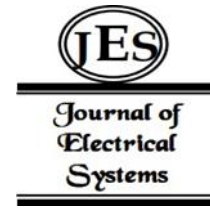


¹ Jongseok Choi² Hyejin Park

Recoverable Decentralized Wallet based on Social Service



Abstract: - The landscape of blockchain wallets has evolved significantly, offering various options such as non-custodial, self-custodial, and custodial solutions. While non-custodial wallets, typified by platforms like Metamask, initially dominated the scene, instances of users losing access to their assets due to misplaced wallet keys have underscored the need for more robust alternatives. Consequently, custodial wallets emerged, entrusting the management of private keys to centralized servers. However, this centralized approach poses vulnerabilities, particularly to insider attacks. In response to these challenges, self-custodial wallets have emerged as a middle ground, employing Multi-Party Computation (MPC) to manage users' private keys while empowering users to retain control over their restore keys. Nevertheless, the encryption of keys on MPC raises concerns regarding the irretrievability of wallets in the event of restore key loss. Both custodial and self-custodial solutions rely on designated entities, potentially leading to wallet inaccessibility or failure. To address these issues, we propose the concept of a non-custodial recoverable wallet. This protocol leverages OpenID and smart contracts to establish a foundational identity framework, with all entity-dependent information transparently documented in smart contracts. The proposed scheme unfolds in three distinct phases: setup, key generation, and key recovery. During the setup phase, users initialize their global identity via OpenID, establishing unique identities through respective providers. Service providers then configure secrets to facilitate domain-specific key generation for their users. Key generation involves the computation of domain-specific keys using the service provider's secret and the user's random number, with proof hints stored on the smart contract using homomorphic encryption. Additionally, this phase entails the creation of a recovery key stored in the user's private storage, associated with the global identity provider. Authentication and authorization of users are enabled during key generation through key computation and message signing. In cases of emergency, users can initiate the key recovery phase, facilitating the restoration of domain-specific keys while validating their correctness through proof hints on the smart contract. This approach offers three significant contributions. Firstly, it pioneers the development of non-custodial wallets integrated with social services such as OpenID. Secondly, it introduces fully server-independent key recovery mechanisms. Lastly, it establishes hierarchical identity structures, enabling global identities to retain control over domain-specific identities even in the event of domain-specific key loss. In conclusion, the recoverable non-custodial wallet, underpinned by social services, presents users with a convenient and familiar avenue for managing their keys. By leveraging trusted service providers, we aim to democratize access to blockchain services while enhancing security through the distribution of restore key fragments across multiple storage locations, akin to MPC principles.

Keywords: blockchain, social wallet, recovery, smart contract.

I. INTRODUCTION

With the growing adoption of blockchain technology, the demand for secure cryptocurrency wallet solutions for financial transactions has surged. Cryptocurrency wallets, which store the private keys necessary to access and manage cryptocurrencies, can be broadly categorized into two main types: custodial and non-custodial.

Non-custodial wallets grant users complete control over their private keys. In this model, users are solely responsible for managing their keys and ensuring their security. Conversely, custodial wallets operate by having a third-party service provider manage users' private keys on their behalf. While this approach may offer convenience, it introduces a level of reliance on the service provider for the security and accessibility of users' funds. To address this reliance issue of custodial wallets, self-custodial wallets have emerged as an alternative. The self-custodial wallets effectively prevent unauthorized access by service providers from controlling users' wallets. This approach requires users to provide their secrets to unlock their wallets even if their keys are stored in MPC managed by the service provider. Therefore, most self-custodial wallets lack users' secret loss and sunset of the service provider.

Especially, lots of researchers have presented attacks and risks on custodial wallets. [1] introduces a risk model for Revault, which is an open-source custody protocol. [2] identified potential security and privacy vulnerabilities in auditability protocol; weak cryptographic operations, lack of data binding and lack of user-ID uniqueness. [3]-[12] have pointed out the security on MPC and reliance issues.

¹ Biyard Corp. miner@biyard.co

² Biyard Corp. summer@biyard.co

This study addresses this challenge by proposing a novel approach to building a recoverable decentralized social wallet that leverages the power of Multi-Party Computation (MPC) and homomorphic encryption. By combining Web2 services and smart contract on Web3, the real MPC can be composed. To guarantee security of data on Web3, the MPC uses homomorphic encryption.

This paper introduces non-custodial wallet based on social services such as Google, Kakao, etc. This scheme has two contributions. First, it serves higher user-experienced wallet based on OAuth. Previously, users should remember their seed phases to generate the same key pairs, or they should depend on service providers such as Web3Auth which manages users' key pairs mapped by OAuth. The next contribution of this paper allows users to recover their key pairs without any single point of a provider.

II. PRELIMINARIES AND RELATED WORKS

2.1 Authentication Factors

Authentication proves an actor to have specified identification. Methodology to prove having identification can be classified into three factors: knowledge, possession and biometric. Knowledge-based authentication means proving if an actor knows their secret such as password. Possession-based authentication verify identification with physical devices. Mobile phones and OTP devices are popular possession-based authentication. Biometric-based authentication is using features to be a person such as fingerprinting, iris etc.

Multi-Factor Authentication (MFA) indicates authentication to require users to prove two or more factors of them [13]-[14].

2.2 Homomorphic Encryption

Homomorphic encryption allows users to perform additional encryption over encrypted data [15]. There are five types of homomorphic encryption. Pre-Fully Homomorphic Encryption (FHE) performs homomorphic encryption using RSA, ElGamal, etc. [16-19]. First-generation FHEs [20]-[21] uses lattice-based cryptography which is much more secure than Pre-FHE. Second-generation FHEs are based on Ring Learning With Errors (RLWE) which starts from [20]. Third-generation FHE features slower growth rate of noise based on [23]. Fourth-generation FHE [24] enhances rescaling operation.

This section only describes the basic concept of homomorphic encryption. Let m_n , E and e be a n-th message, encryption algorithm and an encryption key, respectively.

$$E(m_1)E(m_2) = E(m_1 \cdot m_2)$$

$$m_1^e m_2^e = (m_1 m_2)^e$$

2.3 Secure MPC

Secure Multi-Party Computation (MPC) aims to compute a function by joining multi-party. It indicates no one having full information for a function of a task [25]. Assume full private data, P , and a public homomorphic function, F . A given number of participants, p_1, p_2, \dots, p_N . Each participant has a piece of private data, d_1, d_2, \dots, d_N .

$$P = F(d_1) \cdot F(d_2) \cdots F(d_N)$$

$$= F(d_1, d_2, \dots, d_N)$$

III. PROPOSED SCHEME

Self-custodial and custodial wallets may make users unavailable to transact by their wallet if the service providers shut down their MPC service because MPC demands on MPC hardware. To overcome this issue, each computation resource composing MPC must be operated by other entities. In other words, it needs to employ decentralized MPC which is implemented over decentralized infrastructure.

This section presents a decentralized social wallet which can be recovered by Web2 services and smart contracts. Basically, it generates a wallet by MPC consisting of Web2 and smart contracts. Web2 provides user-private storage smart contracts reliable storage. However, the smart contracts officially open all personal data used to calculate key pairs. To prevent unexpected reveals, this scheme uses homomorphic encryption. In this scheme, there are two

types of identity; global identity and domain-specific identity. All users have a globally unique identity as a master wallet. The global identity generates sub keys, called domain-specific identity. Instead of global identity, Users ease domain-specific identity on each service.

This scheme consists of three phases; Setup, key generation and key recovery. Setup generates users' global identities. This phase may derive a global identity from a unique identity of OpenID. Also, it utilizes secure storage such as Google's app storage to store the global identity. Key generation registers users' domain-specific identities to a smart contract which is dedicated to a service. Key recovery restores domain-specific identities with a secure storage and smart contract.

3.1 Setup

This phase generates users' global identity once each user hopes to use this scheme. This phase depends on secure storage which only the owned user can access. Representative secure storage is Google's app storage which allows the only authorized users to grant access. Specifically, a global identity consists of a pair of a private key and a public key derived from a random seed.

This phase generates the key pair over BIP32. However, other key generation algorithms can be used as an alternative to BIP32. Let G , H and r be key generator such as BIP32, a hash function as SHA3 and a random number.

$$(priv, pub) = G(H(r))$$

3.2 Key generation

Key generation phase is used whenever a user signs in a service. This phase aims to generate a domain-specific identity. The difference between the global identity and a domain-specific identity is that global identity totally depends on a user, but the domain-specific identity fairly depends on the service provider and a user.

The basic concept of fair contribution means a user and a service provider submits a password and a random number, respectively.

Let F be a homomorphic encryption based on global identity. A user sets a password for domain-specific authentication. Let $H(PW)$ be a user contribution where H and PW are a cryptographic hash function and the password, respectively. After hence, the user stores the $F(P)$ in their secure storage. For this, each domain service might need to redirect to user's domain set by Setup phase.

$$F(P) = F(H(PW))$$

A service provider generates a random number, s , which is dedicated to a user. The service provider stores $F(s)$ to a well-known smart contract.

$$F(P) \cdot F(s) = F(H(PW) \cdot s)$$

Then, authorized users can decrypt F and generate a key pair. In this phase, the user cannot know s even if decrypt F function. Only they can know the value of $H(PW) \cdot s$. More specifically, it might form as $g^{H(PW) \cdot s}$. Let F' be a decrypt function of F . In the case of RSA, it can be inverse multiplication over Ring.

$$\begin{aligned} & BIP32(H(H(PW) \cdot s)) \\ &= BIP32\left(H\left(F'\left(F(H(PW) \cdot s)\right)\right)\right) \end{aligned}$$

Typically, a user generates their key pair by interacting with a service and their password. Note that the service provider stores the secret or uses reproducible random number such as derivation from their secret and user's identity.

3.3 Key recovery

This phase restores a domain-specific identity using a global key identity. Key generation phase stored $F(H(PW) \cdot s)$ into a smart contract which is immutable. And F is a homomorphic encryption based on a user

private key. It means that the only authorized user can decrypt it with F' . To restore user's key pair where the user lost their password, the key pair can be re-generated by F' and $F(H(PW) \cdot s)$ on smart contract. This phase extracts F' from user's private storage used in Setup phase. After that, the user can calculate $H(PW) \cdot s$ by the equation below.

$$F'(F(H(PW) \cdot s)) = H(PW) \cdot s$$

Key recovery phase is based on security over two parties; a private storage and a smart contract. Note that the information on a smart contract is encrypted and secure based on hardness of homomorphic encryption.

IV. COMPARISONS

This section compares functionalities of four types of wallets; non-custodial, custodial, self-custodial wallets and proposed scheme. First the non-custodial wallet allows users to fully control over their wallet, but it is vulnerable against loss of seed phrase. In the case of custodial wallets, users do not consider losing their seed phrase to restore their wallet. However, it can forcibly lock up users' wallet because key pairs are managed by service providers. Additionally, custodial wallets can be abused by insiders. To prevent insider attacks, self-custodial wallets have been employed. The self-custodial wallets utilize MPC to store private data and compute by multi parties. It prevents insiders from abusing users' keys. The MPC requires users to provide their secrets to sign transactions. Therefore, insiders, who do not know users' secret, cannot abuse users' wallets. Additionally, it can manage users' keys based on OAuth. It is still vulnerable to losing users' secret themselves and shutting down services. To overcome this vulnerability, the proposed scheme combines non-custodial wallet with MPC. It can restore their key independent to service providers.

	Non-custody	Custody	Self-custody	Proposed
OAuth	X	O	O	O
Recovery	O	O	O	O
Insider attack resistance	O	X	O	O
Fully controllable	O	X	X	O

V. CONCLUSION

5.1 Key Contributions

The proposed scheme allows users to generate their wallets by OAuth as self-custodial and custodial wallets. The proposed scheme resistant insider attack and provides users with fully controllable wallets.

5.2 Discussion

Currently, the proposed scheme comprises MPC of Web2 private storage and a smart contract. And it needs all of them to recover a key from the smart contract. Therefore, we need to research how to distribute global identity and apply threshold cryptography.

5.3 Future works

We need to research threshold cryptography using multi-party Web2 storages and MPC on smart contracts which only response and have consensus with same data.

VI. ACKNOWLEDGEMENT

This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. RS-2024-00397538, Development of public opinion polling technology based on web3 that ensures fairness, anonymity, and transparency, 100%)

REFERENCES

- [1] J. Swambo, A. Poinot. "Risk framework for bitcoin custody operation with the revault protocol." In Financial Cryptography and Data Security. FC 2021 International Workshops: CoDecFin, DeFi, VOTING, and WTSC, Virtual Event, March 5, 2021, Revised Selected Papers 25, pp. 3-20. Springer Berlin Heidelberg, 2021.

- [2] K. Chalkias, P. Chatzigiannis, Y. Ji. "Broken proofs of solvency in blockchain custodial wallets and exchanges." In International Conference on Financial Cryptography and Data Security, pp. 106-117. Cham: Springer International Publishing, 2022.
- [3] M. Rauchs, A. Blandin, K. Klein, G. C. Pieters, M. Recanatini, B. Z. Zhang. "2nd global cryptoasset benchmarking study." Available at SSRN 3306125 (2018).
- [4] A. Blandin, G. C. Pieters, Y. Wu, A. Dek, T. Eisermann, D. Njoki, and S. Taylor. "3rd global cryptoasset benchmarking study." Available at SSRN 3700822 (2020).
- [5] M. Conti, E. S. Kumar, C. Lal, S. Ruj. "A survey on security and privacy issues of bitcoin." IEEE communications surveys & tutorials 20, no. 4 (2018): 3416-3452.
- [6] A. Feder, N. Gandal, J. T. Hamrick, T. Moore. "The impact of DDoS and other security shocks on Bitcoin currency exchanges: Evidence from Mt. Gox." Journal of Cybersecurity 3, no. 2 (2017): 137-144.
- [7] M. Vasek, M. Thornton, T. Moore. "Empirical analysis of denial-of-service attacks in the bitcoin ecosystem." In Financial Cryptography and Data Security: FC 2014 Workshops, BITCOIN and WAHC 2014, Christ Church, Barbados, March 7, 2014, Revised Selected Papers 18, pp. 57-71. Springer Berlin Heidelberg, 2014.
- [8] T. Moore, N. Christin. "Beware the middleman: Empirical analysis of Bitcoin-exchange risk." In Financial Cryptography and Data Security: 17th International Conference, FC 2013, Okinawa, Japan, April 1-5, 2013, Revised Selected Papers 17, pp. 25-33. Springer Berlin Heidelberg, 2013.
- [9] T. Moore, N. Christin, J. Szurdi. "Revisiting the risks of bitcoin currency exchange closure." ACM Transactions on Internet Technology (TOIT) 18, no. 4 (2018): 1-18.
- [10] G. Dagher, B. Büinz, J. Bonneau, J. Clark, D. Boneh. "Provisions: Privacy-preserving proofs of solvency for bitcoin exchanges." In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pp. 720-731. 2015.
- [11] T. Bamert, C. Decker, R. Wattenhofer, S. Welten. "Bluewallet: The secure bitcoin wallet." In Security and Trust Management: 10th International Workshop, STM 2014, Wroclaw, Poland, September 10-11, 2014. Proceedings 10, pp. 65-80. Springer International Publishing, 2014.
- [12] R. Gennaro, S. Goldfeder, A. Narayanan. "Threshold-optimal DSA/ECDSA signatures and an application to bitcoin wallet security." In Applied Cryptography and Network Security: 14th International Conference, ACNS 2016, Guildford, UK, June 19-22, 2016. Proceedings 14, pp. 156-174. Springer International Publishing, 2016.
- [13] A. Ometov, S. Bezzateev, N. Mäkitalo, S. Andreev, T. Mikkonen, Y. Koucheryavy. "Multi-factor authentication: A survey." Cryptography 2, no. 1 (2018): 1.
- [14] A. S. AlQahtani, Z. El-Awadi, M. Min. "A survey on user authentication factors." In 2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), pp. 0323-0328. IEEE, 2021.
- [15] F. Armknecht, C. Boyd, C. Carr, K. Gjøsteen, A. Jäschke, Christian A. Reuter, and M. Strand. "A guide to fully homomorphic encryption." Cryptology ePrint Archive (2015).
- [16] R. L. Rivest, L. Adleman, M. L. Dertouzos. "On data banks and privacy homomorphisms." Foundations of secure computation 4, no. 11 (1978): 169-180.
- [17] T. Sander, A. Young, M. Yung. "Non-interactive cryptocomputing for $nc/sup 1$." In 40th Annual Symposium on Foundations of Computer Science (Cat. No. 99CB37039), pp. 554-566. IEEE, 1999.
- [18] D. Boneh, E. Goh, K. Nissim. "Evaluating 2-DNF formulas on ciphertexts." In Theory of Cryptography: Second Theory of Cryptography Conference, TCC 2005, Cambridge, MA, USA, February 10-12, 2005. Proceedings 2, pp. 325-341. Springer Berlin Heidelberg, 2005.
- [19] Y. Ishai, A. Paskin. "Evaluating branching programs on encrypted data." In Theory of Cryptography Conference, pp. 575-594. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007.
- [20] C. Gentry. "Fully homomorphic encryption using ideal lattices." In Proceedings of the forty-first annual ACM symposium on Theory of computing, pp. 169-178. 2009.
- [21] C. Gentry, S. Halevi. "Implementing gentry's fully-homomorphic encryption scheme." In Annual international conference on the theory and applications of cryptographic techniques, pp. 129-148. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.
- [22] Z. Brakerski, C. Gentry, V. Vaikuntanathan. "(Leveled) fully homomorphic encryption without bootstrapping." ACM Transactions on Computation Theory (TOCT) 6, no. 3 (2014): 1-36.
- [23] C. Gentry, A. Sahai, B. Waters. "Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based." In Advances in Cryptology—CRYPTO 2013: 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I, pp. 75-92. Springer Berlin Heidelberg, 2013.
- [24] J. H. Cheon, A. Kim, M. Kim, Y. Song. "Homomorphic encryption for arithmetic of approximate numbers." In Advances in Cryptology—ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I 23, pp. 409-437. Springer International Publishing, 2017.
- [25] M. Yung. "From mental poker to core business: Why and how to deploy secure computation protocols?." In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pp. 1-2. 2015.