4} + ay - \sqrt{(ay)^2-4}}{2} = ay = aT_m(x)$$

from which we can derive that $k_1 + k_2 = \lambda_1^m + \lambda_2^m$. We also get

$$k_1k_2 = \frac{(ay + \sqrt{(ay)^2-4})}{2}\frac{(ay - \sqrt{(ay)^2-4})}{2} = \frac{(ay)^2 - (ay)^2 + 4}{4} = 1$$

from which we derive that $k_1 = k_2^{-1}$ (similiarly we get that $\lambda_1 = \lambda_2^{-1}$). This means that $k_1 + k_1^{-1} = \lambda_1^m + \lambda_1^{-m}$ and now multiplying both sides by $k_1\lambda_1^m$, we get

$$\begin{aligned}
&k_1^2\lambda_1^m + \lambda_1^m - k_1\lambda_1^{2m} - k_1 = 0\\
&k_1\lambda_1^m(k_1 - \lambda_1^m) + (\lambda_1^m - k_1) = 0\\
&k_1\lambda_1^m(k_1 - \lambda_1^m) - (k_1 - \lambda_1^m) = 0\\
&(k_1\lambda_1^m - 1)(k_1 - \lambda_1^m) = 0.
\end{aligned}$$

As we are in a field, then either $k_1 = \lambda_1^m$ or $k_1 = \lambda_1^{-m} = \lambda_2^m$. No matter which, we still get

$$T_n(T_m(x)) = \frac{k_1^n + k_2^n}{2} = \frac{\lambda_1^{mn} + \lambda_2^{mn}}{2} = T_{mn}(x),$$

which is what we wanted to show. It is obvious that then also $T_m(T_n(x)) = T_{mn}(x)$ and therefore applying general Chebyshev polynomials on a fixed element forms a commutative semigroup with regards to composition of polynomials. □

As we have now proved that applying different Chebyshev polynomials to a common element forms a semigroup, we can define a semigroup action on a subset of all $k \times k$ matrices over $\mathbb{F}_p$.

Definition 17 (MDGCP semigroup action)**.** Let $n \in \mathbb{N}_0$, p a large prime, $a \in \mathbb{F}_p^*$ and $X \in \mathbb{F}_p^{k\times k}$. Let $\mathbf{T} = \{T_n(Y)|n \in \mathbb{N}_0\}$ denote all the multidimensional general Chebyshev polynomials over $\mathbb{F}_p^{k\times k}$ of indeterminate Y and let $\mathbf{T}(X) = \{T_n(X)|n \in \mathbb{N}_0\}$ be the set consisting of evaluations of all the Chebyshev polynomials on the input X. We define the **MDGCP semigroup action** $*$ as follows:

$$*\colon \mathbf{T} \times \mathbf{T}(X) \to \mathbf{T}(X); \quad (T_n(Y), Z) \mapsto T_n(Z),$$

where $T_n(Z)$ denotes the evaluation of $T_n(Y)$ on Z.

We note that thanks to the previously proven semigroup property of the MDGCP, the semigroup action is properly defined. Also, notice that $\mathbf{T}(X) = \mathcal{O}_X$.

3.2 Periods of Chebyshev polynomials

In order to create Diffie-Hellman-like schemes using Chebyshev polynomials, we require that the cycle that appears in the sequence $(T_n(X))_{n=0}^{\infty}$ has a long

period as it is an upper bound on $|\mathcal{O}_X|$. We noted before, that for general Chebyshev polynomials, the length of the cycle is a divisor of either $p-1$ or $p+1$. For MDCGPs, it happens that the situation is a bit better. The following two theorems describe the situation for MDGCPs:

Theorem 3 (Period of the MDGCP sequence [2])**.** *For a MDGCP $T_n(X)$, when $X^p = X \mod p$ and all of its eigenvalues $\lambda \neq \pm 2a^{-1} \mod p$, the MDGCP sequence will necessarily have a period of $p^2 - 1$.*

Theorem 4 (Minimal period of the MDGCP sequence [2])**.** *For a MDGCP $T_n(X)$, when $X^p = X \mod p$ and all of its eigenvalues $\lambda \neq \pm 2a^{-1} \mod p$, the minimal period of the MDGCP sequence must be a factor of $p^2 - 1$.*

As we can see, when changing to MDGCP polynomials, the situation is improved upon slightly when $X^p = X \mod p$, but we still have to deal with factors. However, in [2] the authors ran experiments and found that if $X^p \neq X \mod p$, the probability for the period to be less than $p-1$ becomes very small quite quickly when increasing the modulus p and the dimension of the initial matrix X.

In [2] it is shown that when the matrix X is diagonalizable (meaning that there exists a diagonal matrix $\Lambda \in \mathbb{F}_p^{k\times k}$ and an invertible matrix $Q \in \mathbb{F}_p^{k\times k}$ such that $X = Q^{-1}\Lambda Q$), we can state that $X^p = X \mod p$ and if it is not diagonalizable it must hold that $X^p \neq X \mod p$. This fact is used to generate the initial state matrix by instead constructing it from a Jordan decomposition. We note that the authors of [2] have only provided experimental results for the longer periods in the case, where $X^p \neq X \mod p$ and therefore it might still happen that short periods appear.

We, however, are interested in a more specific case, which is given in the following theorem:

Theorem 5 (MDGCP period of rank-one matrices [2])**.** *For a rank-one square matrix, when $X^p \neq X \mod p$, the minimum period iterating with X as the initial matrix of the MDGCP sequence is $4p$, where $p > 2$ is prime.*

While the increased period of the MDGCP sequence is welcome, we show in the next subsection that in the above case the Diffie-Hellman problem based on the MDGCP semigroup action becomes vulnerable to a version of the attack presented in the previous chapter. Before presenting the attack, we prove an essential property of rank-one matrices needed for the attack:

Proposition 1. [2] *For a rank-one square matrix, either $A^p = A \mod p$ or $A^n = 0$ for all $n \geq 2$.*

Proof. We first note that a square matrix A is of rank one if and only if it can be written as an outer product of two non-zero vectors. This is easily seen as

$$\begin{pmatrix} x_1 y^T \\ x_2 y^T \\ \dots \\ x_n y^T \end{pmatrix} = xy^T = (y_1 x, y_2 x, \dots, y_n x),$$

where the whole matrix consists of multiples of x or y and x and y are uniquely defined by these equalities.

Also, a rank-one matrix possesses maximally one non-zero eigenvalue. Due to the rank-nullity theorem, the dimension of the kernel of A is $n-1$. This means that the dimension of the eigenspace corresponding to the eigenvalue 0 is also $n-1$ and that the geometric multiplicity of 0 is $n-1$. As the algebraic multiplicity of an eigenvalue is always greater than or equal to the geometric multiplicity, then it is either $n-1$ or n. Thus the rank-one matrix has eigenvalues 0 and λ, where λ may also be zero. Now, $Ax = (xy^T)x = (y^T x)x$, as $y^T x$ is a scalar and thus $y^T x$ is also an eigenvalue of A. If it is non-zero, then we have found all the eigenvalues of A and $\lambda = y^T x$. If $y^T x = 0$, then let $\lambda \neq 0$ and let $w \neq 0$ be the corresponding eigenvector. We first find that $Aw = (xy^T)w = (y^T w)x = \lambda w$. This means that w is a scalar multiple of x and letting $w = \alpha x$, we get

$$Aw = A\alpha x = \alpha Ax = \alpha(y^T x)x = 0 = \lambda w.$$

Because w is an eigenvector, we get that $\lambda = 0$ and thus the only eigenvalues of A are 0 and $y^T x$.

We note that for all $n \geq 1$

$$A^n = xy^T xy^T \dots xy^T = x(y^T x)^{n-1} y^T = (y^T x)^{n-1} A.$$

Now if $y^T x \neq 0$, we get $A^p = (y^T x)^{p-1} A = A \mod p$ due to Fermat's little theorem. If $y^T x = 0$, we necessarily get $A^n = 0 \neq A \mod p$ for all $n \geq 2$. □

3.3 Attack against the Chebyshev polynomial based Diffie-Hellman problem

Now we will describe a public key encryption scheme, which we can break using the semigroup action attack, given that the initial matrix X chosen for MDGCP is a rank-one matrix with all eigenvalues being zero or equivalently

$X^p \neq X \mod p$. M-CEPKC [2] (Multi-dimensional General Chebyshev-ElGamal Public Key Cryptosystem) is in essence just an ElGamal instantiation using Chebyshev polynomials for the underlying group action. One can easily see that if you can solve the extended Diffie-Hellman problem instantiated with a MDGCP group action, then you can break M-CEPKC.

Key generation:

1. Choose a large prime p, integers $s < 4p$, $a \in \mathbb{F}_p^*$ and a matrix $X \in \mathbb{F}_p^{k\times k}$ satisfying $X^p \neq X \mod p$.
2. Calculate $A = T_s(X)$.
3. (X, a, A, p) is the public key, s is the private key.

Encryption:

1. Represent the message as a matrix $M \in \mathbb{F}_p^{k\times k}$.
2. Select a random integer $r < 4p$.
3. Compute $B = T_r(A)$, $C_1 = T_r(X)$ and $C_2 = (M + B) \mod p$.
4. The ciphertext is (C_1, C_2).

Decryption:

1. Compute $B_1 = T_s(C_1)$.
2. Find $M = (C_2 - B_1) \mod p$

As said before, we can break this scheme, by solving the extended Diffie-Hellman problem. Given $A = T_s(X), C_1 = T_r(X)$, if we can find $B = T_r(A) = T_r(T_s(X)) = T_{rs}(X)$, we can recover the message $M = C_2 - B \mod p$ with only the ciphertext and the public key.

As proven before, if X is a rank-one matrix, where $X^p \neq X \mod p$, then $X^2 = 0$. This also means that every Chebyshev polynomial evaluated on X reduces to (or already is) of the form $bX + cE_k \mod p$, where $b, c \in \mathbb{F}_p$. Namely, we find that while there are no cycles specifically, then the MDGCP sequence follows a very specific pattern.

Proposition 2. *Given $X^2 = 0$ for $X \in \mathbb{F}_p^{k\times k}$ and $a \in \mathbb{F}_p^*$, for every $l \in \mathbb{N}_0$, we have*

$$T_{4l}(X) = 2a^{-1}E_k, \quad T_{4l+1}(X) = (4l+1)X,$$
$$T_{4l+2}(X) = -2a^{-1}E_k \quad \textit{and} \quad T_{4l+3} = -(4l+3)X.$$

Proof. Choose an arbitrary $l \in \mathbb{N}_0$ and assume that $T_{4l} = 2a^{-1}E_k$ and $T_{4l+1} = (4l+1)X$. Then

$$\begin{aligned}
T_{4l+2}(X) &= aX(4l+1)X - 2a^{-1}E_k = -2a^{-1}E_k, \\
T_{4l+3}(X) &= aX(-2a^{-1}E_k) - (4l+1)X = -2X - (4l+1)X = -(4l+3)X \\
T_{4l+4}(X) &= T_{4(l+1)}(X) = -aX(4l+3)X + 2a^{-1}E_k = 2a^{-1}E_k \\
T_{4(l+1)+1}(X) &= aX(2a^{-1}E_k) + (4l+3)X = 2X + (4l+3)X \\
&= (4(l+1)+1)X.
\end{aligned}$$

□

This means that we have a specific rule for computing $T_n(X)$ for every $n \in \mathbb{N}_0$. Although in theory, we still have a cycle of $4p$, then the orbit of X with regards to the Chebyshev semigroup action becomes so structured, that for the purposes of cryptanalysis, we can arrange it into 4 classes, specifically residue classes modulo 4. If we are to find correctly defined $\rho\colon \mathbf{T} \to \mathbb{F}_p^{n\times n}$ and $\iota\colon \mathbf{T}(X) \to \mathbb{F}_p^n$, where $\mathbf{T}(X) = \mathcal{O}_X$, to use for the linearization attack, we must consider that although ρ and ι may not be strictly invertible, then the shared Diffie-Hellman secret will still be uniquely defined. Secondly, ρ and ι have to preserve the information regarding the degree of the polynomial, if $n \equiv 1, 3 \mod p$.

One can show by rudimentary, but rather messy computations, that trying to find a 2×2 representation of the semigroup action fails due to properties required from the homomorphism. But a 3×3 representation exists and we present it now:

$$\rho\colon \mathbf{T} \to \mathbb{F}_p^{3\times 3}; \quad \rho(T_n(Y)) = \begin{cases} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, & \text{if } n \equiv 0 \mod 4 \\ \begin{pmatrix} 1 & 0 & 0 \\ 0 & n & 0 \\ 0 & 0 & 1 \end{pmatrix}, & \text{if } n \equiv 1 \mod 4 \\ \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, & \text{if } n \equiv 2 \mod 4 \\ \begin{pmatrix} 1 & 0 & 0 \\ 0 & -n & 0 \\ 0 & 0 & 1 \end{pmatrix}, & \text{if } n \equiv 3 \mod 4 \end{cases}$$

$$\iota\colon \mathcal{O}_X \to \mathbb{F}_p^3; \quad \iota(T_n(X)) = \begin{cases} \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, & \text{if } n \equiv 0 \mod 4 \\ \begin{pmatrix} 0 \\ n \\ 1 \end{pmatrix}, & \text{if } n \equiv 1 \mod 4 \\ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, & \text{if } n \equiv 2 \mod 4 \\ \begin{pmatrix} 0 \\ -n \\ 1 \end{pmatrix}, & \text{if } n \equiv 3 \mod 4 \end{cases}$$

It is straightforward to check that ρ is a homomorphism and that $\iota(T_n * Y) = \rho(T_n)\iota(Y)$ for all $T_n \in \mathbf{T}$ and $Y \in \mathcal{O}_X$ by checking all possibilities. Obviously, ρ and ι are efficiently computable and ι is also efficiently invertible.

We also note, that the complexity estimation $O(d^2n+n^3)$ reduces to constant time, as n is now always equal to 3 and the dimension d of $\mathbb{F}[\mathbf{T}]$ is also 3, as the basis can be chosen as

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

A SageMath implementation of the attack can be found in appendix A.

Conclusions

In this master's thesis we explored how semigroup actions can be used to generalize and in some cases break some existing cryptographic schemes. We presented a general attack strategy and showed how it can or cannot be used in different scenarios.

The main result of the thesis was an attack against a special case of an ElGamal-like scheme based on Chebyshev polynomials. We showed how the scheme can be broken if the initial state matrix is rank-one and nilpotent. The semigroup action representation in that case was simple and allowed for a constant-time attack. It remains to be seen if the attack can somehow be expanded to work against more general setups.

References

[1] G. Maze, C. Monico, and J. Rosenthal. "Public Key Cryptography based on Semigroup Actions". In: *Advances in Mathematics of Communications* 1.4 (2007), pp. 489–507.

[2] R. Min et al. "A public key encryption algorithm based on multi-dimensional general Chebyshev polynomial". In: *Designs, Codes and Cryptography* (2025).

[3] G. D'Alconzo and A. J. Di Scala. "Representations of Group Actions and their Applications in Cryptography". In: 99 (2024), p. 102476.

[4] O. W. Gnilke. "The Semigroup Action Problem in Cryptography". PhD thesis. University College Dublin, 2014.

[5] C. J. Monico. "Semirings and Semigroup Actions in Public-Key Cryptography". PhD thesis. University of Notre Dame, 2002.

[6] A. Childs, D. Jao, and V. Soukharev. "Constructing elliptic curve isogenies in quantum subexponential time". In: *Journal of Mathematical Cryptology* 8.1 (2014), pp. 1–29.

[7] H. Huang, C. Peng, and L. Deng. "Reduction of the semigroup-action problem on a module to the hidden-subgroup problem". In: *Quantum Information Processing* 23.300 (2024).

[8] A. Menezes and Y. Wu. "The Discrete Logarithm Problem in GL(n, q)". In: *Ars Comb.* 47 (1997).

[9] P. W. Shor. "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer". In: *SIAM Journal on Computing* 26.5 (1997), pp. 1484–1509.

[10] V. Shoup. "Lower Bounds for Discrete Logarithms and Related Problems". In: *Advances in Cryptology - EUROCRYPT '97, International Conference on the Theory and Application of Cryptographic Techniques, Konstanz, Germany, May 11-15, 1997, Proceeding.* Vol. 1233. Springer, 1997, pp. 256–266.

[11] M. Banin and B. Tsaban. "A reduction of semigroup DLP to classic DLP". In: *Designs, Codes and Cryptography* 81 (2013).

[12] W. Diffie and M. Hellman. "New directions in cryptography". In: *IEEE Transactions on Information Theory* 22.6 (1976), pp. 644–654.

[13] P. Etingof et al. *Introduction to Representation Theory.* American Mathematical Society, 2011.

[14] L. Ducas and W. van Woerden. "On the Lattice Isomorphism Problem, Quadratic Forms, Remarkable Lattices, and Cryptography". In: *Advances in Cryptology – EUROCRYPT 2022*. Springer International Publishing, 2022, pp. 643–673.

[15] A. Budroni, J.-J. Chi-Domínguez, and E. Franch. *Don't Use It Twice: Reloaded! On the Lattice Isomorphism Group Action*. Cryptology ePrint Archive, Paper 2025/516. 2025. URL: https://eprint.iacr.org/2025/516.

[16] A. M. Childs and G. Ivanyos. "Quantum computation of discrete logarithms in semigroups". In: *Journal of Mathematical Cryptology* 8.4 (2014), pp. 405–416.

[17] G. J. Fee and M. B. Monagan. *Cryptography using Chebyshev polynomials*. URL: https://wayback.cecm.sfu.ca/CAG/papers/Cheb.pdf.

Appendix A SageMath implementation of the Chebyshev polynomial Diffie-Hellman attack

```
# First, we set the parameters.

from functools import lru_cache  # Used to make recursive functions
                                 # more efficient

p = 10009                        # Prime modulus
d = 3                            # Dimension of F[S]
a = Zp(17)                       # Define the parameter 'a'
                                 # in the definition of Chebyshev
a_inv = a.inverse()
Zp = GF(p)                       # Define the underlying field
M = MatrixSpace(Zp,3,3)          # Define the matrix algebra over Zp
Y = M([[0,1,1],[0,0,0],[0,0,0]]) # Initial state matrix
ZpX.<X> = M[]                    # Polynomials over the matrix
                                 # algebra defined over Zp

# Now we define the functions for finding Chebyshev polynomials
# and for calculating iota and it's inverse.

@lru_cache
def cheb(m):                    # Recursive definition of Chebyshev
    if m == 0:
        return ZpX(2*a_inv*M(identity_matrix(3)))
    if m == 1:
        return X
    else:
        return a*X*cheb(m-1)-cheb(m-2)

def iota(iotamat):
    if iotamat == M([[2*a_inv,0,0],[0,2*a_inv,0],[0,0,2*a_inv]]):
        return vector(Zp,[0,0,0])
    if iotamat == M([[-2*a_inv,0,0],[0,-2*a_inv,0],[0,0,-2*a_inv]]):
        return vector(Zp,[1,0,0])
    if mod(Integer(iotamat[0][1]),4) == 1:
        return vector(Zp,[0,iotamat[0][1],1])
```

```
    if mod(Integer(-iotamat[0][1]),4) == 3:
        return vector(Zp,[0,iotamat[0][1],1])
    else:
        print("Wrong input to iota!")

def inverse_iota(iotavec):
    if iotavec == vector(Zp,[0,0,0]):
        return M([[2*a_inv,0,0],[0,2*a_inv,0],[0,0,2*a_inv]])
    if iotavec == vector(Zp,[1,0,0]):
        return M([[-2*a_inv,0,0],[0,-2*a_inv,0],[0,0,-2*a_inv]])
    if mod(iotavec[1],4) == 1:
        return iotavec[1]*Y
    if mod(-iotavec[1],4) == 3:
        return iotavec[1]*Y
    else:
        print("Wrong input to inverse iota!")

# Now we set up the Chebyshev Diffie-Hellman instance.

m = 13                          # Alice's secret
n = 8                           # Bob's secret
alice = cheb(m)                 # Alice's polynomial
bob = cheb(n)                   # Bob's polynomial
aliceDH = alice(Y)              # Both polynomials evaluated
                                # on the public matrix
bobDH = bob(Y)
mncheb = cheb(m*n)              # Shared secret
solutionDH = mncheb(Y)
u = vector(Zp,[0,1,1])          # Embedding of the public matrix
v = iota(aliceDH)               # Embeddings of the public values
w = iota(bobDH)

# We define the basis of F[S].

basis = []
basis.append(M([[0,0,1],[0,0,0],[0,0,0]]))
basis.append(M([[1,0,0],[0,0,0],[0,0,1]]))
basis.append(M([[0,0,0],[0,1,0],[0,0,0]]))

# We solve the necessary systems of linear equations.
```

```
lineq = []
for i in range(d):
    lineq.append(basis[i]*u)
lineq = matrix(Zp,lineq)
lineq = lineq.transpose()
a_vec = lineq.solve_right(v)
b_vec = lineq.solve_right(w)

# We find sigma and apply the inverse of iota
# to obtain the shared secret.

Mg = M()
Mh = M()
for i in range(d):
    Mg += basis[i]*a_vec[i]
    Mh += basis[i]*b_vec[i]
sigma = Mg*Mh*u
solution_linear = inverse_iota(sigma)
print(solutionDH == solution_linear)

# If the output is "True", then the algorithm
# produced the correct answer.
```

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Calvin Pärn,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose "Applications of semigroup actions to the cryptanalysis of Diffie-Hellman-like schemes", mille juhendaja on Valeh Farzalijev, reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.

2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commonsi litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.

3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.

4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Calvin Pärn
29.05.2025