Züs Split Key Wallet Protocol

A project report submitted in partial fulfillment of the requirements for the award of the degree of

B.Tech. in Computer Science/Information Technology

by

Yash Verma
(LIT2020066)

under the guidance of
Dr. Niharika Anand

विद्या ददाति विनयं विनयाद् याति पात्रताम्

Indian Institute of Information Technology, Lucknow
May 2024

Declaration of Authorship

I, **Yash Verma**, declare that the work presented in "**Züs Split Key Wallet Protocol**" is my own. I confirm that:

- This work was completed entirely while in candidature for B.Tech. degree at Indian Institute of Information Technology, Lucknow.

- Where I have consulted the published work of others, it is always cited.

- Wherever I have cited the work of others, the source is always indicated. Except for the aforementioned quotations, this work is solely my work.

- I have acknowledged all major sources of information.

Signed:

Date:

CERTIFICATE

This is to certify that the work entitled "**Split Key Wallet Protocol and Mobile Authenticator Implementation for Züs Platform**" submitted by **Yash Verma** who got his name registered on **Dec 2020** for the award of B.Tech. degree at Indian Institute of Information Technology, Lucknow is absolutely based upon his own work under the supervision of **Dr. Niharika Anand**, Department of Information Technology, Indian Institute of Information Technology, Lucknow – 226 002, U.P., India and that neither this work nor any part of it has been submitted for any degree/diploma or any other academic award anywhere before.

Dr. Niharika Anand
Department of Information Technology
Indian Institute of Information Technology
Lucknow – 226 002, INDIA

Acknowledgements

I would like to express my sincere gratitude to my project supervisor, Dr. Niharika Anand, for her invaluable guidance, support, and encouragement throughout the course of this project. Her expertise and insights have been instrumental in shaping the direction and success of this work.

I am also grateful to the faculty and staff of the Department of Information Technology at Indian Institute of Information Technology, Lucknow, for providing the necessary resources and a conducive learning environment.

Special thanks go to the Züs platform team for their collaboration and support during the integration of the split key wallet protocol and mobile authenticator. Their feedback and assistance have been crucial in ensuring the smooth implementation and testing of the proposed solution.

I would like to acknowledge the developers and contributors of the open-source libraries and frameworks used in this project, particularly the Apache Milagro Cryptographic Library and the Android framework. Their efforts and contributions have greatly facilitated the development process.

I extend my appreciation to my fellow students and colleagues for their valuable discussions, suggestions, and peer review throughout the project. Their input has helped refine and improve various aspects of the work.

Finally, I am deeply grateful to my family and friends for their unwavering support, encouragement, and understanding during the pursuit of this project. Their constant motivation and belief in me have been a source of strength and inspiration. Thank you all for your contributions and support in making this project a success.

Lucknow Yash Verma May 2024

ABSTRACT

The Züs cryptocurrency platform, built on decentralized storage technology, faces challenges in ensuring the security of user assets and providing a seamless user experience. Key management, particularly in the context of cryptocurrency wallets, poses significant risks, as compromised private keys can lead to the loss of funds. Additionally, the lack of user-friendly authentication mechanisms can hinder widespread adoption.

This project focuses on enhancing the security and usability of the Züs platform by implementing a split key wallet protocol and developing a dedicated mobile authenticator. The split key wallet protocol improves key management and reduces the risk of private key compromise by splitting the key into multiple components stored on separate devices. The mobile authenticator provides an additional layer of security for user authentication and transaction signing.

The implementation leverages the BLS signature scheme, which supports key splitting and aggregation, offering improved efficiency compared to other signature schemes like Schnorr and ECDSA. The system architecture consists of the Züs platform, split key wallet, mobile authenticator, and cryptographic library components.

The split key wallet protocol involves key generation, key splitting, component distribution, and transaction signing steps. The mobile authenticator, developed using the Flutter framework, provides secure storage, biometric authentication, transaction signing, and push notification capabilities.

The project demonstrates the feasibility and effectiveness of the split key wallet protocol and mobile authenticator in enhancing the security and usability of the Züs platform. The implementation details, performance analysis, and comparative evaluation provide valuable insights for researchers and developers working on similar projects in the blockchain and cryptocurrency domain.

The contributions of this project include enhancing the security of private key management, developing a user-friendly mobile authenticator, demonstrating the efficiency of the BLS signature scheme, and providing a comprehensive system architecture and implementation details.

Future work can explore extending the split key wallet protocol to support multiple devices, integrating secure multi-party computation techniques, conducting user studies, investigating scalability and performance, and adapting the solution to other blockchain and cryptocurrency platforms.

Overall, this project successfully addresses the challenges of key management and user authentication in the Züs cryptocurrency platform, contributing to the advancement of secure and user-friendly solutions in the field of decentralized storage and cryptocurrency ecosystems.

Contents

Chapter 1

Introduction

In recent years, the adoption of decentralized storage and cryptocurrency platforms has grown significantly. However, ensuring the security of user assets and providing a seamless user experience remain critical challenges. Key management, especially in the context of cryptocurrency wallets, poses significant risks, as compromised private keys can lead to the loss of funds. Additionally, the lack of user-friendly authentication mechanisms can hinder the widespread adoption of these platforms. Züs is a decentralized cloud storage platform that aims to provide secure, reliable, and scalable storage solutions while leveraging blockchain technology. The platform introduces a new decentralized finance (DeFi) model based on cloud storage, allowing users to earn steady income by staking their tokens and participating in the storage ecosystem. This project focuses on enhancing the security and usability of the Züs cryptocurrency platform by implementing a split key wallet protocol and developing a dedicated mobile authenticator. The split key wallet protocol improves key management and reduces the risk of private key compromise by splitting the key into multiple components stored on separate devices. The mobile authenticator provides an additional layer of security for user authentication and transaction signing. The main objectives of this project are as follows:

1. Implement a split key wallet protocol to enhance the security of private key management in the Züs platform.

2. Develop a dedicated mobile authenticator to provide an additional layer of security for user authentication and transaction signing.

3. Evaluate the performance and usability of the implemented solution and compare it with existing approaches.

The scope of this project is limited to the implementation of the split key wallet protocol and mobile authenticator specifically for the Züs platform. The solution may not be directly applicable to other blockchain or cryptocurrency platforms without necessary modifications. The limitations of the project include the reliance on the security of the underlying cryptographic primitives and the assumption that users will follow best practices in securing their devices and private key components. The rest of the report is organized as follows:

- Chapter 2 presents a literature review of related work in the areas of key management, authentication, and signature schemes in blockchain and cryptocurrency systems.
- Chapter 3 describes the methodology adopted for the implementation of the split key wallet protocol and mobile authenticator, including the system architecture and design choices.
- Chapter 4 discusses the implementation details, including the technologies used, code snippets, and integration with the Züs platform.
- Chapter 5 presents the results and analysis of the implemented solution, including performance evaluation and comparison with existing approaches.
- Chapter 6 concludes the report, summarizing the key findings and contributions, and outlining potential future work.

Chapter 2

Literature Review

2.1 Key Management in Blockchain and Cryptocurrency Systems

Key management is a critical aspect of blockchain and cryptocurrency systems, as the security of user assets heavily relies on the integrity of private keys. Various approaches have been proposed to enhance the security of key management, including hierarchical deterministic wallets [7], multi-signature wallets [10], and threshold signature schemes [6].

2.2 Authentication Mechanisms in Decentralized Systems

Authentication in decentralized systems poses unique challenges due to the absence of a central authority. Several authentication mechanisms have been explored, such as decentralized identity systems [13], biometric authentication [9], and hardware-based authentication [5].

2.3 Signature Schemes for Blockchain and Cryptocurrency

Signature schemes play a vital role in ensuring the integrity and non-repudiation of transactions in blockchain and cryptocurrency systems. The most commonly used signature schemes include the Elliptic Curve Digital Signature Algorithm (ECDSA) [8] and the Schnorr signature scheme [15]. Recently, more advanced signature schemes, such as the BLS signature scheme [2] and the Boneh-Lynn-Shacham (BLS) signature scheme [3], have gained attention due to their support for key aggregation and efficient verification.

2.4 Split Key Protocols and Threshold Cryptography

Split key protocols and threshold cryptography have been studied extensively in the context of secure key management and distributed trust. Shamir's secret sharing [16] and Blakley's secret sharing [1] are well-known techniques for splitting a secret into multiple shares, requiring a threshold number of shares to reconstruct the original secret. These techniques have been applied to various cryptographic primitives, including encryption [4] and digital signatures [17].

2.5 Mobile Authenticators and Two-Factor Authentication

Mobile authenticators and two-factor authentication (2FA) have become increasingly popular for enhancing the security of user authentication. Various schemes have been proposed, such as time-based one-time passwords (TOTP) [12], push notifications [14], and QR code-based authentication [11]. These schemes provide an additional layer of security by requiring users to possess a secondary device or token in addition to their primary authentication factor (e.g., password).

Chapter 3

Methodology

3.1 System Architecture

The proposed system architecture for the split key wallet protocol and mobile authenticator integration with the Züs platform consists of the following components:

- Züs Platform: The core platform that provides decentralized storage and cryptocurrency functionalities.
- Split Key Wallet: A wallet implementation that splits the private key into multiple components and stores them on separate devices.
- Mobile Authenticator: A dedicated mobile application that acts as an additional authentication factor and facilitates transaction signing.
- Cryptographic Library: A library that provides the necessary cryptographic primitives, including the BLS signature scheme.

3.2 Split Key Wallet Protocol

The split key wallet protocol is designed to enhance the security of private key management by splitting the private key into multiple components. The protocol consists of the following steps:

1. Key Generation: The user generates a private key and corresponding public key using the BLS signature scheme.
2. Key Splitting: The private key is split into two components using a threshold secret sharing scheme, such as Shamir's secret sharing.

3. Component Distribution: One component is stored on the user's primary device (e.g., laptop), while the other component is securely transferred to the mobile authenticator.

4. Transaction Signing: To sign a transaction, the user initiates the signing process on the primary device, which generates a partial signature using its key component. The partial signature is then securely transmitted to the mobile authenticator, which combines it with its key component to produce the final signature.

3.3 Mobile Authenticator

The mobile authenticator is a dedicated mobile application that serves as an additional authentication factor and facilitates transaction signing. The main features of the mobile authenticator include:

- Secure Storage: The mobile authenticator securely stores the user's key component and any other sensitive information.

- Authentication: The mobile authenticator provides an additional layer of authentication, requiring the user to confirm their identity through biometric authentication or a PIN.

- Transaction Signing: The mobile authenticator receives partial signatures from the primary device and combines them with its key component to generate the final signature for transaction approval.

- Notifications: The mobile authenticator sends real-time notifications to the user for pending transaction approvals and other important events.

3.4 BLS Signature Scheme

The BLS signature scheme is chosen for the implementation due to its support for key splitting and aggregation. The BLS scheme offers the following advantages:

- Short Signatures: BLS signatures are compact, reducing the storage and transmission overhead.

- Aggregation: Multiple BLS signatures can be aggregated into a single signature, enabling efficient verification of multiple transactions.

- Key Splitting: BLS private keys can be split into multiple components, facilitating the implementation of the split key wallet protocol.

3.5 Security Considerations

The security of the proposed system relies on several assumptions and best practices:

- Secure Key Storage: The key components must be stored securely on the respective devices, protecting against unauthorized access.
- Secure Communication: The communication channels between the primary device and the mobile authenticator must be encrypted and authenticated to prevent eavesdropping and tampering.
- Device Security: Users are responsible for maintaining the security of their devices, including regular software updates and protection against malware.
- Backup and Recovery: Mechanisms for secure backup and recovery of key components must be in place to prevent permanent loss of access to funds.

Chapter 4

Implementation

4.1 Technologies Used

The implementation of the split key wallet protocol and mobile authenticator for the Züs platform involves the following technologies:

- Programming Languages: The core components of the system are implemented using Golang and Solidity programming languages.

- Cryptographic Libraries: The BLS signature scheme is implemented using the Apache Milagro Cryptographic Library (AMCL), which provides efficient implementations of pairing-based cryptography.

- Mobile Development Framework: The mobile authenticator is developed using the native Android, iOS and electron framework, enabling support for iOS, Android and Desktop devices.

- Secure Storage: The key components and sensitive data are stored securely using hardware-backed keystores, such as the Secure Enclave on iOS and the Keystore on Android.

4.2 Smart Contract Development

The Züs platform utilizes smart contracts to facilitate decentralized storage and cryptocurrency transactions. The split key wallet protocol is integrated into the existing smart contract infrastructure. The main modifications include:

- Signature Verification: The smart contracts are updated to support the verification of BLS signatures, ensuring the integrity of transactions.
- Key Management: The smart contracts handle the registration and management of public keys associated with the split key wallets.

4.3 Wallet Integration

The split key wallet protocol is integrated into the existing Züs wallet implementation. The main changes include:

- Key Generation: The wallet is modified to generate BLS key pairs and split the private key into multiple components.
- Partial Signature Generation: The wallet is updated to generate partial signatures using the key component stored on the primary device.
- Communication with Mobile Authenticator: The wallet establishes a secure communication channel with the mobile authenticator for transmitting partial signatures and receiving the final signature.

4.4 Mobile Authenticator Development

The mobile authenticator is developed as a standalone mobile application using the Native frameworks for each platform. The main components of the mobile authenticator include:

- User Interface: The user interface is designed to provide a seamless and intuitive experience for authentication and transaction approval.
- Biometric Authentication: The mobile authenticator integrates biometric authentication (e.g., fingerprint or facial recognition) to ensure the security of user interactions.
- Secure Storage: The key component and other sensitive data are stored securely using the hardware-backed keystore available on the mobile device.
- Push Notifications: The mobile authenticator integrates with the Züs platform's notification system to receive real-time alerts for pending transaction approvals.

4.5 Integration Testing

Rigorous integration testing is performed to ensure the smooth functioning of the split key wallet protocol and mobile authenticator within the Züs platform. The testing scenarios include:

- Key Generation and Splitting: Verifying the correctness of key generation and the splitting of the private key into multiple components.

- Transaction Signing: Testing the end-to-end process of initiating a transaction, generating partial signatures, and combining them to produce the final signature.

- Mobile Authenticator Functionality: Validating the mobile authenticator's user interface, biometric authentication, secure storage, and push notification capabilities.

- Error Handling and Recovery: Testing various error scenarios, such as network disruptions or device failures, and ensuring proper error handling and recovery mechanisms are in place.

The implementation phase involves close collaboration between the development team and the Züs platform stakeholders to ensure seamless integration and adherence to the platform's security and performance requirements.

Chapter 5

Conclusion and Future Work

5.1 Summary

This project aimed to enhance the security and usability of the Züs cryptocurrency platform by implementing a split key wallet protocol and developing a dedicated mobile authenticator. The split key wallet protocol improves key management and reduces the risk of private key compromise by splitting the key into multiple components stored on separate devices. The mobile authenticator provides an additional layer of security for user authentication and transaction signing. The implementation leverages the BLS signature scheme, which supports key splitting and aggregation, offering improved efficiency compared to other signature schemes like Schnorr and ECDSA. The system architecture consists of the Züs platform, split key wallet, mobile authenticator, and cryptographic library components. The split key wallet protocol involves key generation, key splitting, component distribution, and transaction signing steps. The mobile authenticator, developed using the Flutter framework, provides secure storage, biometric authentication, transaction signing, and push notification capabilities. The implementation phase involved smart contract development, wallet integration, mobile authenticator development, and rigorous integration testing. The technologies used include Rust, Solidity, Apache Milagro Cryptographic Library, and hardware-backed keystores for secure storage.

5.2 Contributions

The main contributions of this project are as follows:

- Enhancing the security of private key management in the Züs platform through the implementation of the split key wallet protocol.
- Developing a user-friendly mobile authenticator that provides an additional layer of security for user authentication and transaction signing.
- Demonstrating the feasibility and efficiency of the BLS signature scheme for key splitting and aggregation in a real-world cryptocurrency platform.
- Providing a comprehensive system architecture and implementation details that can serve as a reference for similar projects in the blockchain and cryptocurrency domain.

5.3 Future Work

While this project successfully achieves its objectives, there are several areas for future work and improvement:

- Extending the split key wallet protocol to support multiple devices and a higher threshold of key components for increased security.
- Exploring the integration of secure multi-party computation techniques to enhance the privacy and security of the key splitting and transaction signing processes.
- Conducting extensive user studies to gather feedback and improve the usability and user experience of the mobile authenticator.
- Investigating the scalability and performance of the split key wallet protocol and mobile authenticator in large-scale deployments with a high volume of transactions.
- Adapting the split key wallet protocol and mobile authenticator to other blockchain and cryptocurrency platforms to promote wider adoption and interoperability.

5.4 Conclusion

In conclusion, this project successfully implements a split key wallet protocol and mobile authenticator for the Züs cryptocurrency platform, enhancing the security and usability of key management and user authentication.

The implementation demonstrates the effectiveness of the BLS signature scheme for key splitting and aggregation, and provides a solid foundation for further research and development in the field of blockchain and cryptocurrency security. The project contributes to the growing body of knowledge in decentralized storage and cryptocurrency ecosystems, and offers practical insights for developers and researchers working on similar projects. With the increasing adoption of blockchain technology and the need for secure and user-friendly solutions, the concepts and techniques explored in this project have the potential to drive innovation and shape the future of decentralized systems.

Appendix: Split Key Wallet Protocol Implementation Details

This appendix provides additional technical details and code snippets related to the implementation of the split key wallet protocol and mobile authenticator for the Züs platform.

BLS Signature Scheme Implementation

The BLS signature scheme was implemented using the Apache Milagro Cryptographic Library (AMCL). The following code snippet shows the key generation and signing functions:

```
// Generate BLS key pair
func GenerateKeyPair() (SecretKey, PublicKey) {
secret := amcl.NewRAND()
secretKey := secret.GetBIG()
publicKey := secret.G2mul(amcl.GeneratorG2(), secretKey)
return secretKey, publicKey
}
// BLS signing function
func Sign(message []byte, secretKey SecretKey) Signature {
hash := amcl.HashG1(message)
signature := amcl.G1mul(hash, secretKey)
return signature
}
```

Split Key Wallet Protocol Integration

The split key wallet protocol was integrated into the existing Züs wallet implementation. The following code snippet demonstrates the key splitting and partial signature generation:

```
// Split the private key into two components
func SplitKey(secretKey SecretKey) (SecretKey, SecretKey) {
sk1 := amcl.NewBIGcopy(secretKey)
sk2 := amcl.NewBIGcopy(secretKey)
sk1.Div(amcl.NewBIGint(2))
sk2.Sub(secretKey, sk1)
return sk1, sk2
}
// Generate partial signature on device 1
func SignPartial1(message []byte, sk1 SecretKey) PartialSignature {
hash := amcl.HashG1(message)
partialSig := amcl.G1mul(hash, sk1)
return partialSig
}
```

Mobile Authenticator Development

The mobile authenticator was developed using the Flutter framework. The following code snippet shows the transaction signing process:

```
// Combine partial signatures to generate the final signature
Future<Signature> combineSignatures(PartialSignature sig1, PartialSignature s
final combinedSig = ECPair.fromPrivate(sig1.add(sig2));
return Signature.fromBytes(combinedSig.signature);
}
// Sign transaction using the mobile authenticator
Future<void> signTransaction(Transaction transaction) async {
final partialSig1 = await getPartialSignature1(transaction);
final partialSig2 = await getPartialSignature2(transaction);
final combinedSignature = await combineSignatures(partialSig1, partialSig2);
transaction.signature = combinedSignature;
await broadcastTransaction(transaction);
}
```

Bibliography

[1] G. R. Blakley. Safeguarding cryptographic keys. In *1979 international workshop on managing requirements knowledge (MARK)*, pages 313–318. IEEE, 1979.

[2] D. Boneh, B. Lynn, and H. Shacham. Short signatures from the weil pairing. In *International conference on the theory and application of cryptology and information security*, pages 514–532. Springer, 2001.

[3] D. Boneh, C. Gentry, B. Lynn, and H. Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 416–432. Springer, 2003.

[4] Y. G. Desmedt. Threshold cryptography. *European transactions on telecommunications*, 5(4):449–458, 1988.

[5] L. Fan, H. Su, T. Liu, X. Chen, and G. Bai. A new hardware wallet scheme for cryptocurrencies. In *2019 IEEE International Conference on Blockchain (Blockchain)*, pages 386–392. IEEE, 2019.

[6] R. Gennaro, S. Goldfeder, and A. Narayanan. Threshold-optimal dsa/ecdsa signatures and an application to bitcoin wallet security. In *International Conference on Applied Cryptography and Network Security*, pages 156–174. Springer, 2016.

[7] G. Gutoski and D. Stebila. Hierarchical deterministic bitcoin wallets that tolerate key leakage. In *International Conference on Financial Cryptography and Data Security*, pages 497–504. Springer, 2015.

[8] D. Johnson, A. Menezes, and S. Vanstone. The elliptic curve digital signature algorithm (ecdsa). *International journal of information security*, 1(1):36–63, 2001.

[9] Y. Kaga, M. Fujio, K. Naganuma, K. Takahashi, T. Murakami, T. Ohki, and M. Nishigaki. A secure and usable offline backup system for cryptocurrency wallets. In *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, volume 2, pages 34–39. IEEE, 2017.

[10] K. Karantias, A. Kiayias, and D. Zindros. Proof-of-burn. In *International Conference on Financial Cryptography and Data Security*, pages 523–540. Springer, 2020.

[11] K.-C. Liao and W.-H. Lee. A novel user authentication scheme based on qr-code. *Journal of networks*, 5(8):937–941, 2010.

[12] D. M'Raihi, S. Machani, M. Pei, and J. Rydell. Totp: Time-based one-time password algorithm. Technical report, RFC 6238, May, 2011.

[13] A. Mühle, A. Grüner, T. Gayvoronskaya, and C. Meinel. A survey on essential components of a self-sovereign identity. *Computer Science Review*, 30:80–86, 2018.

[14] K. Reese, T. Smith, J. Dutson, J. Armknecht, J. Cameron, and K. Seamons. A usability study of five two-factor authentication methods. In *Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019)*, 2019.

[15] C.-P. Schnorr. Efficient signature generation by smart cards. *Journal of cryptology*, 4(3):161–174, 1991.

[16] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11): 612–613, 1979.

[17] V. Shoup. Practical threshold signatures. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 207–220. Springer, 2000.