



Article

Reminisce: Blockchain Private Key Generation and Recovery Using Distinctive Pictures-Based Personal Memory

Jungwon Seo ¹, Deokyoon Ko ², Suntae Kim ³, Vijayan Sugumaran ^{4,5} and Sooyong Park ^{6,*}

¹ Department of Computer Science and Engineering, Sogang University, 915 Ricci Hall 35, Baekbeom-ro, Mapo-gu, Seoul 04107, Korea; jungwon@sogang.ac.kr

² Nonce Lab Inc., 409 Seoul Startup Hub., 14 Magokjungang 8-ro, Gangseo-gu, Seoul 07801, Korea; dykoh@noncelab.com

³ Department of Software Engineering, Jeonbuk National University, 67 Baekje-daero, Deokin-gu, Jeonju-si 54896, Korea; stkim@jbnu.ac.kr

⁴ Department of Decision and Information Sciences, School of Business Administration, Oakland University, Rochester, MI 48309, USA; sugumara@oakland.edu

⁵ Center for Data Science and Big Data Analytics, Oakland University, Rochester, MI 48309, USA

⁶ Department of Computer Science and Engineering, Sogang University, 915A Ricci Hall 35, Baekbeom-ro, Mapo-gu, Seoul 04107, Korea

* Correspondence: sypark@sogang.ac.kr

Abstract: As a future game-changer in various industries, cryptocurrency is attracting people's attention. Cryptocurrency is issued on blockchain and managed through a blockchain wallet application. The blockchain wallet manages user's digital assets and authenticates a blockchain user by checking the possession of a user's private key. The mnemonic code technique represents the most widely used method of generating and recovering a private key in blockchain wallet applications. However, the mnemonic code technique does not consider usability to generate and recover a user's private key. In this study, we propose a novel approach for private key generation and recovery. Our approach is based on the idea that a user can hold long-term memory from distinctive pictures. The user can generate a private key by providing pictures and the location of the pictures. For recovering a private key, the user identifies the locations of the pictures that are used in the private key generation process. In this paper, we experiment with the security and usability of our approach and confirm that our proposed approach is sufficiently secure compared to the mnemonic code technique and accounts for usability.

Keywords: blockchain wallet; blockchain; blockchain private key

MSC: 68U99



Citation: Seo, J.; Ko, D.; Kim, S.; Sugumaran, V.; Park, S. Reminisce: Blockchain Private Key Generation and Recovery Using Distinctive Pictures-Based Personal Memory. *Mathematics* **2022**, *10*, 2047. <https://doi.org/10.3390/math10122047>

Academic Editor: Jan Lansky

Received: 3 May 2022

Accepted: 10 June 2022

Published: 13 June 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Blockchain is expected to be applicable in various industries such as energy, health care, and finance [1–7]. As expected, various blockchain business models are appearing today, especially cryptocurrency business models which are currently attracting the attention of many. The first form of cryptocurrency started with Bitcoin and was initially not popular due to the malicious use of cryptocurrency [8], ICO (Initial Coin Offering) fraud [9,10], and volatility of values. However, recently, through DeFi (Decentralized Finance), CBDC (Central Bank Digital Currencies), and NFT (Non-Fungible Token), the value of cryptocurrency has been re-evaluated by interested individuals and entities. The interest in cryptocurrency can be observed through cryptocurrency exchanges such as CoinMarketCap [11]. In addition, various studies related to cryptocurrency are currently in progress [12–17].

In order to bring true changes beyond expectations, developers must not only overcome the technical limitations of blockchain; but also need to comprehensively analyze the

blockchain wallet. The blockchain wallet is an application that bridges the gap between blockchain networks and the real world. The blockchain wallet helps a user access personal digital assets in blockchain networks. For example, a user can send cryptocurrency to others using a blockchain wallet. Moreover, the blockchain wallet authenticates a blockchain user by checking possession of a private key. The private key of the blockchain is an object that identifies the user without additional authentication from other institutions [18] and the private key can be generated and recovered through the blockchain wallet.

Mnemonic codes are the most commonly used technique to generate and recover a private key in the blockchain wallet. The mnemonic code technique used in blockchain uses the word list in BIP-0039 [19]. A private key is generated by combining 12 to 24 words out of 2048 words of the BIP-0039 list as a seed. Despite the widespread use of blockchain key generation and recovery using the mnemonic code technique, the technique is inefficient in private key generation and recovery. For example, when a user tries to generate a private key, they are faced with the inconvenience of finding a word list that can be used as a mnemonic code or the user needs to employ a mnemonic code generator on the Internet. Furthermore, if a user tries to recover a private key but cannot recall or locate the mnemonic code, recovery becomes impossible. A 2017 survey found that four-million Bitcoins were inaccessible due to the user's loss of a private key [20]. In addition, a company went bankrupt after it failed to recover its lost private key [21]. With these mentioned examples, accidents involving mnemonic code recovery failure continue to occur, suggesting that the mnemonic code technique does not consider usability in private key generation and recovery for users. Therefore, we propose a novel approach considering usability and security to improve the current private key generation and recovery process.

Various studies [22–25] to utilize a blockchain wallet are being conducted, as well as academic studies [26–32] to improve the current private key generation and recovery process. The most common studies [26–29] allowed other users or external repositories to participate in the process of generating and recovering a private key. These studies suggested storing core information for recovering a private key to external users or external repositories. These studies ensured private key recovery by storing core information to other users and external repositories during a private key generation process. However, there is a limitation that if external users or repositories with core information are attacked, there is a high risk that a private key is recovered by another user. In other studies [30,31], authors suggested using unique biometric information such as fingerprints to generate and recover private keys. In these studies, the safety and usability of generating and recovering a private key were ensured by using biometric information possessed only by the user. However, there is the limitation of requiring special devices to collect biometric information. Another study [32] suggested that a user includes information for recovering a private key when generating the private key. In this study, the safety and usability of private key generation and recovery were ensured by utilizing information generated by a user. However, the results were limited in effectiveness as it did not improve significantly from the mnemonic code technique.

In this paper, we propose a novel method to generate and recover a private key via a user's recall of natural memories. Unlike the mnemonic code technique, which assigns words for the user to memorize for private key generation and recovery, our approach is novel in that it utilizes a user's long-term memories to generate and recover a private key. In our approach, a user provides a specific number of pictures that can evoke natural memories. After that, the user inputs the location of each provided picture, and a user's private key is generated based on the location provided by the user. When a user needs to recover the private key, they recall the location of the pictures they selected during the initial private key generation process. In addition, experiments are conducted on the basis of our approach and they show that our approach is sufficiently secure compared to the mnemonic code technique and takes into account usability. The contributions of this paper are as follows:

- We propose a new approach that is based on a user's long-term memory using distinctive pictures for generating and recovering a private key.
- We conduct various experiments with at least 20 to 105 participants to assess the usability and security of the proposed approach.
- We develop a real-world mobile wallet application to identify the development possibilities and feasibility of the proposed approach.

The remainder of this paper is organized as follows. Section 2 presents background knowledge of blockchain and related work for private key generation and recovery. Section 3 explains our approach and Section 4 presents the experiments conducted to demonstrate the efficacy of our approach. Section 5 concludes the paper and discusses future work.

2. Background and Related Work

2.1. Background

This section presents background knowledge related to blockchain and cryptocurrency (Section 2.1.1), blockchain wallet and cryptographic key (Section 2.1.2) to understand our approach.

2.1.1. Blockchain and Cryptocurrency

Blockchain was started by fundamental academic studies from Stuart Haber [33], David Chaum [34], and Dave Bayer [35]. Blockchain is a data storage technology that allows all users to share and store all the data in the blockchain network. Because users share and store all data, the blockchain maintains the integrity and transparency of data. Data are validated by a consensus algorithm and they are stored in one block. Each block is connected back and forth and stored in the network, thus being called a blockchain. Blockchain technology consists largely of P2P (Peer-to-Peer) network, cryptography, and consensus algorithm.

Unlike the server–client network structure where servers are responsible for processing the broadcasting data and clients only read data and request to servers, in the P2P network each participating system acts as a server and client at the same time. The role of the P2P network in a blockchain ensures all participants can store data in a distributed manner. Furthermore, blockchain uses cryptographic techniques with a hashing algorithm and asymmetric key technology. By these cryptographic techniques, blockchain guarantees and verifies the integrity of data stored on a blockchain. Furthermore, a consensus algorithm ensures that all participants in a blockchain can store the same data. In a blockchain, cryptocurrency plays a role in inducing blockchain users to participate in a consensus algorithm. Blockchain users can acquire cryptocurrency as a reward to participate in a consensus algorithm or users can purchase cryptocurrency using traditional currency from cryptocurrency exchanges.

2.1.2. Blockchain Wallet and Cryptographic Key

An individual can use a blockchain wallet application after verifying ownership of the blockchain wallet by a simple mechanism such as a password. Furthermore, the blockchain wallet helps to connect a blockchain network by verifying possession of a private key from a blockchain user. Blockchain wallets can be classified as hot wallets and cold wallets. Hot wallets are a form of user's computer application or web extension program such as the Meta Mask [36]. Hot wallets have the advantage of being user-friendly because it is always connected to the Internet. However, there is a risk that cryptocurrency exploitation through network attacks can occur. Cold wallets refer to the storage of wallets on other physical devices separated from the user's computer such as a USB. Cold wallets can be secured from a hacker's network attack because it is separated from the user's computer, but has the disadvantage of being difficult to manage.

In cryptography, keys can typically be divided into symmetric-key algorithms that can be encrypted and decrypted with one key or asymmetric-key algorithms that require

encryption and decryption to proceed with another key. Blockchain uses an asymmetric-key algorithm with ECDSA (Elliptic Curve Digital Signature Algorithm) [37] which consists of a private key and a public key. In a blockchain network, a private key is a unique random number that does not overlap with other users. It should be kept safe and not disclosed to others. A private key is used to encrypt transactions with other users. The public key can be disclosed to others and is used to decrypt transactions that are encrypted by a private key.

2.2. Related Work

This section includes existing studies [26–32] on blockchain private key generation and recovery and these studies can be divided into three main categories.

- Studies that store core information related to a private key in other users' or external repositories at the time of the private key generation process and utilizes it for recovery [26–29].
- Studies to generate and recover a private key using user biometric information [30,31].
- A study that includes information for the recovery of a private key by users when they create private key [32].

Soltani and Nguyen et al. [26] suggested generating and recovering a private key by using KEP (Key Escrow Providers). In this study, KEP generates a main private key, and multiple sub-private keys and then provides them to a user. The main private key and sub-private keys have the same public key and the user mainly uses only the main private key. When the user loses the main private key, the user provides their assigned sub-private keys to KEP. The KEP recovers the main private key by applying the Lagrange Polynomial. Zhu and Chen et al. also proposed a study called HA-eWallet [27] which provides private key recovery information to an external repository. According their study, a user generates multiple private keys at the time of the private key generation and stores all private keys in DRC (Disaster Recovery Center). Among the stored private keys, only one private key that has been user-selected is used primarily. During the private key recovery process, DRC recovers a lost private key to the user through ownership authentication for the other known keys then the lost ones.

He and Wu et al. [28] proposed an approach that combines specific seeds to generate a private key and transfers each seed to friends of a user in a blockchain network for enabling private key recovery. In this study, when a private key recovery is requested, the user's friends provide seeds to the user, and based on this, the user can recover the private key. Another study by He and Lin et al. [29] focused on private key generation and recovery using the DMCD (Dependent Multi-Constrained Derangement) and SKN (Shamir-Kademlia-Neighbor) methods. When a user generates a private key, they create additional private key fragments together. The user sends the fragments to other specific users who are selected by DMCD. During the private key recovery process, a user receives private key fragments from other users. The user verifies private key fragments by SKN and proceeds to recover the private key.

These studies [26–29] ensure private key recovery by storing core information to other users and external repositories during a private key generation process. However, our approach does not involve other users in a private key generation process to ensure the private key recovery. Furthermore, in our approach, an external repository stores only limited information that is not directly related to a private key, unlike previous research.

Another type of study utilizes biometric information from a user to generate and recover a private key. Aydar and Cetin et al. [30] utilized fingerprints to generate and recover a private key. To ensure the security of user biometric information, the researchers use the Reed–Solomon technique [38]. Zhao and Zhang et al. [31] proposed the use of BSN (Body Sensor Network) and Fuzzy Vault [39] techniques for generating and recovering a private key in the healthcare blockchain. In this study, a user utilizes IoT devices to build BSN to obtain user biometric information. When a user requests the recovery of a private key, the user can obtain their biometric information via BSN. After that, the user can recover the private key using the biometric information authenticated by Fuzzy Vault.

These studies [30,31] ensure the generation and recovery of a private key using the body characteristics of a user. However, there is a limitation of requiring devices such as fingerprint recognizers to collect biometric data from a user. Our approach can provide better usability than previous methods as a user needs only their cell phone.

Signth and Stefanidis et al. [32] proposed PKRS (Partial Knowledge Recovery Scheme) for generating and recovering a user’s private key. Their research proposal is similar to the mnemonic code technique used in traditional wallet applications. In wallet applications that use a mnemonic code technique, the user is required to remember multiple words to recover their private key, while Signth’s study requires a user to remember answers to the user’s own questions. When a user generates a private key, the user splits the private key to create private key fragments. Each of those private key fragments is created in a form that includes specific user-generated questions and answers. When a user requests private key recovery, they must answer questions that have been stored in private key fragments to activate and combine fragments to recover the private key.

The approach of Signth et al. is similar to our approach in that a private key can be recovered via memory recall of the user. However, our approach uses pictures for memory recall instead of the word-based recall technique proposed by Signth et al.

Table 1 shows comparisons of each related work and the proposed approach. The first column indicates each related work and proposed approach. The second column indicates the core factors of the private key generation and recovery process of each study. The third column shows the possible attack scenario by core factors. Furthermore, the last column indicates the limitations of the approach.

For example, in Singth’s research, a user’s private key is generated and recovered by the user’s memory, to be precise, by the user’s generated query and response. This core factor is opens a vulnerability whereby the private key can be stolen by someone who knows all the answers to queries and responses. Furthermore, Singth’s research has limitations in that the approach is dependent on the short-term memory of users.

Table 1. Comparison of related work.

Related Work	Core Factor	Possible Attack	Limitation
Soltani and Nguyen et al. [26] and Zhu and Chen et al. [27]	External repository	Repository operators can steal a private key	Third-party dependency
He and Wu et al. [28] and He and Lin et al. [29]	External users	External users can steal a private key	Third-party dependency
Aydar and Cetin et al. [30] and Zhao and Zhang et al. [31]	Biometric information	Stealing a private key by copying and stealing biometric information	Specific equipment dependency
Signth and Stefanidis et al. [32]	User’s memory	Anyone who knows all the answers to queries and responses	Short-term memory dependency
Proposed Approach	User’s memory	Anyone who knows location the picture was taken	Long-term memory dependency

In the proposed approach, the core factor is the user’s memory of a particular location in a picture. In addition, there is possibility that the private key may be stolen by someone who knows where the picture was taken, and the approach is limited by reliance on the user’s long-term memory.

3. Approach

We propose to generate and recover a private key using the distinctive picture-based personal memory for the blockchain wallet. Our Reminisce technique does not store core information that allows a private key to be directly recovered. The Reminisce technique also does not require special devices to generate and recover a private key and does not force the user to remember specific things. Our approach, Reminisce, is based on a phenomenon from neuronal science and cognitive psychology: (1) Long-term memory is held in a particular section of the brain, (2) special and emotional memory becomes long-term memory by hormone secretion, and (3) the picture superiority effect.

The case study of HM (Henry Molaison) [40] is a famous case study for discovery of long-term memory being held in a particular section of the brain. Henry Molaison's memory of the 11-year period before his surgery disappeared due to the surgeons removing the hippocampus. Based on this case, neuronal scientists discovered that short-term and long-term memories are classified in the human brain. Furthermore, a neuronal science study [41] identified sections of the brain associated with long-term memory and short-term memory. Moreover, other neuronal science research works [42,43] assert that stimulating memory such as special or emotional memory is considered long-term memory via hormone secretion. In other words, although general long-term memory is created by repeating short-term memory learning, special or emotional memory is stored directly in a specific part of the human brain by hormones as long-term memory. In addition, according to Paivio and Caspo's research [44] on cognitive psychology, a person can recall picture-based memories easier than simple letters or words such as an alphabet, and this phenomenon is called the picture superiority effect. Based on this research, other cognitive psychology studies [45–47] related to the picture superiority effect have been conducted.

The Reminisce uses pictures from the user to generate and recover a private key since the user's distinctive pictures contain the special or emotional memory unique to that individual. Thus, the user can recall the information from pictures stored in their long-term memory. In addition, in the proposed approach, the pictures are specific objects that can be easily recalled by the user. When taking a picture with a cell phone, various metadata such as date, size, and location are stored in the picture. Among various metadata of a picture, the Reminisce technique uses location coordinates to generate and recover the private key.

3.1. Module Design

Before explaining the Reminisce technique process, it is worth mentioning the module design to explain how the Reminisce technique executes within a wallet application. The Reminisce technique is a part of a blockchain wallet application and it can be described in Figure 1. The general blockchain wallet application can be largely divided into two main systems: cryptocurrency storage system and key management system. Each system is a basic system that should be included to perform a blockchain wallet application.

The cryptocurrency storage system is responsible for storing cryptocurrency in the wallet by connecting with blockchain platforms. The key management system helps a user generate and recover a private key. The Reminisce technique is applied as a part of the key management system. Furthermore, as shown in Figure 1, the key management system of which the Reminisce technique consists of include: (a) private key generation component, and (b) private key recovery component. The key generation component consists of the following modules: (a) picture collection, (b) GPS extraction, (c) location query, (d) picture validation and (e) key generation. Similarly, the key recovery component consists of the following modules: (a) picture load, (b) memory query, (c) temporary key generation, and (d) key validation. The role of each module is briefly described below.

The picture collection module is responsible for collecting pictures from the user. The module checks that the user's pictures contain location metadata and then interacts with the GPS extraction module. The GPS extraction module obtains GPS data from the metadata of the pictures and refines the GPS values. The location query module has a similar role to the GPS extraction module. The location query module obtains GPS data

from the user’s answers and also refines the GPS values. The GPS extraction module and the location query module interact with the picture validation module. The picture validation module compares GPS data from the GPS extraction module and the location query module. The picture validation module checks whether the difference between the GPS extraction module and the location query module is below a predefined value. Moreover, the picture validation module deletes metadata of pictures. The key generation module generates a private key through GPS data that is from the picture validation module. After this, the key generation module sends a private key and a public key to the user. Furthermore, the module stores the metadata-deleted pictures and public key in a repository.

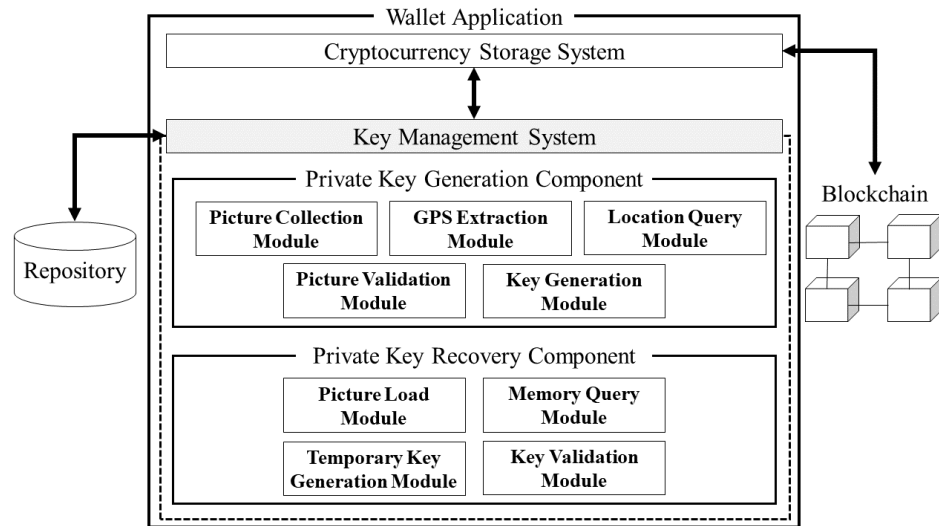


Figure 1. Module Design.

During the key recovery process, the four modules are executed. The picture load module brings an original public key and metadata-deleted pictures from the repository that matches the user who requested the private key recovery. The memory query module shows each of the pictures to a user and asks where the pictures are located. Based on the user’s answer, the temporary key generation module generates a temporary private key and a temporary public key. After this, the key validation module compares the temporary public key and the original public key. If the temporary public key and the original public key are the same, the key validation module determines that the private key recovery is successful.

3.2. Process Overview

This section explains the private key generation and recovery process. Furthermore, the Reminisce technique process can be described in Figure 2. As shown in Figure 2, the Reminisce technique is composed of two processes: private key generation process (Left part of Figure 2), and private key recovery process (Right part of Figure 2).

In the private key generation process, a user selects a picture whose location they can remember. Furthermore, the user provides the location of the provided picture. In step A-1, the provided picture is checked to see whether it contains metadata. If the picture contains metadata, the GPS data (m_gps) of the picture are extracted in step A-2. The user’s answer of the location (u_answer) is applied to a specific map application in step B-1. Based on u_answer , GPS data (u_gps) are obtained from a map application in step B-2. After this, m_gps and u_gps are represented up to α decimal places in step C.

In step D, m_gps and u_gps are compared to verify the location. If differences of m_gps and u_gps are greater than a specific δ value, the user has to select another picture. If the difference is less than or equal to the value of δ , the process is repeated until N pictures ($N(pic)$) are collected. When $N(pic)$ is collected, a seed is generated by multiple u_gps , and based on the seed, a private key is created in step E-1. Finally, the original private key

(o_sk), and original public key (o_pk) are generated and provided to the user. Moreover, the metadata for all pictures is deleted, and the metadata deleted pictures and o_pk are stored in a repository.

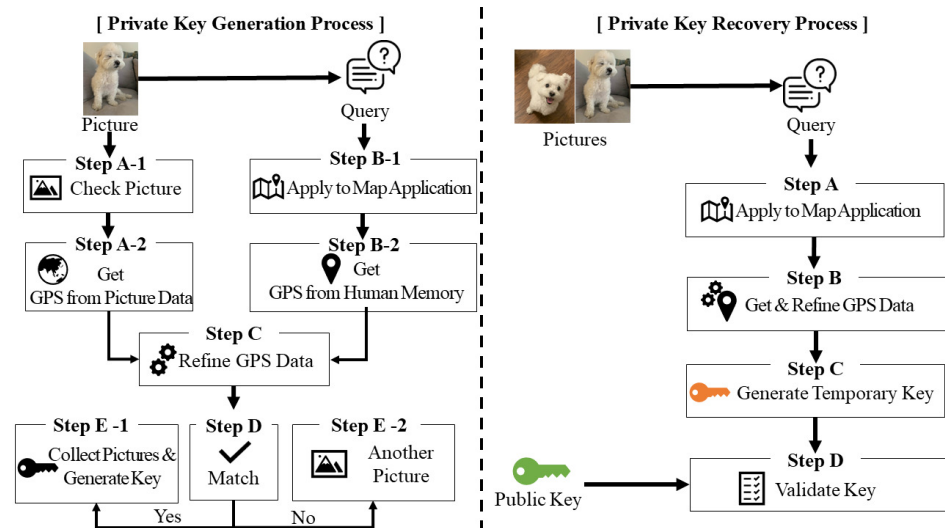


Figure 2. Process Overview.

In the private key recovery process, based on u_answer , a temporary private key (t_sk) is generated through steps A to C and this process is the same as steps B-1 to C of the private key generation process. After this, a temporary public key (t_pk) is generated by t_sk and t_pk is compared with o_pk . If t_pk and o_pk are identical, the key recovery process is successful, and o_sk that is the same as t_sk is provided to the user. The next sections describe the Reminisce process with the module design in more detail.

3.3. Private Key Generation Process

This section explains the private key generation process. The private key generation process works through a picture collection module, GPS extraction module, location query module, picture validation module, and key generation module. The private key generation process is represented in Figure 3.

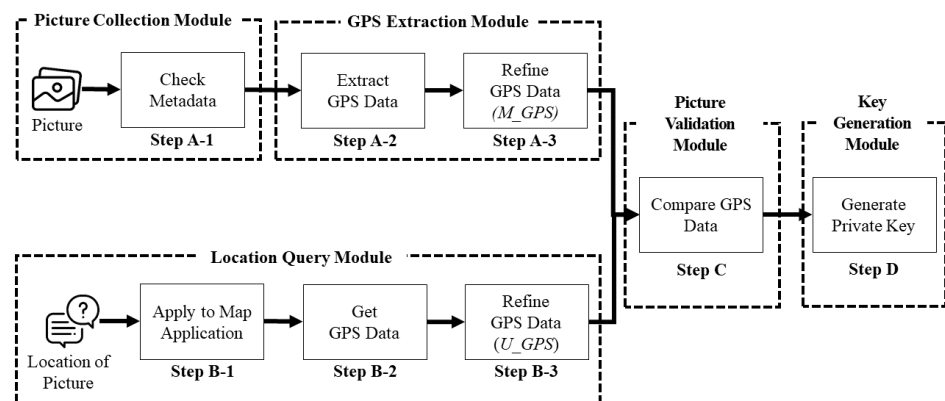


Figure 3. Private Key Generation Process.

The picture collection module collects a user’s picture. If GPS is recorded in the metadata of the received picture (Step A-1), the module sends the picture to the GPS extraction module. If GPS is not recorded in the metadata of the received picture, the process is terminated or the module asks the user to select another picture.

After the picture collection module sends the user’s picture, the GPS extraction module extracts GPS data from the picture. The GPS extraction module obtains GPS data (m_gps)

expressed in latitude (*lat*) and longitude (*lon*) directly from the picture's location metadata (Step A-2). Unlike the GPS extraction module, the location query module should work in conjunction with a user and a specific map application for obtaining GPS data. The module asks the user about the location of the picture and obtains the user's answer (*u_answer*). After this, the module applies *u_answer* to a specific map application (Step B-1) to obtain GPS data that is *u_gps* (Step B-2).

After obtaining GPS data in both modules which are the GPS extraction module and the location query module, they separate the *lat* and *lon* values of *m_gps* and *u_gps*. Next, each separated *lat* and *lon* value is expressed to α decimal places (Step A-3, B-3). The reason to express GPS data of *m_gps* and *u_gps* to α decimal places is to facilitate the comparison of each GPS data.

In the case of *m_gps*, the location of the GPS recorded when taking a picture may record a place that is different from the user's memory due to various factors such as device performance, GPS transmission location, or shadow fading. Furthermore, the *u_gps* value obtained by *u_answer* is extremely difficult to maintain the same level of accuracy as the *m_gps* recorded in the picture. For example, Google Maps which is a popular map application expresses GPS numbers up to six decimal places. However, it is difficult for users to respond to the exact location up to six decimal places. Thus, both modules express *lat* and *lon* of *m_gps* and *u_gps* to α decimal places and send them to the picture validation module.

The picture validation module compares differences in GPS data between *m_gps* and *u_gps* (Step C). The reason *m_gps* and *u_gps* are compared is to ensure that the user actually remembers the location of the picture based on *m_gps*. The δ value is an allowable difference between *m_gps* and *u_gps*. If the δ value is too large, the module determines that the user remembers the location of the picture even if another location is entered, and if the δ value is too small, the user may have difficulty answering exactly where *m_gps* points to. If the difference between *m_gps* and *u_gps* is equal to or less than δ , the picture validation module determines that the user remembers the exact location and the module completes the process.

After a predefined amount of $N(pic)$ gathering is completed, the picture validation module checks the timestamp of each picture to check if pictures were taken on the same day. If provided pictures were taken on the same day, the module determines that the locations of the pictures have been duplicated and subsequently terminates the process. If the timestamp of the pictures indicate a different date, the module deletes metadata of all provided pictures.

Following this, the picture validation module generates a seed and sends the seed with the pictures to the key generation module. The key generation module generates a private key (Step D) using the seed created by multiple *u_gps* from the picture validation module. The module generates a private key (*o_sk*), and public key (*o_pk*), respectively. The pictures that have deleted metadata and *o_pk* are stored in a repository. When the whole process is complete, the module sends *o_sk*, and *o_pk* to the user.

3.4. Private Key Recovery Process

This section explains the private key recovery process. The private key recovery process works through picture load module, memory query module, temporary key generation module, and key validation module. The private key recovery process represents in Figure 4. The private key recovery process starts with bringing pictures and *o_pk* from a repository by the picture load module. In this paper, we do not consider repository techniques such as the form of storing pictures and *o_pk*, security of repository, or authentication technique.

When a user requests the private key recovery in a wallet application, the wallet application sends the user's basic information to the picture load module. Based on the user's information, the picture load module brings pictures and *o_pk* that are related to the user from the repository. After this, the picture load module sends pictures to the memory query module. Similar to the location query module in the private key generation

process, the memory query module works in conjunction with the user and the specific map application (Step A). The memory query module asks the user about the location of the picture. The module collects GPS data until $N(pic)$ and the module expresses GPS data to α decimal places (Step B). Based on GPS data, the memory query module creates a temporary seed and sends the temporary seed to the temporary key generation module.

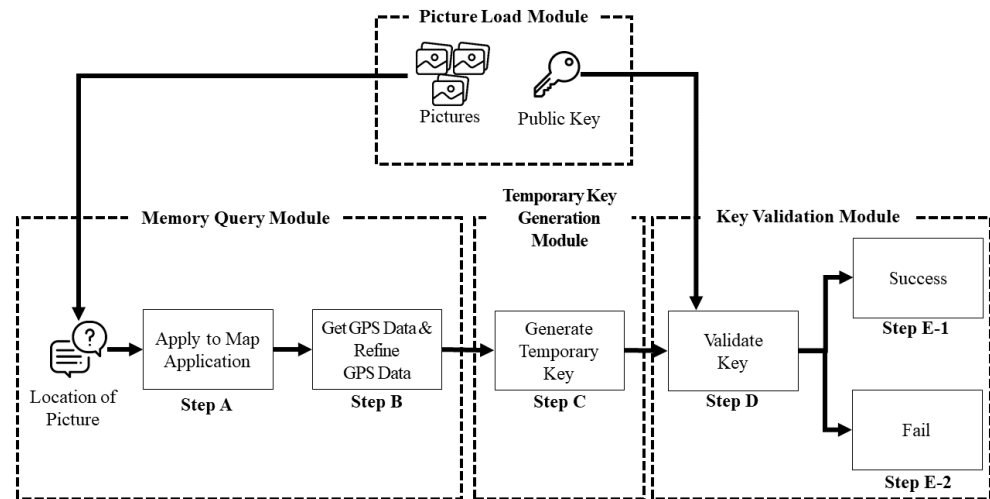


Figure 4. Private Key Recovery Process.

The temporary key generation module generates t_sk, t_pk by the temporary seed (Step C). After the generation t_pk is complete, the temporary key generation module sends t_sk, t_pk , and o_pk to the key validation module.

After the key validation module is received t_sk, t_pk from the temporary key generation module, the key validation module brings o_pk from the picture load module. Next, the key validation module verifies t_sk has been completed in the same way as o_sk (Step D). If the user responds to each picture correctly in the memory query module, the seed used in the temporary key generation and the key generation module have the same value. Eventually, o_sk and t_sk will be the same, and o_pk and t_pk generated through o_sk and t_sk will be the same. If o_pk and t_pk are the same, the module determines that the private key recovery process is completed and provides the value of t_sk to the user (Step E-1). If o_pk and t_pk are different, the key validation module terminates the process (Step E-2) because the user responded incorrectly in the memory query module.

4. Experiment

In this section, we describe the different experiments conducted to solve several research questions related to our approach.

- (RQ1) How secure is the proposed approach when variables indicate in the preliminary study (Section 4.1) are applied?
- (RQ2) What is the usability of the real-world application when used by users?

4.1. Preliminary Study

In this section, we conduct experiments to obtain the three variables mentioned in the approach before answering the research questions (RQ1, RQ2). There are three variables defined in our approach: (1) *decimal places in GPS data* (α), (2) *allowable difference value* (δ), and (3) *number of pictures for generating a private key* ($N(pic)$). Deciding upon the three variables should be undertaken carefully because these variables directly affect security and usability. If high security is applied to our approach through the variables, usability is decreased. On the contrary, increased usability reduces security. We experimented to determine applicable values for each of these variables. The experiments conducted are described in subsections.

4.1.1. Decimal Places in the GPS Data (α)

As discussed in Section 3.3, the reason to express GPS data of m_gps and u_gps to α decimal places is to facilitate comparison because decimal places in the GPS data can be changed easily by external factors, and the user. Decimal places in GPS data influence the number of elements in the GPS set (g_set). The g_set is a collection of GPS data that can express from a combination of lat and lon . In other words, the g_set is the number of cases that a user can have by u_answer . If the g_set contains a large number of GPS data, the security can be increased because of the lower risk of private key overlapping. Furthermore, a large number of GPS data can decrease the hacking probability of a brute-force attack. On the contrary, a small number of GPS data increases the risk of private key duplication and a brute-force attack.

Table 2 shows the number of GPS data in g_set depending on α . The first column indicates decimal places in the GPS data. The GPS data can be expressed up to eight decimal places ($\alpha = 8$). However, because Google Maps, the most commonly used GPS application, expresses GPS values up to six decimal places ($\alpha = 6$), the first column represents only $\alpha = 0$ to $\alpha = 6$. The second column indicates the number of GPS data that can occur from a combination of lat and lon . Since 70% of the Earth’s surface is water, 30% of the places where users can take pictures are presented as the *Number of GPS data on land* in the last column.

Table 2. Number of GPS data depending on α .

α	Number of GPS Data	Number of GPS Data on Land
$\alpha = 0$	65,341 ($lat = 181, lon = 361$)	19,602
$\alpha = 1$	6,485,401 ($lat = 1,801, lon = 3,601$)	1,945,620
$\alpha = 2$	648,054,001 ($lat = 18,001, lon = 36,001$)	194,416,200
$\alpha = 3$	64,800,540,001 ($lat = 180,001, lon = 360,001$)	19,440,162,000
$\alpha = 4$	6,480,005,400,001 ($lat = 1,800,001 lon = 3,600,001$)	1,944,001,620,000
$\alpha = 5$	648,000,054,000,001 ($lat = 18,000,001, lon = 36,000,001$)	194,400,016,200,000
$\alpha = 6$	64,800,000,540,000,001 ($lat = 180,000,001, lon = 360,000,001$)	9,440,000,162,000,000

As shown in Table 2, increasing the decimal places of GPS data increases the number of elements in g_set . For security concerns, choosing $\alpha = 6$ may seem the right choice because a large α value can prevent overlapping a private key and the potential of a brute-force attack. However, the proposed approach requires consideration of how many decimal places users can perceive through u_answer . For example, when $\alpha = 6$, if a user cannot recognize the difference of place between 0.000001 and 0.000002 on GPS data, $\alpha = 6$ is exceptionally difficult to apply even if it is more secure.

The decimal places of GPS data are used to accurately represent a location, and a numerical change of a specific digit means a change in the place of distance. In addition, the changes in distance value are different depending on lat and lon . The lat 1° is always expressed at 111 km (40,075,161.2 m/360°) because the circumference of the Earth is 40,075,161.2 m ($2\pi \times 6,378,106$ m) and the lon consists of a total of 360°.

The lon of 1° is characterized by varying distance depending on the lat because the Earth is not perfectly spherical. The lon can be obtained by a formula of $2\pi \times 6,378,106 \times \frac{\cos(lat)}{360^\circ}$ and it can be arranged as shown in Table 3. According to Table 3, when lat 0° ,

change of *lon* by 1° means a change of 111 km. Furthermore, when *lat* 10°, a change in *lon* by 1° indicates a change of 109 km. In this paper, to simplify the calculation, it is assumed that *lat* and *lon* have the same distance change (second row in Table 3).

Table 3. Distance changes by *lon* 1°.

<i>lat</i>	Distance Change by <i>lon</i> 1° Changes
0°	111 km
10°	109 km
20°	105 km
30°	99 km
40°	90 km
50°	78 km
60°	65 km
70°	50 km
80°	34 km
90°	17 km

If the GPS data are expressed with six decimal places ($\alpha = 6$), a unit change in GPS data at the 6th decimal place represents a change in distance of 11.1cm. Similarly, a unit change in other decimal places are as follows: $\alpha = 5$ means a change in distance of 1.11 m, $\alpha = 4$ is 11.1 m, $\alpha = 3$ is 111 m, $\alpha = 2$ is 1.11 km $\alpha = 1$ is 11.1 km.

If decimal places of GPS data are expressed as values between $\alpha = 4$ and $\alpha = 6$, it means that the user should recognize in distance smaller than a tennis court (length: 23.77 m, width: 10.97 m). For example, when $\alpha = 4$ is applied to the proposed approach, a user should separate positions at the opposing ends of a tennis court for answering the location of a picture. Case of $\alpha = 3$ also means that a user should be aware of the change in location within a range less than an American football field (length: 109.7 m, width: 48.4 m) or soccer field (length: 150 m, width: 68 m).

As a result, choosing a value for $\alpha = 3\sim 6$ may increase the security of the approach due to the number of GPS data in *g_set* being large, but it may pose difficulties to the user in terms of usability. In other words, the value of α should be defined as less than three ($\alpha < 3$) for a user to answer location information. Since the highest number of GPS data among the three values ($\alpha = 0\sim 2$), the case of $\alpha = 2$ can be considered as an applicable value for the proposed approach.

4.1.2. Allowable Difference in GPS Data (δ)

The previous section established that expressing GPS data in second decimal places ($\alpha = 2$) is the best option for a user to indicate the location of a picture. However, even if the GPS data is expressed in second decimal places, the user’s response may differ from the actual GPS value recorded in the picture. For example, when a user identifies the location of a picture taken in front of Niagara Falls, a user can give a detailed answer such as a street address but another user can give an ambiguous answer such as Buffalo. Through an experiment, we tried to find out how users tend to answer the location question for a picture and how great is the difference from the actual *m_gps* value.

An experiment was conducted with a total of 51 participants. Each participant provided 10 pictures of their own and answered the locations of each picture. Based on the participant’s answer (*u_answer*), *u_gps* was obtained through Google Maps. If *u_gps* was not immediately specified on Google Maps because *u_answer* was ambiguous, participants selected a location by themselves. Afterwards, the *m_gps* and *u_gps* were compared to measure differences. Because of the large amount of raw data, the raw data of this experiment are shared on Github (https://github.com/jungwonrs/experiment_rawdata/blob/main/mppm_raw_data.md), accessed on 2 May 2022. The results are as below:

- The smallest difference value of *lat* is 0.00002 and *lon* is 0.000004.
- The largest difference value of *lat* is 0.291382 and *lon* is 0.182889.

- The average difference value of *lat* is 0.009597341 and the standard deviation is 0.027383761.
- The average difference value of *lon* is 0.009685855 and the standard deviation is 0.023168967.

As a result of this experiment, most participants preferred to refer to the town rather than the street address. The difference between *m_gps* and *u_gps* was about 0.01. Based on this experiment, we can conclude that the location identified by the participants and the actual location are the same when the difference in GPS data is less than 0.01, and hence, a value of 0.01 can be set as the threshold for allowable difference ($\delta = 0.01$).

4.1.3. Number of Pictures for Generating a Private Key ($N(pic)$)

Pictures are the most important element of our approach and should be provided by a user. Because pictures can expose a user’s privacy, the user may not want to provide pictures even if pictures help to generate and recover the user’s private key. Thus, we wanted to know whether a user is willing to provide their pictures for generating and recovering a private key. We used a Google Survey form where we provided a brief description of our approach and asked the participants how many pictures they would be willing to provide. A total of 105 participants answered the survey and the results of the survey are shown in Figure 5 and below:

- 11 participants responded that they were not willing to provide a single picture because of privacy concerns (Orange).
- 44 participants responded that they were willing to provide 10 pictures (Yellow).
- 9 participants responded that they were willing to provide more than 10 pictures (Green).
- 2 participants did not respond properly and they are expressed as N.P (Purple)

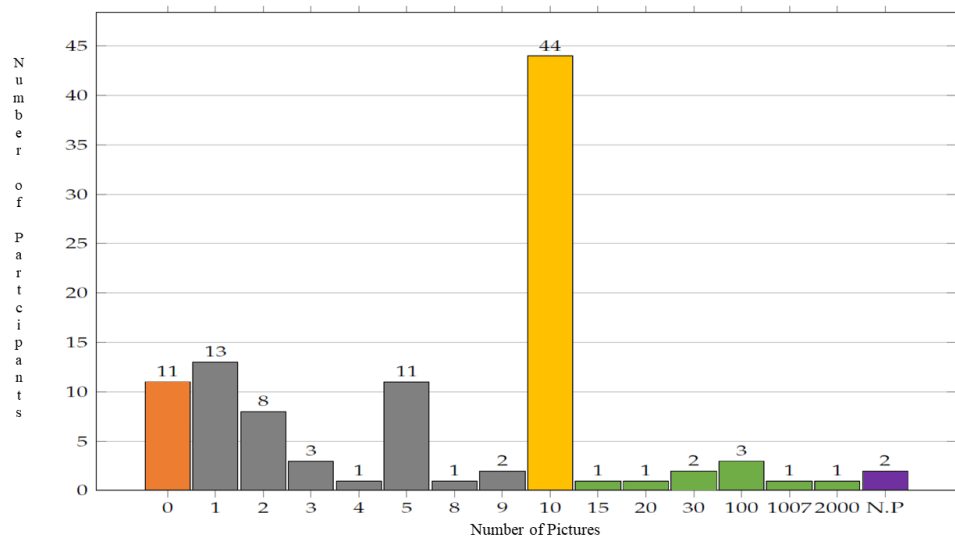


Figure 5. Survey results for $N(pic)$.

As shown in Figure 5, participants who were willing to provide 10 pictures comprise 41.9% of the total responses. Based on the survey, we concluded that requiring a minimum of 10 pictures would not be excessively burdensome for a user who wants to use key generation and recovery and, hence $N(pic)$ value can be set as more than 10 ($N(pic) \geq 10$) for our approach. Moreover, using a minimum of 10 pictures means that our approach has at least as many cases as $194,416,200^{10} \approx 7.77 \times 10^{82}$ to generate a private key when $\alpha = 2$.

4.2. Security of the Proposed Approach

This section describes the experiment we conducted to answer the first research question, (RQ1) **How secure is the proposed approach when variables indicate in the**

preliminary study are applied? The security of the proposed approach should be considered with respect to two aspects: (1) Risk of overlapping a private key, and (2) risk of private key recovery by others. Security was evaluated by applying $\alpha = 2$, $\delta = 0.01$, and $N(pic) \geq 10$ in the experiments conducted for the two aforementioned scenarios.

4.2.1. Risk of Overlapping Private Key

The risk of overlapping a private key refers to the overlap of seeds used to generate a private key between users. In this section, the risk of the overlapping private key is considered in view of two aspects: (A) Entropy, and (B) Probability.

A. Entropy

We conducted several experiments to calculate disorder through entropy and compared it with the mnemonic code to identify the risk of private key duplication. Entropy can generally express how disordered a particular set is. Entropy is also used by applying it to password generation, and password entropy is the measure of the quality of a password [48]. Furthermore, high entropy indicates difficulty in inferring and overlapping passwords due to the high disorder.

A private key is generated by a seed and the private key is used for identifying a blockchain user. Thus, a private key can be considered the same as a password. This section uses password entropy to compare the entropy between the mnemonic code technique and the proposed approach. The entropy is calculated using Equation 1, from [49,50].

$$Entropy = \log_2 S^L \tag{1}$$

where S refers to the size of a set in which a seed can be generated. L indicates the number of elements used from the set. Table 4 shows Equation 1 applied to the mnemonic code and the proposed approach.

Table 4. Entropy results.

Technique	S	L	Entropy
Mnemonic code	1.10×10^{31}	12	1237.416 bits
	1.72×10^{33}	13	1435.318 bits
	2.51×10^{35}	14	1646.295 bits
	3.40×10^{37}	15	1870.136 bits
	4.32×10^{39}	16	2106.642 bits
	5.16×10^{41}	17	2355.628 bits
	5.82×10^{43}	18	2616.919 bits
	6.22×10^{45}	19	2890.351 bits
	6.31×10^{47}	20	3175.767 bits
	6.09×10^{49}	21	3473.019 bits
	5.61×10^{51}	22	3781.966 bits
	4.94×10^{53}	23	4102.474 bits
	4.17×10^{55}	24	4434.412 bits
Reminisce	7.77×10^{82}	10	2753.457 bits
	1.50×10^{91}	11	3331.683 bits
	2.92×10^{99}	12	3964.978 bits
	5.67×10^{107}	13	4653.346 bits
	1.10×10^{116}	14	5396.736 bits
	2.14×10^{124}	15	6195.250 bits
	4.17×10^{132}	16	7048.873 bits
	8.10×10^{140}	17	7957.494 bits
	1.57×10^{149}	18	8921.125 bits
	3.06×10^{157}	19	9939.969 bits
5.95×10^{165}	20	11013.821 bits	

Mnemonic code requires a user to select from 12 words to 24 words from BIP-0039 for generating a private key, and the BIP-0039 consists of 2048 words [19]. Because the mnemonic code does not allow duplication to select words, S can be calculated with the $\frac{N!}{(K! \times (N-K)!)}$ formula. N is a pool of word size which is 2048 words, and K is the number of selected words which is the same as L . For example, if a user wants to create a private key using the mnemonic code of 12 words, $S = \frac{2048!}{(12! \times (2048-12)!)} \approx 1.10 \times 10^{31}$, and $L = 12$.

For Reminisce, S can be calculated with N^K because our approach does not consider the duplication of GPS data, and N can be obtained depending on α from Table 2. So, if a user wants to create a private key using the proposed approach of 10 pictures, $S = 194,416,200^{10} \approx 7.77 \times 10^{82}$ and $L = 10$.

As shown in Table 4, when a user uses 10 pictures to create a private key in our approach, the entropy is higher than using the mnemonic code of 18 words. In addition, the maximum entropy of the mnemonic code is 4434.412 bits by using 24 words and it is smaller than the proposed approach by using 13 pictures. Furthermore, although the mnemonic code has a maximum limit of 24 words, the entropy of the proposed approach may continue to increase depending on the user's choice because our approach has no maximum limit. As a result, the proposed approach is more disordered than mnemonic codes and which means that overlapping a private key is more difficult and unlikely to occur than using a mnemonic code.

B. Probability

In this section, we experimented to find out the probability of private key overlapping. A seed is created by u_gps based on u_answer and the number of seeds available for private key generation in a single picture is shown in Table 2.

The number of seeds available for private key generation is affected by the value of α and $N(pic)$. The value of α was fixed as 2 ($\alpha = 2$) as discussed in Section 4.1.1 which lets the user specify as many as 648,054,001 GPS data as a seed in a single picture but the actual number of GPS data available for a seed is 194,416,200 (Number of GPS data on land).

A user needs to use a minimum of 10 pictures ($N(pic) \geq 10$) for generating a private key in our approach. If the user uses 10 pictures, the number of GPS data available for a seed is $194,146,200^{10} \approx 7.71 \times 10^{82}$, and probability of overlapping a single seed is 1.296×10^{-83} which is smaller than $8.636 \times 10^{-78} \approx \frac{1}{2^{256}}$ (known as SHA-256 hash algorithm collision probability). Furthermore, 1.296×10^{-83} is smaller than $2.398 \times 10^{-56} \approx \frac{1}{4.17 \times 10^{55}}$ the probability of overlapping a private key when generating a private key with 24 words as seeds in the mnemonic code. As a result, a seed that is used to generate a private key in the situation $\alpha = 2$, $N(pic) \geq 10$ has a small risk of overlap.

4.2.2. Risk of Overlapping Private Key

To steal a user's private key, a hacker must attack the repository where pictures are safely stored for accessing a specific user's pictures and public key. Even if the repository hacking is successful, the hacker can only obtain the number of pictures that have been used to generate a private key. In this section, two attack scenarios are described, assuming that the hacker obtained 10 pictures of users.

A. Brute-Force Attack

A brute-force attack is an attack where hackers substitute all possible values to identify a particular value. In the proposed approach, a hacker can try the brute-force attack by combining all GPS data to create a seed. In this section, we describe an experiment that was conducted to see whether the theft of a private key was practically possible.

We experimented to find out how much time would be required to generate a private key. The hardware specifications of the computer used had an Intel i7-8700 CPU with 64GB RAM and Windows 10 OS installed. We continued generating GPS data for 10 locations as a seed and based on the seed, private and public keys were generated. The experiment

measured how many private and public keys could be generated over an hour, and the same process was repeated 300 times.

As a result of the experiment, the number of private keys generated in an hour, on average was 42,934,122 and the public keys, 28,662,193. Based on this result, if the hacker somehow knows $\alpha = 2$, the hacker must spend 2.68995×10^{75} hours ($\approx \frac{7.71 \times 10^{82}}{28,662,193}$) to complete public key generation through 10 random GPS combinations. The time of 2.68995×10^{75} hours is much larger than 4.03223×10^{13} hours ($\approx ((4.603 \times 10^9 \text{ years}) \times 8760)$), which equates to the age of the Sun. In conclusion, the brute-force attack may not be possible because it may take longer than the age of the Sun.

B. Random guessing attack

In the proposed approach, the random guessing attack can be executed by guessing GPS data where a hacker guesses a location by analyzing pictures stored in a repository. In order for the attack to succeed, a hacker needs to find a target that contains pictures with landmarks or topographic features.

Furthermore, assuming the hacker finds the target, the target has to contain less than three pictures, the locations of which are unknown to the hacker. For example, when $N(pic) = 10$, the hacker knows the location of seven pictures, and the hacker then tries to find the location of the remaining pictures. Because the hacker cannot guess the three pictures, the hacker should use a brute-force attack on the three pictures to obtain the GPS data. In this case, it takes 2.56×10^{17} h ($\approx \frac{194,416,200^3}{28,662,193}$) to make a seed by combining the GPS data of the three pictures with the seven pictures again. The time of 2.56×10^{17} h is less than 2.68995×10^{75} h but is still larger than 4.03223×10^{13} h, which is the age of the Sun.

Furthermore, in order to find out how many pictures users provide containing landmarks or topographic features, we analyzed 510 pictures that were previously provided by 51 participants and used in Section 4.1.2. We identified what was captured in the picture and classified it into six categories: animals, landscapes, food, people, undefined, and landmarks.

Figure 6 shows examples of pictures classified in each category. Each picture shown in Figure 6 is numbered in the upper left corner. The pictures in Figure 6 represent animals, landscapes, food, and people in numerical order, respectively. The undefined category included pictures such as accessories, posters, and cell phones, while the landmark category included pictures that could infer places by signboards or famous places such as Torre Di Pisa.

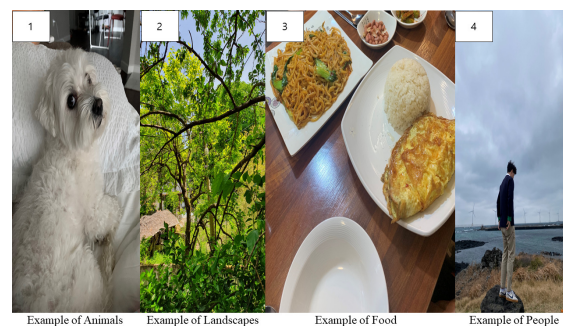


Figure 6. Example of pictures.

Table 5 shows the results of classifying 510 pictures and indicates the category and the number of pictures in each category.

Table 5. Classifying pictures by category.

Category	Animals	Landscapes	Food	People	Undefined	Landmark
Number of Pictures	46	56	222	60	68	58

As a result of classification, only 11.4% of the pictures can infer specific locations. Additionally, we classified pictures to see how many landmark pictures a person provides in a set that contains 10 pictures given by one person, and the results are in Table 6.

Table 6. Classifying sets by number of landmark pictures.

Number of Landmarks Pictures	0	1	2	3	4
Number of set	14	23	9	3	2

As shown in Table 6, the maximum number of landmark pictures in one set is four, and sets including four landmark pictures is two. Furthermore, most sets are classified as containing only one landmark picture. As a result of classifying 510 pictures, we observed that when people provide pictures, they do not consciously provide pictures that are inherently easy to remember such as places with landmarks, but instead provide pictures randomly based on their own choices.

In addition, for a successful random guessing attack to occur, a hacker not only needs the ability to attack the repository, but the hacker is likely to succeed only when there are fewer than two pictures that cannot be inferred from targets. In the proposed approach, the picture validation module prevents the use of multiple pictures taken on the same day. Thus, it is safe from the random guessing attack unless the user intentionally provides pictures of the landmark taken at the same place on different days.

4.3. Usability of the Real-World Application

In this section, we describe the experiment with real-world application of our proposed approach in order to answer the second research question, **(RQ2) What is the usability of the real-world application of our approach by users?** We used the variables obtained from Section 4.1 for this usability experiment. We asked the participants who provided 10 pictures in response to the survey to participate in this experiment and a total of 20 participants agreed to join the experiment.

Figure 7 shows the user interface (UI) of the application. The shaded boxes do not appear in the real application and they are inserted in this paper only to hide the information of pictures. The left screen shows the UI of the user choosing 10 pictures, and allows the user to select only pictures that contain GPS data for the picture selection. The right UI is the process of validating the picture one by one that has been selected from the left UI. The right UI is the same used in the process of generating and recovering a private key. On the right screen, the user can enter a location for a picture and press the enter button to search on the map application. If a searched location is correct, the user can press the verification button, and if a searched location is different from a location they thought, they can move the map to select the desired location. A picture verification is complete if the difference between m_gps and u_gps is equal to or less than 0.01 ($\delta = 0.01$).

We worked with 20 participants one by one and showed them how to use the application and generate a private key using their own pictures. Each participant joined the experiment by downloading the wallet application called COLET from Apple App Store (<https://apps.apple.com/kr/app/colet/id1503144673>, accessed on 2 May 2022) or Google Play Store (<https://play.google.com/store/apps/details?id=com.colet> accessed on 2 May 2022). All participants demonstrated no difficulties in generating a private key and a month later, the private key recovery experiment was conducted.

All participants were able to recover their own private keys without difficulty after a month. Of the 20 participants, 18 succeeded in recovering their private keys in their first attempt, 1 (P) required two attempts, and the other (K) was successful in recovering after four attempts. The results of this experiment are shown in Table 7.

The Index column represents the participants and the attempt column represents the number of attempts. The last column shows the time taken to recover a private key

when participants succeeded in the private key recovery. As shown in Table 7, participants successfully performed private key recovery after a month but private key recovery time varied from person to person.

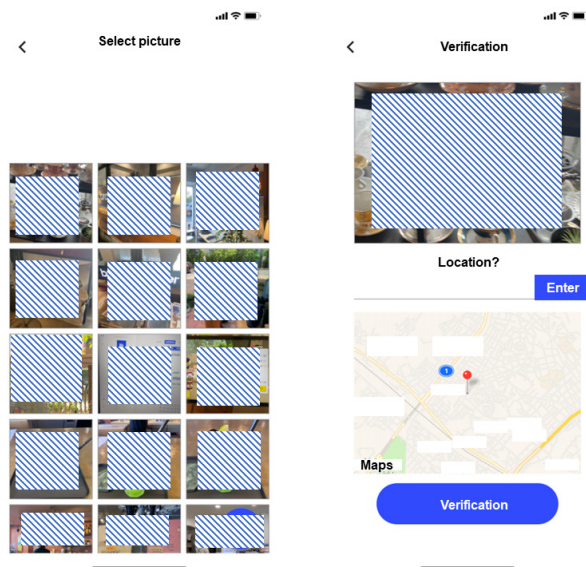


Figure 7. Application UI.

Table 7. Private key recovery attempt and time.

Index	Attempt	Recovery Time (min)
A	1	2.36
B	1	2.05
C	1	2.03
D	1	1.10
E	1	1.31
F	1	1.20
G	1	1.51
H	1	1.31
I	1	3.54
J	1	4.36
K	4	8.29
L	1	1.25
M	1	1.39
N	1	5.18
O	1	1.80
P	2	1.60
Q	1	4.35
R	1	1.51
S	1	1.60
T	1	3.29

The fastest recovery recorded was 1.10 min, while the slowest was 8.29 min. The average amount of time spent recovering a private key is 2.56 min. According to this experiment, depending on which pictures were used to generate the private key, the time to recall the locations of pictures differed for each participant. Participants who were quick to recover their private keys had used pictures of their favorite or frequently visited places in the private key generation process. Those who were slower to recover their private keys consumed time recalling overseas locations which were less familiar locations.

4.4. Threats to Validity

(Threat to construct validity) The results of entropy calculations in Section 4.2.1 may be affected by using different formulas. Since the private key is used to identify a user such as the password technique, the formula for measuring password entropy was used in the entropy measurement of the private key seed. However, the password entropy continues to be studied and if the new password entropy formula is applied to Section 4.2.1, the results of the entropy calculations may be affected. Moreover, if there is a direct way to measure the entropy of a private key seed, the entropy calculation results in Section 4.2.1 also may be affected.

(Threats to statistical conclusion validity) The results of δ and $N(pic)$ may be influenced by the number of participants. We experimented with 51 participants and used 510 pictures to define δ . Furthermore, $N(pic)$ is defined by 105 participant surveys. In order to avoid bias from participants, we tried to organize recruitment diverse as possible by age, and nationality. However, if the number of participants is increased, the value of the defined variables could be affected.

(Threats to external validity) In Section 4.3, the 20 participants had experience using blockchain wallets and were between the ages of 20 and 40. In other words, the participants in the experiment were already familiar to some extent in using blockchain wallet applications. However, if participants have never used a blockchain wallet application and are unfamiliar with mobile applications, the results in Section 4.3 may be affected.

5. Conclusions

Cryptocurrency is expected to spark new innovations and be highly valuable for not only individual trade but also in various other industries. The blockchain wallet is an application that bridges the gap between blockchain networks and the real world. The mnemonic code technique is the most widely used method to generate and recover a private key in the blockchain wallet. However, the mnemonic code technique does not consider usability to generate and recover a private key for users. Our approach is based on the idea that a user can hold long-term memory from personal pictures.

When generating a private key, a user provides distinctive pictures to a wallet application. Based on the provided pictures, the user can generate a private key. Furthermore, it is possible for the user to recover a private key through the pictures that were used in the private key generation process. In order to demonstrate the security and usability of our approach, we have considered various perspectives from participants and used mathematical methods. Through our experimentation, we have demonstrated that our approach is usable and secure.

This study only considers applying fixed variables to all the pictures. Thus, we plan to utilize dynamic variables and apply different parameters to each picture in our future work. Furthermore, additional research is needed to process personal information such as faces and landmarks that can be used to infer the location of each picture and at the same time protect the privacy of individuals in the pictures.

Author Contributions: Conceptualization, J.S. and D.K.; Funding acquisition, S.P.; Investigation, J.S.; Methodology, D.K.; Supervision, S.P.; Writing—original draft, J.S., V.S., S.P.; writing—review and editing, S.K., V.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2022-2017-0-01628) supervised by the IITP (Institute for Information & Communications Technology Promotion).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. 5 Blockchain Trends for 2020. Available online: <https://www.fm-house.com/wp-content/uploads/2020/07/5-Blockchain-Trends-for-2020.pdf> (accessed on 4 March 2022).
2. Chen, Y.; Lu, Y.; Bulysheva, L.; Kataev, M.Y. Applications of Blockchain in Industry 4.0: a Review. *Inf. Syst. Front.* **2022**, *24*, 1–15. [CrossRef]
3. Zile, K.; Strazdina, R. Blockchain Use Cases and Their Feasibility. *Appl. Comput. Syst.* **2018**, *23*, 12–20. [CrossRef]
4. Makridakis, S.; Christodoulou, K. Blockchain: Current Challenges and Future Prospects/Applications. *Future Internet* **2019**, *11*, 258. [CrossRef]
5. Burer, M.J.; de Lapparent, M.; Pallotta, V.; Capezzali, M.; Carpita, M. Use cases for Blockchain in the Energy Industry Opportunities of emerging business models and related risks. *Comput. Ind. Eng.* **2019**, *137*, 106002. [CrossRef]
6. Le, Tu.; Hsu, Ch.; Chen, We. A Hybrid Blockchain-Based Log Management Scheme with Non-Repudiation for Smart Grids. *IEEE Trans. Ind. Inform.* **2021**, 1–12. [CrossRef]
7. Choi, Ts.; Siqin, T. Blockchain in logistics and production from Blockchain 1.0 to Blockchain 5.0: An intra-inter-organizational framework. *Transp. Res. Part E* **2022**, *160*, 102653. [CrossRef]
8. Foley, S.; Karlsen, J.R.; Putnins, T.J. Sex, Drugs, and Bitcoin: How Much Illegal Activity Is Financed through Cryptocurrencies?. *Rev. Financ. Stud.* **2019**, *32*, 1798–1853. [CrossRef]
9. Trozze, A.; Kamps, J.; Akartuna, E.A.; Hetzel, F.J.; Kleinberg, B.; Davies, T.; Johnson, S.D. Cryptocurrencies and future financial crime. *Crime Sci.* **2022**, *11*, 1. [CrossRef]
10. Hornuf, L.; Kuck, T.; Schwienbacher, A. Initial coin offerings, information disclosure, and fraud. *Small Bus. Econ.* **2022**, *58*, 1741–1759. [CrossRef]
11. CoinMarketCap. Available online: <https://coinmarketcap.com/> (accessed on 7 June 2022).
12. Lansky, J. Possible State Approaches to Cryptocurrencies *J. Syst. Integr.* **2018**, *9*, 19–31. [CrossRef]
13. Pelaez-Repiso, A.; Sanchez-Nunez, P.; Calvente, Y.G. Tax Regulation on Blockchain and Cryptocurrency: The Implications for Open Innovation. *J. Open Innov. Technol. Mark. Complex.* **2021**, *7*, 98. [CrossRef]
14. Choi, T. Creating all-win by blockchain technology in supply chains: Impacts of agents' risk attitudes towards cryptocurrency. *J. Oper. Res. Soc.* **2021**, *72*, 2580–2595. [CrossRef]
15. Mas'ud, M.Z.; Hassan, A.; Shah, W.M.; Abdul-Latip, S.F.; Ahmad, R. A Review of Digital Forensics Framework for Blockchain in Cryptocurrency Technology. In Proceedings of the 2021 3rd International Cyber Resilience Conference (CRC), Langkawi Island, Malaysia, 29–31 January 2021.
16. Liu, X.F.; Jiang, Xi.; Liu, Si.; Tse, C.K. Knowledge Discovery in Cryptocurrency Transactions: A survey. *IEEE Access* **2021**, *9*, 37229–37254. [CrossRef]
17. Varghese, H.M.; Nagoree, D.A.; Anshu; Jayapandian, N. Cryptocurrency Security and Privacy Issues: A Research Perspective. In Proceedings of the 2021 6th International Conference on Communication and Electronics Systems (ICCES), Coimbatre, India, 8–10 July 2021.
18. Li, X.; Jiang, P.; Chen, T.; Luo, X.; Wen, Q. A survey on the security of blockchain systems. *Future Gener. Comput. Syst.* **2020**, *107*, 841–853. [CrossRef]
19. bitcoin/bips. Available online: <https://github.com/bitcoin/bips/blob/master/bip-0039/english.txt> (accessed on 5 February 2022).
20. Exclusive: Nearly 4 Million Bitcoins Lost Forever, New Study Says. Available online: <https://fortune.com/2017/11/25/lost-bitcoins/> (accessed on 3 February 2022).
21. \$190 Million in Crypto Gone Forever, How Canada's Biggest Bitcoin Exchange Lost it All. Available online: <https://finance.yahoo.com/news/190-million-crypto-gone-forever-213010166.html> (accessed on 3 February 2022).
22. Li, G.; You, L. A Consortium Blockchain Wallet Scheme Based on Dual-Threshold Key Sharing. *Symmetry* **2021**, *13*, 1444. [CrossRef]
23. Gurfidan, R.; Ersoy, M. Blockchain-Based Music Wallet for Copyright Protection in Audio Files. *J. Comput. Sci. Technol.* **2021**, *21*, 11–19. [CrossRef]
24. Han, J.; Song, M.; Eom, H.; Son, Y. An Efficient Multi-signature Wallet in Blockchain Using Bloom Filter. In Proceedings of the SAC'21: Proceedings of the 36th Annual ACM Symposium on Applied Computing. New York, United States, 22–26 March 2021.
25. Sung, S. A new key protocol design for cryptocurrency wallet. *ICT Express* **2021**, *7*, 316–321. [CrossRef]
26. Soltani, R.; Nguyen, U.T.; An, A. Practical Key Recovery Model for Self-Sovereign Identity Based Digital Wallets. In Proceedings of the 2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCOM/CyberSciTech), Fukuoka, Japan, 4 November 2019.
27. Zhu, F.; Chen, W.; Wang, Y.; Lin, P.; Li, T.; Cao, X.; Yuan, L. Trust your wallet: A new online wallet architecture for Bitcoin. In Proceedings of the 2017 International Conference on Progress in Informatics and Computing (PIC), Nanjing, China, 15–17 December 2017.
28. He, S.; Wu, Q.; Luo, X.; Liang, Z.; Li, D.; Feng, H.; Zheng, H.; Li, Y. A Social-Network-Based Cryptocurrency Wallet-Management Scheme. *IEEE Access* **2018**, *6*, 7654–7663. [CrossRef]

29. He, X.; Lin, J.; Li, K.; Chen, X. A Novel Cryptocurrency Wallet Management Scheme Based on Decentralized Multi-Constrained Derangement. *IEEE Access* **2019**, *7*, 185250–185263. [CrossRef]
30. Private key encryption and recovery in blockchain. Available online: <https://arxiv.org/abs/1907.04156> (accessed on 19 December 2020).
31. Zhao, H.; Zhang, Y.; Peng, Y.; Xu, R. Lightweight Backup and Efficient Recovery Scheme for Health Blockchain Keys. In Proceedings of the 2017 IEEE 13th International Symposium on Autonomous Decentralized System (ISADS), Bangkok, Thailand, 22–24 March 2017.
32. Singh, H.P.; Stefanidis, K.; Kirstein, F. A Private key Recovery Scheme Using Partial Knowledge. In Proceedings of the 2021 11th IFIP International Conference on New Technologies, Mobility and Security (NTMS), Paris, France, 19–21 April 2021.
33. Haber, S.; Stornetta, W.S. How to time-stamp a digital document. *J. Cryptol.* **1991**, *3*, 99–111. [CrossRef]
34. Chaum, D.; Fiat, A.; Naor, M. *Untraceable Electronic Cash*. *CRYPTO 1988: Advances in Cryptology*; Springer: New York, NY, USA, 1990.
35. Bayer, D.; Haber, S. Improving the Efficiency and Reliability of Digital Time-Stamping. In *Sequences II Methods in Communication, Security, and Computer Science*; Springer: New York, NY, USA, 1993.
36. Metamask. Available online: <https://metamask.io/> (accessed on 8 June 2022).
37. Johnson, D.; Menezes, A.; Vanstone, S. The Elliptic Curve Digital Signature Algorithm (ECDSA). *Int. J. Inf. Secur.* **2001**, *1*, 36–63 [CrossRef]
38. Reed, I.S. G.Solomon Polynomial codes over certain finite fields. *J. Soc. Ind. Appl. Math.* **1960**, *8*, 300–304. [CrossRef]
39. Juels, A.; Sudan, M. A Fuzzy Vault Scheme. *Des. Codes Cryptogr.* **2006**, *38*, 237–257. [CrossRef]
40. HM, the Man with No Memory. Available online: <https://www.psychologytoday.com/us/blog/trouble-in-mind/201201/hm-the-man-no-memory> (accessed on 15 February 2021).
41. Squire, L.R.; Zola-Morgan, S. The Medial Temporal Lobe Memory System. *Science* **1991**, *253*, 1380–1386. [CrossRef] [PubMed]
42. Robertson, L.T. Memory and the Brain. *J. Dent. Educ.* **2002**, *66*, 30–42. [CrossRef] [PubMed]
43. Cahill, L. Neurobiological mechanisms of emotionally influenced, long-term memory. *Prog. Brain Res.* **2000**, *126*, 29–37.
44. Allan Paivio, Kalman Csapo. Picture superiority in free recall: Imagery or dual coding?. *Cogn. Psychol.* **1973**, *5* 176–206. [CrossRef]
45. Gay, M.M.Z.S.J. The picture superiority effect: Support for the distinctiveness model. *Am. J. Psychol.* **1999**, *112*, 113–146.
46. Whitehouse, A.J.O. The development of the picture-superiority effect. *Br. J. Dev. Psychol.* **2010**, *24*, 767–773. [CrossRef]
47. Hockley, W.E. The picture superiority effect in associative recognition. *Mem. Cogn.* **2008**, *36*, 1351–1359. [CrossRef]
48. Ma, W.; Campbell, J.; Tran, D.; Kleeman, D. Password Entropy and Password Quality. In Proceedings of the 2010 Fourth International Conference on Network and System Security, Melbourne, VIC, Australia, 1–3 September 2010.
49. Nizamani, S.Z.; Hassan, S.R.; Shaikh, R.A.; Abozinadah, E.A.; Mehmood, R. A Novel Hybrid Textual-Graphical Authentication Scheme with Better Security, Memorability, and Usability. *IEEE Access* **2021**, *9*, 51294–51312. [CrossRef]
50. How to Calculate Password Entropy? Available online: <https://generatepasswords.org/how-to-calculate-entropy> (accessed on 4 February 2022).