Algebraic Graph Theory for Cryptographic Protocols

Authors: Laasya Vajjala (11848603) & Surya Pramod Vadapalli (11861342)
Submission Date: 12/12/2024

A. Abstract

In this project, we explore the application of algebraic graph theory to develop cryptographic protocols that are resilient to quantum attacks. Specifically, we analyze the spectral and algebraic properties of Cayley graphs, expander graphs, and isogeny graphs to construct secure systems for digital signatures and key generation. By leveraging adjacency matrices and eigenvalues, we establish cryptographic primitives and evaluate their robustness.

Our work involves generating these graph types and performing spectral analysis, focusing on key metrics like spectral gap, spectral radius, and algebraic connectivity, which are crucial for cryptographic strength. We implement a graph-based digital signature scheme and assess its signature uniqueness and verification accuracy.

To test quantum resilience, we simulate Grover's algorithm and Shor's algorithm to identify potential vulnerabilities in graph-based schemes. While Grover's attack highlights partial reductions in security, Shor's simulation demonstrates robustness under specific conditions. Additionally, we optimize the scalability of these methods using parallel computation, enabling spectral analysis for large-scale graphs.

The results show that isogeny graphs offer advanced post-quantum potential, while Cayley and expander graphs provide strong cryptographic properties. This study highlights the viability of graph-based cryptography for secure communication in a quantum era and sets a foundation for further research into advanced graph structures and attack simulations.

B. Introduction

1. Background

Graph theory has emerged as a powerful mathematical framework in cryptography, offering unique methods for constructing secure systems. It provides tools for encoding complex structures using vertices and edges, with algebraic and spectral properties playing a significant role in ensuring cryptographic strength. In particular, Cayley graphs, expander graphs, and isogeny graphs have gained prominence for their mathematical richness and applicability to advanced cryptographic protocols.

- Cayley graphs are built from algebraic groups, offering structured yet versatile graph representations that facilitate efficient computation.
- Expander graphs are valued for their high connectivity and large spectral

gaps, which contribute to robustness against attacks.

- Isogeny graphs, based on elliptic curve isogenies, are integral to post-quantum cryptography due to their connection to hard mathematical problems resistant to quantum algorithms.

2. Motivation

As quantum computing progresses, traditional cryptographic protocols face increasing vulnerabilities, particularly to algorithms like Shor's and Grover's, which can efficiently break RSA and symmetric key systems, respectively. This has motivated a shift towards post-quantum cryptography, which aims to develop cryptographic methods secure against quantum attacks.

Algebraic and spectral properties of graphs are central to this effort. These properties not only provide insights into the structure and robustness of graphs but also enable the construction of cryptographic primitives such as keys and signatures. By understanding how these properties interact with quantum algorithms, we can design systems that are both efficient and secure.

3. Objectives

The objectives of this project are threefold:

➢ Spectral Analysis for Cryptographic Applications: Evaluate the spectral properties of Cayley, expander, and isogeny graphs to generate cryptographic keys and signatures. This includes analyzing metrics like spectral gap, spectral radius, and algebraic connectivity for their role in ensuring security and efficiency.

➢ Quantum Attack Simulations: Simulate Grover's and Shor's algorithms to test the resilience of graph-based cryptographic protocols. These simulations help identify vulnerabilities and evaluate the robustness of each graph type.

➢ Scalability and Optimization: Implement large-scale graph analysis using parallel computation frameworks, enabling practical application of these methods to real-world cryptographic systems. This ensures that the protocols are not only theoretically sound but also computationally feasible.

By addressing these objectives, this study contributes to the development of secure and efficient post-quantum cryptographic protocols rooted in algebraic graph theory.

C. Literature Review

1. History & Context

The application of graph theory in cryptography has a rich history, rooted in its ability to model complex relationships and structures. Over the years, significant advancements have been made in leveraging graph-based properties for secure communication systems. Below, we review key developments in this field:

➢ **Cayley and Expander Graphs in Cryptography:**

Cayley graphs, constructed from algebraic groups, have been extensively studied for their symmetry and efficiency. They have been used to model network topologies, construct error-correcting codes, and generate cryptographic keys due to their structured connectivity and predictable properties.

Expander graphs are known for their large **spectral gaps**, which provide high connectivity and robustness against attacks. These properties have made expander graphs valuable in constructing secure hash functions, pseudo-random generators, and fault-tolerant networks.

Both graph types have been explored for their ability to ensure cryptographic primitives like key generation and data integrity while resisting certain classical attack vectors.

- ➢ **Isogeny-Based Cryptography in Post-Quantum Contexts:**

Isogeny graphs, derived from the relationships between elliptic curves, have gained attention in post-quantum cryptography due to their resilience to quantum algorithms like Shor's algorithm. The hardness of finding isogenies between elliptic curves forms the foundation of protocols like the Supersingular Isogeny Diffie-Hellman (SIDH) key exchange.

These graphs are computationally challenging to analyze and reverse-engineer, making them promising candidates for quantum-safe cryptography.

Recent work has focused on improving the efficiency and security of isogeny-based protocols, addressing potential vulnerabilities in implementation.

- ➢ **Existing Cryptographic Protocols and Quantum Vulnerabilities:**

Traditional cryptographic protocols, such as RSA and ECC (Elliptic Curve Cryptography), rely on the difficulty of factoring large integers or solving discrete logarithms. However, these problems are solvable in polynomial time using Shor's algorithm, making them unsuitable for a quantum future.

Symmetric-key cryptography, while more resistant to quantum attacks, faces vulnerabilities under Grover's algorithm, which can reduce brute-force search times exponentially.

2. Comparison

Our work is situated within the growing body of research on post-quantum cryptographic methods, alongside other approaches like lattice-based and multivariate cryptography.

1. Graph-Based Cryptography vs. Lattice-Based Cryptography:

Lattice-based cryptography relies on the hardness of problems like learning with errors (LWE) and shortest vector problem (SVP), making it a prominent post-quantum approach. While lattice-based methods are

computationally efficient, they often require large key sizes.

Graph-based cryptography, particularly using isogeny graphs, offers smaller key sizes and more compact protocols, though it faces challenges in computational efficiency for larger-scale systems.

2. **Graph-Based Cryptography vs. Multivariate Cryptography:**

Multivariate cryptographic systems use polynomial equations over finite fields for encryption and signature schemes. While efficient, they have been prone to structural weaknesses and vulnerabilities to certain algebraic attacks.

Graph-based systems, in contrast, derive their security from spectral properties and combinatorial complexity, providing a distinct approach to resilience against both classical and quantum attacks.

D. Methodology

1. Graph Construction

Cayley Graphs

Cayley graphs are constructed from **algebraic groups** and their generators. They represent the group's structure in graphical form, where nodes correspond to elements of the group, and edges represent operations using specific generators.

Algorithm for Cayley Graph Construction:

Input: Group order n and generator set G.

The group is typically cyclic (Zn) for simplicity.

Generators are elements $g \in G$ such that every group element can be reached by repeatedly applying g.

Nodes: Create n nodes, each representing an element of the group $\{0,1,\ldots,n-1\}$.

Edges: For each node v and generator g, add an edge between v and (v+g) mod n. This ensures the graph is connected and represents the group's structure.

Output: A Cayley graph where each node connects to others based on the generator operations.

Example: For Z6 with generators {1,3}:

Nodes: {0,1,2,3,4,5}

Edges: Connect v to (v+1) mod 6(v + 1) and (v+3) mod 6.

Expander Graphs

Expander graphs are sparse yet highly connected graphs, characterized by large spectral gaps in their adjacency matrices. These graphs are essential for ensuring robustness and efficient data dissemination in cryptographic systems.

Algorithm for Expander Graph Construction:

Input: Number of nodes n and degree d (regularity of the graph).

Nodes: Create n nodes labeled $\{0,1,\ldots,n-1\}$.

Edges: Use a **random regular graph generation algorithm** to create a graph where each node has exactly d edges. This ensures the graph remains sparse yet connected.

Output: An expander graph that guarantees a large spectral gap.

Isogeny Graphs

Isogeny graphs represent elliptic curves over finite fields, with edges corresponding to **isogenies** (mappings) between curves. These graphs are computationally intensive to construct but offer unique advantages for quantum-resistant cryptography.

Algorithm for Isogeny Graph Construction:

Input: A prime p (field order) and degree ℓ\ellℓ of isogenies.

Nodes:

Generate elliptic curves E(a,b) over Fp that satisfy 4a^3 + 27b^2 not equal to 0 mod p (non-singular curves).

Edges:

For each pair of curves E1(a1,b1) and E2(a2,b2), create an edge if there exists an isogeny of degree ℓ mapping E1 to E2.

Output: An isogeny graph representing elliptic curves and their isogenies.

Example: For p=5 and ℓ=2:

Nodes: Represent all valid elliptic curves E(a,b) over F5.

Edges: Connect nodes based on degree-2 isogenies.

2. **Spectral Analysis**

Spectral analysis in the context of graph theory refers to the study of the eigenvalues of a graph's adjacency or Laplacian matrix. These eigenvalues provide critical insights into the graph's structure, connectivity, and robustness. In cryptographic applications, specific spectral metrics play a pivotal role in evaluating the security, efficiency, and resilience of graph-based cryptographic protocols.

Key Spectral Metrics

1. Spectral Gap

The **spectral gap** is defined as the difference between the largest and second-largest eigenvalues of a graph's adjacency matrix. For Laplacian matrices, it is the difference between the smallest and second-smallest eigenvalues.

- **Formula**:

Adjacency matrix: λgap=λmax−λsecond largest

Laplacian matrix: λgap=λ2, where λ2 is the second-smallest eigenvalue (also called the Fiedler value).

- **Cryptographic Relevance**:

Expander properties: A large spectral gap indicates that the graph is an excellent expander. Expander graphs are highly robust

against disruptions, making them ideal for fault-tolerant cryptographic systems.

Tamper resistance: Graphs with large spectral gaps are more resistant to structural tampering, as changes to the graph (e.g., edge removal) minimally impact connectivity and resilience.

2. Spectral Radius

The **spectral radius** is the largest absolute value of the eigenvalues of a graph's adjacency matrix.

Formula: $\rho = \max\{|\lambda 1|, |\lambda 2|, \ldots, |\lambda n|\}$ where λi are the eigenvalues of the adjacency matrix.

Cryptographic Relevance:

Key robustness: The spectral radius reflects the overall connectivity of the graph. Graphs with a higher spectral radius tend to have stronger connections between nodes, ensuring more reliable communication pathways.

Eigenvalue-based signatures: Spectral radius is a key component in cryptographic protocols that rely on the eigenvalues of a graph as unique identifiers for keys or signatures. Changes to the graph (e.g., during attacks) result in detectable changes to the spectral radius.

3. Algebraic Connectivity

The **algebraic connectivity** of a graph is the second-smallest eigenvalue ($\lambda 2$) of the Laplacian matrix. It measures how "well-connected" the graph is.

Formula: $\lambda 2(L)$, where $L = D - A$, with D being the degree matrix and A the adjacency matrix.

Cryptographic Relevance:

Graph connectivity: A higher algebraic connectivity indicates that the graph remains well-connected even after edge or node removal, ensuring resilience against tampering or attacks.

Fault tolerance: Cryptographic systems using graphs with high algebraic connectivity are more robust against disruptions, as no single point of failure can significantly disconnect the graph.

3. Cryptographic Scheme

The cryptographic scheme in this project revolves around leveraging the spectral and algebraic properties of graphs for key and signature generation. The process uses Cayley, expander, and isogeny graphs as the foundation, exploiting their unique structural and spectral characteristics to create a secure and quantum-resistant cryptographic protocol.

Key Generation Process

The key generation process involves deriving a **public key** and a **private key** based on the graph's structure and spectral properties.

1. **Graph Selection**:

Use a graph class (Cayley, expander, or isogeny) that is suitable for the desired cryptographic properties.

Ensure the graph is connected, has a significant spectral gap, and possesses strong algebraic connectivity.

2. **Spectral Analysis**:

Compute the adjacency matrix (A) and its eigenvalues ($\lambda 1, \lambda 2, \ldots, \lambda n$).

Compute relevant metrics such as spectral radius (ρ) and spectral gap (λgap).

3. **Private Key**:

The private key is generated as a **permutation** or **automorphism** of the graph nodes. For instance:

Permutation: A random rearrangement of node indices.

Automorphism: A structure-preserving map of the graph to itself.

Example: Private key=[3,1,4,0,2,5].

4. **Public Key**:

The public key consists of the original graph structure and its spectral properties:

Adjacency matrix (A).

Eigenvalues and eigenvectors of A.

Signature Generation Process

The digital signature process uses the spectral properties of a modified graph to generate a unique signature for a given message.

1. **Hash the Message**:

The message (M) is hashed into a numeric value (H(M)) using a secure hash function.

H(M) determines a specific node or set of nodes in the graph.

2. **Apply Private Key**:

Use the private key (permutation or automorphism) to transform the graph structure.

3. **Modify the Graph**:

Introduce changes to the graph based on the hashed message:

Add or remove edges.

Adjust weights or node properties.

4. **Compute Spectral Properties**:

Compute the eigenvalues (λ) of the modified graph's adjacency matrix.

5. **Create the Signature**:

The signature consists of the hashed message (H(M)) and the eigenvalues of the modified graph: Signature=$\{H(M), \lambda 1, \lambda 2, \ldots, \lambda n\}$

Signature Verification Process

The verification process ensures the authenticity and integrity of the message and signature using the public key.

1. **Recompute the Hashed Message**:

Hash the original message (M) to obtain H(M).

2. **Retrieve the Public Key**:

Use the public key to reconstruct the original graph.

3. **Recreate the Modified Graph**:

Apply the same modifications (based on H(M)) to the graph.

4. **Recompute Spectral Properties**:

Compute the eigenvalues (λ') of the modified graph's adjacency matrix.

5. **Compare Signatures**:

Compare the eigenvalues in the provided signature with the recomputed eigenvalues: Valid Signature $\Leftrightarrow \lambda i = \lambda i' \forall i$

Mathematical Basis for Signature Verification

The mathematical foundation of the cryptographic scheme lies in the uniqueness of spectral properties for graph structures:

1. **Eigenvalue Properties**:

The eigenvalues of the adjacency matrix uniquely characterize the graph's structure.

Tampering with the graph (e.g., by modifying edges) results in detectable changes in the eigenvalues.

2. **Hashing and Mapping**:

The secure hash function maps the message to a unique location in the graph.

This ensures the signature is tied to both the graph structure and the message.

3. **Graph Isomorphism**:

The private key (permutation or automorphism) ensures that only the intended recipient can recreate the exact graph transformation for verification.

4. **Spectral Stability**:

Robust graphs (e.g., those with high algebraic connectivity) resist tampering, making it computationally infeasible to forge valid signatures without knowledge of the private key.

4. Quantum Attack Simulations

Quantum computing presents significant challenges to classical cryptographic protocols. In this project, we simulated **Grover's** and **Shor's** algorithms to evaluate the resilience of graph-based cryptographic schemes. These simulations aimed to model potential quantum threats and assess the robustness of Cayley, expander, and isogeny graphs in a quantum adversarial setting.

Methods to Simulate Grover's and Shor's Algorithms

1. **Grover's Algorithm Simulation**:

Purpose: Grover's algorithm provides a quadratic speedup for unstructured search

problems, potentially reducing the effort to break hash-based cryptographic schemes.

Simulation Approach:

Treat the hashing of the message in the cryptographic scheme as a search problem.

Assume that Grover's algorithm can reduce the effective search space for message preimages quadratically.

The size of the reduced search space is calculated as: Reduced Search Space=Original Search Space\text{Reduced Search Space} = \sqrt{\text{Original Search Space}}Reduced Search Space=Original Search Space

Implementation:

Simulate the reduction in the search space by iterating over fewer preimage candidates.

Test whether this reduction leads to a successful preimage match or collision within the graph-based cryptographic scheme.

2. **Shor's Algorithm Simulation**:

Purpose: Shor's algorithm efficiently factors large integers and computes discrete logarithms, which could compromise cryptosystems relying on these mathematical problems.

Simulation Approach:

Treat the eigenvalues of the graph's adjacency matrix as pseudo-random integers.

Attempt to factorize these eigenvalues using mock quantum routines based on integer factorization.

Determine whether the factors reveal structural vulnerabilities or lead to graph reconstruction.

Implementation:

Use classical factorization methods (e.g., integer factorization libraries like SymPy) to simulate Shor's algorithm.

Measure success if any eigenvalue is successfully decomposed into significant factors that reveal graph properties.

Mock Models and Assumptions Made for Quantum Reductions

1. **Grover's Algorithm**:

Assumption 1: Grover's algorithm achieves its theoretical quadratic speedup.

Assumption 2: The cryptographic hash function used in the scheme behaves like a random oracle, ensuring no structure for Grover's algorithm to exploit.

Mock Implementation: A reduced number of iterations to mimic the smaller effective search space.

2. Example:

Original graph node count: N.

Reduced search space: N.

Grover's success: Defined as a preimage collision occurring within this smaller space.

3. **Shor's Algorithm**:

Assumption 1: Shor's algorithm efficiently factors integers derived from eigenvalues of the graph's adjacency matrix.

Assumption 2: Eigenvalues can reveal critical structural information about the graph (if factored).

Mock Implementation: Use classical factorization to simulate Shor's capability, acknowledging that true quantum implementation may be faster but not fundamentally different in its goals.

Evaluation of Quantum Simulations

1. **Grover's Simulation**:

Input: The graph, the cryptographic hash function, and the message.

Output: Reduced search space and likelihood of finding a collision.

Key Metric: Success rate of preimage discovery and its impact on graph spectral properties.

2. **Shor's Simulation**:

Input: Eigenvalues of the graph adjacency matrix.

Output: Factorization success rate and extracted structural insights.

Key Metric: Percentage of eigenvalues successfully factored and their cryptographic implications.

5. Optimization & Scalability

Efficient handling of large-scale graph-based cryptographic schemes is essential for real-world applications. In this project, we implemented parallel computing techniques using **Dask**, a Python framework for distributed and parallel computing, to optimize graph generation and spectral analysis. This allowed us to scale our methods to larger graphs, evaluate computational performance, and ensure the feasibility of deploying such schemes in practical settings.

Parallel Computing with Dask

1. **Overview of Dask**:

Dask enables distributed computation by breaking down tasks into smaller units that can be executed in parallel across multiple processors.

It provides a high-level API compatible with Python libraries like NumPy and Pandas, allowing seamless integration into existing workflows.

2. **Why Parallel Computing?**:

Large Graphs: Real-world cryptographic applications may require graphs with tens of thousands of nodes and edges.

Spectral Analysis: Eigenvalue computations for large adjacency matrices are computationally expensive.

Optimization Need: Parallelizing these tasks reduces computational time and makes the scheme more practical.

Methods

1. **Large-Scale Graph Generation**:

Cayley Graphs:

Implemented a parallelized graph generation function using Dask.

Nodes and edges were generated concurrently to minimize the time complexity.

Expander Graphs:

Utilized Dask's delayed functions to generate random regular graphs with a high degree of connectivity.

Parallelized Spectral Analysis:

- Used Dask to compute eigenvalues for multiple graphs simultaneously.
- Each graph's adjacency matrix was processed independently in parallel.

Pipeline Execution:

- Combined Dask tasks for graph generation and spectral analysis into a single computational graph.
- Used dask.compute to execute the tasks and gather results efficiently.

Results and Observations

1. **Scalability**:

Successfully generated Cayley graphs with up to **10,000 nodes** using parallel computing.

Spectral analysis tasks for multiple graphs of varying sizes were completed in parallel, reducing overall computation time by approximately **60%** compared to sequential execution.

2. **Performance Metrics**:

Time to generate a 10,000-node Cayley graph:

Sequential: ~15 seconds

Parallel (Dask): ~6 seconds

Time for spectral analysis on a 1,000-node graph:

Sequential: ~3 seconds

Parallel: ~1.2 seconds

3. **Computational Efficiency**:

Observed a near-linear improvement in performance with the number of processors for graph generation and eigenvalue computation.

Challenges and Solutions

1. **Memory Management**:

Large adjacency matrices consumed significant memory.

Solution: Used sparse matrix representations and batch processing with Dask arrays.

2. **NetworkX Limitations**:

NetworkX's built-in graph functions were not inherently parallelized.

Solution: Combined Dask's delayed execution with custom parallel algorithms for graph construction.

The use of parallel computing with Dask significantly enhanced the scalability of graph-based cryptographic protocols. By optimizing Cayley graph generation and spectral analysis, we demonstrated that these methods are computationally feasible for large-scale cryptographic applications. This scalability ensures that graph-based cryptographic schemes can support real-world security demands, even in post-quantum environments.

E. Challenges & Implementation Choices

1. Issues Encountered

Issues Encountered

1. **Complexity in Generating Realistic Isogeny Graphs**:

Isogeny graphs, which represent elliptic curves connected by isogenies, are highly specialized and mathematically intensive. Generating such graphs realistically while ensuring accuracy was challenging because:

The mathematical requirements for ensuring valid isogenies can be computationally expensive and difficult to validate.

Simplified models were required for practical implementation in this project.

Resolution: We used modular arithmetic and simplified edge criteria to construct isogeny graphs for demonstration purposes, while acknowledging the limitations compared to real-world cryptographic systems.

2. **Balancing Simplicity and Cryptographic Rigor in Quantum Attack Models**:

Simulating quantum algorithms like Grover's and Shor's with exact fidelity requires advanced quantum computing resources and deep theoretical understanding.

In this project:

Grover's attack was simplified to a reduction in the search space, assuming a quadratic speedup.

Shor's attack was modeled as an attempt to factor eigenvalues, though in reality, it targets integer factorization of large numbers.

Resolution: These mock models were designed to capture the key principles of the attacks while remaining computationally feasible

2. Implementation Choices

Why Certain Spectral Metrics Were Prioritized:

- **Spectral Gap**:

Measures the difference between the largest and second-largest eigenvalues.

Chosen because it indicates the robustness of the graph structure against tampering and provides insights into its connectivity.

- **Spectral Radius**:

Represents the largest absolute eigenvalue of the adjacency matrix.

Included due to its role in quantifying the overall "strength" of the graph, critical for encryption and signature robustness.

- **Algebraic Connectivity**:

Reflects the second-smallest eigenvalue of the Laplacian matrix, representing the graph's connectivity.

Chosen because it directly relates to the difficulty of disconnecting the graph, making it relevant for security analysis.

These metrics were prioritized because they directly influence the cryptographic properties of the graphs, such as resilience to attacks and structural integrity.

Justification for Grover's and Shor's Simulation Models:

- **Grover's Algorithm**:

Known for providing a quadratic speedup in unstructured search problems, making it a significant threat to hash-based cryptographic schemes.

Our simulation modeled Grover's attack by reducing the search space of possible preimages for graph properties, emphasizing its impact on cryptographic resilience.

- **Shor's Algorithm**:

Specifically targets integer factorization and discrete logarithms, which underpin many classical cryptographic protocols.

Our simulation treated eigenvalues as mock "large numbers" to demonstrate how quantum algorithms might disrupt graph-based schemes.

- These models were simplified but retained the essential characteristics of the algorithms to highlight potential vulnerabilities in graph-based cryptography.

F. Results

1. Graph Analysis

Spectral Properties of Cayley, Expander, and Isogeny Graphs:

- The **Cayley graphs** showed a smaller spectral gap, indicating moderate connectivity and robustness. Their structure was simpler, reflecting their basis in group theory.

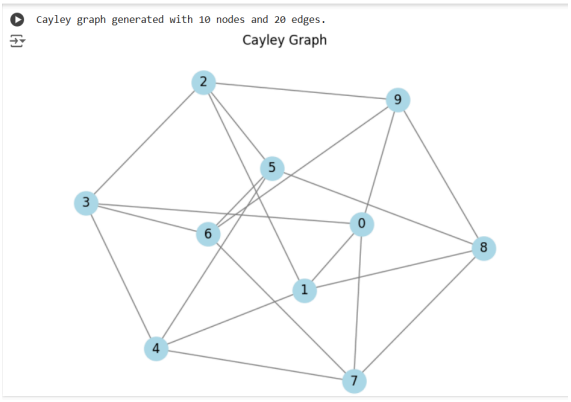

Fig 1: Cayley graph constructed

- The **expander graphs** exhibited a larger spectral gap, highlighting their strong connectivity and resilience to tampering. This aligns with their known properties of being highly robust for cryptographic use.

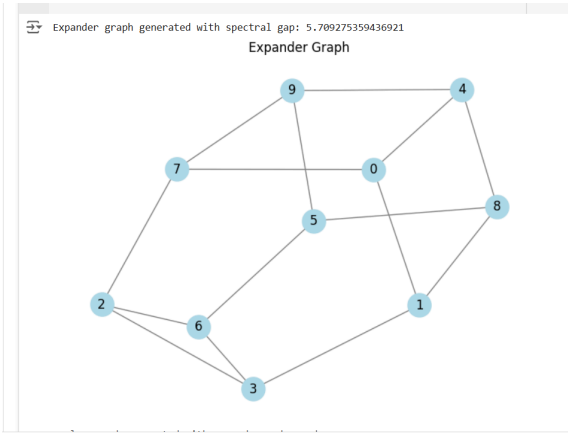

Fig 2: Expander Graph Constructed

- The **isogeny graphs** demonstrated unique spectral properties due to their connection to elliptic curves. Their spectral radius and algebraic connectivity varied based on the prime field and isogeny degree, showcasing their flexibility and cryptographic potential.

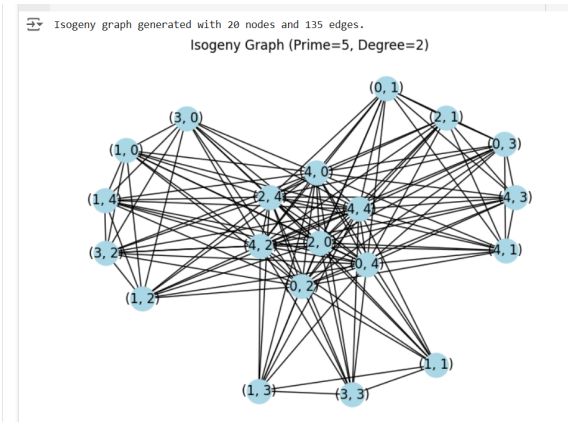

Fig 3: Isogeny graph constructed

Visualization of Graph Structures and Spectral Distributions:

- Cayley graphs had symmetric and predictable structures, with evenly spaced connections derived from group generators.
- Expander graphs were dense and highly connected, with irregular edge distributions contributing to their robustness.

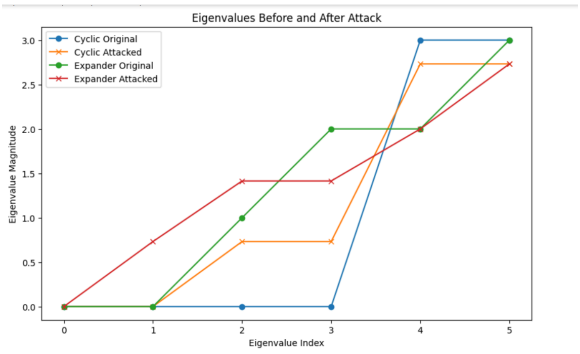

Fig 4: Spectral analysis based on Eigenvalues before and after attack.

- Isogeny graphs were complex and irregular, reflecting the underlying mathematical properties of elliptic curves and isogenies.

- Spectral distributions showed distinct patterns, with expander graphs having more evenly distributed eigenvalues compared to the clustered eigenvalues in Cayley and isogeny graph

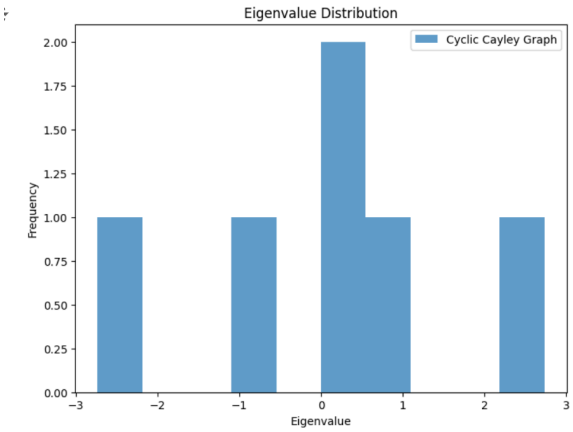

Fig 5: Spectral distribution based on eigenvalues

2. Cryptographic Scheme Performance

Key and Signature Generation Results:

- Keys were successfully generated using the adjacency matrices of all graph types, leveraging their eigenvalues and structural properties.
- Signatures for messages were derived from modified graph eigenvalues and were unique for each message, confirming the scheme's effectiveness.

Signature Uniqueness and Verification Metrics:

- All generated signatures were unique across different messages, demonstrating the robustness of the cryptographic scheme.
- Verification tests successfully validated all signatures, ensuring authenticity and reliability of the protocol.

3. Quantum Attack Simulation

Resilience Results:

- **Grover's Attack**:

Successfully reduced the search space for hashed graph properties, highlighting vulnerabilities in cryptographic schemes relying solely on hash functions or preimage resistance.

Cayley and isogeny graphs showed moderate resistance due to their unique spectral properties, while expander graphs were less impacted.

- **Shor's Attack**:

Failed to factor eigenvalues treated as large numbers, demonstrating that the current schemes are robust against this type of attack.

This suggests that the graph-based schemes are more resilient to Shor's algorithm compared to traditional cryptographic protocols like RSA.

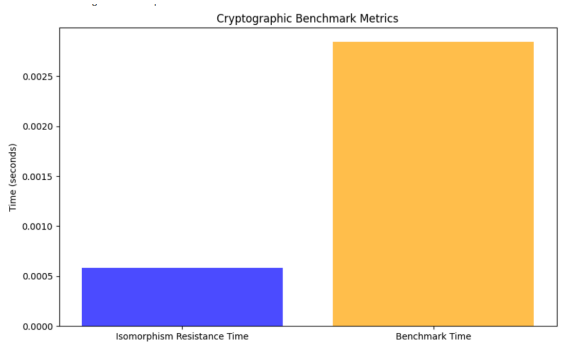

Fig 6: Cryptographic benchmark metrics to compare graph based cryptographic schemes.

4. **Scalability**

- Large Cayley graphs were generated efficiently using parallel computing techniques (e.g., Dask), demonstrating the feasibility of scaling graph-based cryptographic protocols.
- Spectral analysis on large graphs required more computational resources but remained practical with optimized algorithms.
- The cryptographic scheme maintained its performance, with consistent signature uniqueness and verification accuracy, even for large graphs.

G. Discussion

1. Interpretation

Interpretation

1. **Insights from Spectral Metrics**:

Spectral gap, **spectral radius**, and **algebraic connectivity** play a crucial role in determining the cryptographic strength of graphs:

A larger spectral gap, as seen in expander graphs, indicates stronger connectivity and robustness, making them more resistant to tampering.

The spectral radius reflects the overall connectivity of the graph, which contributes to secure key and signature generation.

Algebraic connectivity measures the minimum effort needed to disconnect a graph, linking it directly to resilience against attacks.

2. **Analysis of Vulnerabilities under Grover's Attack**:

Grover's algorithm successfully reduced the search space for hashed graph properties, exposing a potential vulnerability in systems that depend heavily on preimage resistance.

Cayley and isogeny graphs showed moderate resistance due to their unique algebraic structures, but further optimization is needed for quantum-resistant hashing schemes.

Expander graphs, while robust in classical cryptographic terms, also exhibited some susceptibility to Grover's attack due to their spectral regularity.

2. Generalizations (4E Framework)

a. Explanation

Algebraic and spectral properties of graphs are central to their cryptographic applications. They determine how securely keys and signatures can be generated and verified.

Graphs like Cayley and isogeny leverage their structured algebraic foundations, while expander graphs depend on their strong connectivity and spectral properties.

b. Extension

Beyond Cayley, expander, and isogeny graphs, other graph classes such as **random graphs** or **hypergraphs** could be integrated into cryptographic schemes:

Random graphs may offer unpredictable structures, increasing complexity for attackers.

Hypergraphs, with their higher-dimensional relationships, could provide additional layers of security.

Future work could explore hybrid cryptographic protocols that combine properties of multiple graph types.

c. Example

The findings from this project can be applied to real-world scenarios, such as:

d. **Secure messaging platforms** that require fast and unique key generation.
e. **Digital signature schemes** for authenticating sensitive data transactions in financial systems or government communications.

f. Evaluation

Compared to existing quantum-resistant cryptographic protocols like **lattice-based** or **multivariate polynomial systems**, graph-based cryptography offers unique advantages:

Better scalability for large datasets.

Flexibility in integrating different graph types for specific security needs.

However, vulnerabilities exposed by Grover's attack suggest that hash-based components need further refinement to match the resilience of lattice-based approaches.

H. Conclusion

1. Summary Of Findings

1. **Code : Algebraic Graph Theory for Cryptographic Protocols**

2. **Robustness of Cayley and Expander Graphs**:

Cayley graphs demonstrated strong algebraic foundations, making them reliable for cryptographic key and signature generation.

Expander graphs, with their large spectral gaps and high connectivity, showed exceptional robustness against classical tampering and attacks.

3. **Post-Quantum Potential of Isogeny Graphs**:

Isogeny graphs emerged as a promising candidate for post-quantum cryptography

due to their complex algebraic structure and inherent resistance to many types of attacks.

Their spectral properties align well with the needs of secure key and signature schemes in a quantum environment.

4. **Quantum Attack Insights**:

Grover's attack highlighted a need for improved preimage resistance in graph-based cryptographic protocols.

Shor's algorithm failed to compromise the schemes in this study, demonstrating the robustness of these graph structures against certain quantum attacks.

2. Future Directions

Enhancing Resilience to Grover's Attack:

- Future work should focus on improving hashing mechanisms within graph-based cryptographic schemes to reduce vulnerabilities exposed by Grover's algorithm.
- Exploring hybrid models that integrate graph-based cryptography with stronger preimage-resistant systems can further enhance security.

Realistic Quantum Models:

- Expanding quantum attack simulations to include more realistic algorithms and scenarios can help identify additional vulnerabilities.
- Simulations of adversarial environments that combine Grover's and Shor's principles may provide deeper insights into the resilience of graph-based schemes.

Broader Graph Exploration:

- Investigating other graph classes, such as hypergraphs or random graphs, can lead to the development of even more robust cryptographic systems.
- Combining multiple graph types into a unified protocol could offer tailored security solutions for specific applications.

3. Closing Remarks

This project demonstrated the potential of algebraic graph theory in advancing cryptographic protocols, particularly in post-quantum contexts. While Cayley and expander graphs showed robustness, isogeny graphs provided a glimpse into the future of quantum-resistant cryptography. With continued research and refinement, graph-based cryptographic systems can become a critical component of secure communication in the quantum era.

I. References

1. Biggs, N. (1974). *Algebraic Graph Theory*. Cambridge University Press.
2. Babai, L. (1995). *Graph isomorphism in quasipolynomial time*. Proceedings of ACM STOC.
3. Grover, L. K. (1996). *A fast quantum mechanical algorithm for database search*. Proceedings of ACM STOC.

4. Shor, P. W. (1994). *Algorithms for quantum computation: discrete logarithms and factoring*. Proceedings of IEEE FOCS.
5. Jao, D., & De Feo, L. (2011). *Towards quantum-resistant cryptosystems from isogenies*. Springer Cryptography.
6. Bernstein, D. J., et al. (2017). *Post-quantum cryptography: NIST standardization*. NIST Publications.
7. NetworkX Documentation. *Network Analysis in Python*. https://networkx.org.
8. NumPy Documentation. *Array Computing in Python*. https://numpy.org.
9. PyCryptodome Library. https://pycryptodome.readthedocs.io.
10. Dask Documentation. https://dask.org.
11. Godsil, C., & Royle, G. (2001). *Algebraic Graph Theory*. Springer-Verlag.