**Real-Time Secure Communication using Sequential Half-Aggregation of Lattice-Based Signatures**

**Team Members:**

1. E. Sarath Chandra (CH.EN.U4CYS21017),
2. K. Shanmukha Sai (CH.EN.U4CYS21032)

**Supervisor:** Dr. Dharminder Chaudhary

**Title:** “Real-Time Secure Communication using Sequential Half-Aggregation of Lattice-Based Signatures.”

**Importance of proposed project:**

***Context and relevance:***

In the digital age, secure communication is critical for protecting sensitive information and ensuring privacy. With the advent of quantum computing, traditional cryptographic methods such as RSA and ECC are at risk of being broken. Lattice-based cryptography offers a promising alternative that is resistant to quantum attacks, making it a crucial area of research and application.

***Project Significance:***

This project aims to implement a web application that leverages the advanced cryptographic techniques discussed in the paper "Sequential Half-Aggregation of Lattice-Based Signatures" by Katharina Boudgoust and Akira Takahashi. The significance of this project lies in its potential to:

* *Enhance Security:* Implementing lattice-based cryptography for authentication and communication ensures robust security against both classical and quantum threats.
* *Promote Efficiency:* Using sequential half-aggregation allows for more efficient signature schemes, reducing data transmission size and computational overhead.
* *Advance Cryptographic Research:* Demonstrating a practical implementation of advanced cryptographic techniques contributes to the broader field of cryptographic research and its applications.

**Project planning & Methodology:**

***Phase 1: Research and Requirement Analysis***

* *Literature Review:* Study the paper and related works on lattice-based cryptography and aggregate signatures.
* *Requirement Gathering:* Define the functional and non-functional requirements of the web application.

***Phase 2: Design***

* *System Architecture:* Design the overall architecture, including frontend, backend, and database.
* *Cryptographic Design:* Design the key generation, authentication, and communication protocols based on lattice-based signatures.

***Phase 3: Implementation***

* *Backend Development:* Implement the server-side logic for key generation, authentication, and real-time communication using Node.js or Python.
* *Frontend Development:* Develop the user interface using React or Angular, incorporating WebSocket for real-time communication.
* *Integration:* Ensure seamless integration between the frontend and backend components.

***Phase 4: Testing***

* *Unit Testing:* Test individual components for correctness.
* *Integration Testing:* Test the interaction between different components.
* *Security Testing:* Conduct thorough security testing to identify and mitigate vulnerabilities.

***Phase 5: Deployment and Evaluation***

* *Deployment:* Deploy the application on a cloud platform like AWS or Google Cloud.
* *User Testing:* Conduct user testing to gather feedback and make necessary improvements.
* *Performance Evaluation:* Evaluate the performance and efficiency of the application.

**Key References:**

***Primary Reference:***

* Boudgoust, K., & Takahashi, A. (2023). Sequential Half-Aggregation of Lattice-Based Signatures.

***Supporting Reference:***

* Alkim, E., Barbosa, M., Orsini, E., & Rijneveld, J. (2021). The lattice-based digital signature scheme Dilithium. In International Conference on Cryptographic Hardware and Embedded Systems (pp. 1-23).
* Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., & Stehlé, D. (2018). CRYSTALS-Dilithium: A lattice-based digital signature scheme. IACR Transactions on Cryptographic Hardware and Embedded Systems, 2018(1), 238-268.
* M. R. Albrecht, V. Cini, R. W. F. Lai, G. Malavolta, and S. A. Thyagarajan. Lattice-based SNARKs: Publicly verifiable, preprocessing, and recursively composable. Cryptology ePrint Archive, Report 2022/941, 2022
* K. Boudgoust, C. Jeudy, A. Roux-Langlois, and W. Wen. On the hardness of module learning with errors with short distributions. J. Cryptol., 36(1):1, 2023.
* K. Chalkias, F. Garillot, Y. Kondi, and V. Nikolaenko. Non-interactive half-aggregation of EdDSA and variants of Schnorr signatures. In CT-RSA 2021, vol. 12704 of LNCS, pp. 577–608. Springer, Heidelberg, 2021.
* M. Drijvers, K. Edalatnejad, B. Ford, E. Kiltz, J. Loss, G. Neven, and I. Stepanovs. On the security of two-round multi-signatures. In 2019 IEEE Symposium on Security and Privacy, pp. 1084–1101. IEEE Computer Society Press, 2019.
* C. Gentry, A. O’Neill, and L. Reyzin. A unified framework for trapdoor-permutation-based sequential aggregate signatures. In PKC 2018, Part II, vol. 10770 of LNCS, pp. 34–57. Springer, Heidelberg, 2018.

**Knowledge:**

The journal paper titled "Sequential Half-Aggregation of Lattice-Based Signatures" by Katharina Boudgoust and Akira Takahashi provides an in-depth discussion on the construction and security of lattice-based aggregate signatures, specifically focusing on the Fiat-Shamir with Aborts (FSwA) framework.

Here's a breakdown of the key concepts and steps you can follow to achieve your project goals:

***Understanding the Basics:***

* Lattice-Based Signatures: These are cryptographic signatures based on the hardness of lattice problems, which are believed to be secure against quantum attacks. The paper discusses two main types: Dilithium (FSwA-based) and Falcon (GPV-paradigm-based).
* Aggregate Signatures: These allow multiple signatures on different messages to be combined into a single, compact signature. This is useful for reducing the size of transmitted data in applications like certificate chains.
* Sequential Aggregate Signatures (SAS): In SAS, signatures are aggregated in a sequential manner, with each signer adding their signature to the aggregate. This fits well in scenarios requiring round-robin communication.

***Key Sections to Focus On:***

Section 2: Preliminaries:

Provides foundational knowledge on probability, module lattice problems, and the basics of FSwA signatures.

Introduces SAS and their properties.

Section 3: Sequential Half-Aggregation of FSwA Signatures:

Definition and Correctness of the Scheme: Describes the formal definition and correctness proofs for the proposed sequential half-aggregate signature scheme.

Security Proof: Details the security proof, showing that the scheme is secure under certain cryptographic assumptions.

Section 4: Performance Estimates and Comparison:

Performance Estimates: Evaluates the performance of the proposed scheme, including signature size and compression rates.

Comparison With SAS Using Trapdoors: Compares the new scheme with existing SAS schemes based on different paradigms.

Section 5: Attacks on Existing Schemes:

Analyzes vulnerabilities in existing lattice-based SAS schemes and proposes solutions to improve security.

***Steps to Implement the Scheme:***

Study the Preliminaries: Make sure you understand the underlying lattice problems and the basics of FSwA signatures. This will help you grasp the construction and security proofs.

**Implement the Sequential Half-Aggregation:**

Follow the detailed construction in Section 3 to implement the sequential half-aggregate signature scheme.

Pay special attention to the aggregation of u and z parts of the signature, as described in the paper.

**Validate the Scheme:**

Use the correctness proofs provided to ensure your implementation is accurate.

Implement the security checks as outlined in the security proof to verify the robustness of your scheme.

**Performance Testing:**

Conduct performance tests to compare the size and efficiency of your aggregate signatures against the naive concatenation of individual signatures.

Use the estimates in Section 4 as benchmarks.

**Review and Address Security Concerns:**

Analyze potential attacks on your implementation using the insights from Section 5.

Apply any necessary fixes to enhance the security of your scheme.

*By following these steps and thoroughly understanding the concepts and methods presented in the paper, you should be able to develop a secure and efficient key generation and authentication mechanism based on lattice-based aggregate signatures. If you have specific questions or need further clarification on any part of the paper, feel free to ask!*