Escrovery: Decentralized Account Recovery With Escrowed Payments

Jeff Pickhardt
Palo Alto, California
pickhardt@gmail.com

Abstract

In decentralized systems like Bitcoin or Ethereum, when a user loses their secret key, they lose their entire account. Existing methods of account recovery either give up decentralization or require persistence of information by the user. Instead, this paper introduces Escrovery, a method of account recovery in trustless systems that is decentralized, non-persistent, anonymous, and secure. Escrovery uses escrowed payments to deter malicious recovery attempts. Any user may make an Escrovery challenge to recover any account by first placing an amount of money in escrow. If the original account owner responds to the challenge in a certain amount of time, they earn the escrow. Otherwise, the challenger takes ownership of the account and recovers their escrow.

1 Introduction

Despite their best intentions, online users often forget their login credentials, locking them out of their accounts. To mitigate this, account recovery mechanisms enable users to recover their accounts. For example, Google had over 11 million account recovery attempts in 2013 [1].

Many trustless systems like Bitcoin and Ethereum have no built-in method of account recovery. It is estimated that 13% to 22% of bitcoins are lost forever, with a present value of $23 billion dollars [2]. The ability to recovery accounts in case of a forgotten password or lost private key is important for user trust and mass adoption, and is a compelling reason many users prefer to keep their bitcoin with centralized companies like Coinbase. Other commonly suggested recovery methods include backing up the secret or writing down a mnemonic that can be used to recover the secret, however these introduce additional risks, and are often misunderstood by users [3].

A method of account recovery compatible with trustless systems, without resorting to a trusted, centralized service like Coinbase, would help decentralized systems better fulfil the dream of decentralization.

Specifically, we seek an account recovery mechanism that satisfies the following four conditions. First, it is decentralized, meaning it does not depend on one or a small number of third party authorities. Second, it is non-persistent, meaning it does not require users to persist any data they wouldn't already know, such as memorizing a sequence of random words. Importantly, the non-persistent requirement doesn't prevent remembering data the user would already know, like their name or favorite movie. Third, it is anonymity-preserving, meaning it is at least as anonymous as the underlying account. If the underlying account is anonymous, then recovering it should be able to preserve that anonymity. Fourth, it is secure, which can only be defined relatively as secure against a given threat model.

We propose an account recovery mechanism where any user can submit a challenge to recover any account, called an Escrovery challenge, by depositing an amount in escrow and identifying the account to recover. If the true account owner responds to the challenge within a preset amount of time, they keep their account and earn the escrowed amount. However, if the account owner never responds in time, the account is given to the challenger, along with the escrowed amount. This relies on the true account owner being the first one to realize that they've lost access to their account, and a degree of vigilance in case someone else makes a malicious Escrovery challenge against their account.

2 Other Account Recovery Methods

Although there are many existing methods of account recovery, none are decentralized, non-persistent, anonymity-preserving, and secure against relevant threat models.

Recovery with a designated secondary account is the method used by many centralized online services. In this case, the user designates a secondary account they trust, for example their phone number or second email address. To recover their account, a password reset link or security code is sent to the secondary account. This often deanonymizes the account by tying it to a phone number, which for centralized services is often a useful feature and not a bug. It does, however, limit its use in systems that value anonymity. This recovery method is not decentralized, since it relies on a centralized service provider able to send the reset link to the secondary account.

Challenge questions are another frequently used method online. In this method, the user gives answers

to questions like “What was your high school mascot?” To recover the account, a user must answer these challenge questions. Challenge questions satisfy the non-persistent requirement, since it is assumed a user can easily recall their answers. However, this is an insecure method of account recovery, since many users have common answers, and it is vulnerable to statistical inference. For example, if you know the person is Korean, there’s a 25% chance you can answer a place of birth challenge question within 3 tries [1]. Further, if the attacker has offline access to the encrypted answer with unlimited tries, like they would with a blockchain, then challenge questions are not useful. In contrast, in the online system, the user is typically rate limited to a few number of tries. Thus, this method would not be secure if adapted to a blockchain like Ethereum.

Brain wallets use a memorable passphrase to deterministically generate private keys. This allows a user to remember a private key more easily by just remembering the brain wallet. This is similar to challenge questions, if the challenge question were “What is your memorable passphrase?”. Brain wallets have been suggested as a useful way for bitcoin holders to store their bitcoin, because their key isn’t tied to a single computer. However, they don’t ofter any protection against forgetting the passphrase. Further, there are hackers actively looking for brain wallet usage by guessing common passphrases, then stealing the accounts [4]. Use of brain wallets is not very widespread and not recommended. For our purposes, brain wallets do not satisfy the non-persistent condition, since it requires the user recall a passphrase.

Secret sharing can be used to split a secret into n parts, where only k are needed to recover the secret [5]. This can be used as a form of account recovery by sharing the n parts amongst various cloud storage, hard drives, or friends. For example, if n=5 and k=3, a person could split their secret into five parts, and share one part with a different trusted friend. If they need to recover their secret, they simply ask their friends for their secret parts. As long as at least three of the friends still have their parts, the user can reconstruct their original secret. Since a single compromised hard drive or account often leads to more comprised accounts, sharing the parts among a single user’s accounts is not very secure. Further, the need to trust and bother n friends to hold a secret part is a large downside to this strategy. If k friends are compromised or collude, the account would be at risk. If a user stores the keys themselves, this method does not satisfy the non-persistent condition; however, if the user relies on friends, it is insecure.

Password backups are often used by less security minded users. In this case, they simply write their password on a post-it note or save it on an external hard drive or backup email. Predictably, this has serious security implications, like when a Hawaiian emergency agency worker’s post-it note ended up on the internet [6]. This is insecure and also persistent.

Hybrid methods combine two or more methods together, providing further advantages than a single method by itself. For example, it is common for challenge questions to be the first line of defence, followed by secondary account recovery. Only if a user answers a challenge question successfully will a recovery link be sent to their secondary email or a code is texted to their phone number.

3 Design

Escrovery is a method for account recovery using escrow, the name being a portmanteau of escrow and recovery. In Escrovery, any challenger can initiate an Escrovery challenge against an account by concatenating the account identifier with a new public key, then submitting a hash of the concatenated identifiers along with some amount of tokens as payment. After this challenge achieves consensus, the Escrovery challenger submits a second part containing the account identifier and the new public key in the clear. The second part should come no more than a short time period (such as four hours) after the first part to be valid. This two-phase commit and reveal procedure prevents front-running, and has been used with name registration by blockchains including Namecoin and Blockstack [7, 8].

The amount is kept in escrow for a waiting period, during which the original account owner may respond to the Escrovery challenge. If the original account owner responds to the challenge, some fraction of the escrowed amount is awarded to them. If not, after the waiting period, the account is considered to be transferred to the new public key, along with all or a fraction of their escrow. In case there are two or more unresponded Escrovery challenges, the earliest one takes precedence.

The amount of escrow awarded can either be the entire escrowed amount, or a fraction, with the remaining fraction being given to some other account (like a governance organization) or simply burned. This amount can differ in cases of successful and unsuccessful challenges. For example, unsuccessful challenges could award 100% of the escrow to the rightful owner, while successful challenges could return 90% and award a 10% fee to a governance

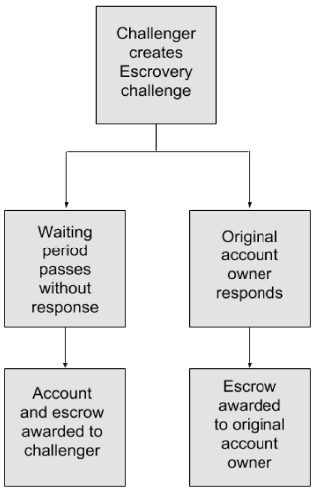

Figure 1: Two possible outcome paths

organization, analogous to how many landlords charge for forgotten key replacement.

To make it frictionless for users, Escrovery checks can be built in to applications or protocols so that ordinary usage also responds to all outstanding Escrovery challenges. For example, consider a decentralized social network, like a decentralized application of Facebook. Users typically sign in to Facebook at least once a week, often more than once a day. If the waiting period is set to a week, as long as users open their application at least once a week, they will be able to respond to fake Escrovery challenges against their account.

Both the amount of tokens and the waiting period duration are parameters that a user might want to change. For example, valuable or famous accounts may want to set their Escrovery amount high, whereas a typical user might make it more affordable. Further, a user going on a vacation without internet might want to set their recovery duration to a longer amount, or even disable recovery entirely, while a typical user might want to keep their recovery period short.

Users do not need to remember anything like a backup password or brain wallet passphrase. All they need to remember is how to find their account. In the case of Escrovery on a user-based system like a social network, this would be their username. In the case of Escrovery in a cryptocurrency like Bitcoin or Ethereum, they would need to be able to find the public key addresses of the accounts they want to claim. In some cases, they would be unable to do so, but in cases where they can remember other accounts or businesses they've transacted with, and the approximate amounts or time the transactions occurred, they may be able to trace the blockchain to find the public key addresses of their accounts.

To facilitate finding the account that a user wants to recover, Escrovery accounts allow an optional description string. Descriptions allow a user to mark their account, making it easier to find in the future. This may involve a certain loss of anonymity but would improve recoverability. For example, a user might tag their account with their last name, the name of their college, or the name of their favorite athlete.

4 Discussion

4.1 Deterrents to Malicious Challenges

While account recovery is a useful feature, it's also a target for malicious actors looking to steal accounts, by hackers, miners, or acquaintances.

In most cases, only the real account owner knows when they've lost their account credentials, and therefore when to make it is safe to make an Escrovery challenge. In a few cases like memory-affecting head injuries, coma, kidnapping, or death, it's possible that a third party will know that the user has lost their password before the user is able to submit the challenge. For example, an attacker could kidnap someone, then submit an Escrovery challenge, knowing the person is unable to respond in time. This is not unique to Escrovery, however; an attacker could already kidnap and physically interrogate a Bitcoin user to give up their secret keys. Nor is this even unique to cryptocurrencies, since an attacker that knows a person is in the hospital or kidnapped could burglarize their house.

Miners are in a privileged position to see an Escrovery challenge before putting it onto the blockchain. If miners knew the target account for a challenge, they could front-run an Escrovery for themselves ahead of placing the actual Escrovery challenge, thereby stealing the account. To prevent this, we use a two-phase commit and reveal process.

4.2 Escrovery is Decentralized, Non-Persistent, Anonymity-Preserving, and Secure

Escrovery is decentralized, non-persistent, anonymity-preserving, and secure, meeting all four conditions described in the introduction.

It is decentralized because there is no central authority or authorities involved in reclaiming ownership of an account.

It is non-persistent because either the accounts are naturally memorable (like decentralized usernames on Ethereum) or the accounts are tagged with secondary information that makes lookup possible.

It is anonymity-preserving because the new public key can be brand new, and there is no requirement that a user verify their real identity by attaching their phone number or email address to the account.

It is secure against common threat models, like offline rainbow attacks or guessing answers to challenge questions. While insecure against the threat of kidnapping, we argue this is not unique to Escrovery since a kidnapped user could be forced to give up their public keys anyway. It is also insecure against incapacitation or imprisonment for the duration of the waiting period, however for most users, this should be considered unlikely. High profile dissidents or journalists in totalitarian countries may be better off avoiding Escrovery.

4.3 Implementation

We envision three ways to implement an Escrovery mechanism. The first is by coding it directly into a blockchain protocol. For example, a new

cryptocurrency or a fork of an existing cryptocurrency could come with Escrovery built in. The second is with smart signatures, where the script encodes the process of Escrovery [9]. The third is with smart contracts, such as an Ethereum smart contract. We implemented such a proof-of-concept smart contract at github.com/pickhardt/escrovery.

4.4 Escrovery in Non-Trustless Systems

Although motivated by and most useful in trustless systems, Escrovery does not require trustless systems for its implementation. It is possible for centralized systems like Facebook or Twitter to implement Escrovery. For example, Twitter could allow anyone to pay $500 to initiate an Escrovery challenge against any account. If the account does not sign in within some time period, the account would be given to the new owner along with the $500. However, if the account owner does sign in within that time period, the $500 would be awarded to the original account owner. Since centralized systems already have established account recovery systems that work well, there is little benefit for them to use Escrovery for account recovery. This merely highlights that Escrovery does not require a decentralized or trustless system.

5 Conclusion

Escrovery has many advantages that make it ideal for trustless systems. To begin with, unlike most account recovery systems, it does not require a central authority. Further, anonymous accounts remain anonymous, since the challenge doesn't require verifying their identity, date of birth, phone number, or answering challenge questions. Finally, Escrovery checks can be made frictionless for the end user, if built into applications with high frequency of regular use like email, messaging apps, or social networks. Users would not have to remember any extra information like a backup username or passphrase.

Disadvantages include the time a user has to wait to recover their account and the amount they must temporarily place in escrow. Further, it does not actually protect against stolen passwords, just lost or forgotten passwords. If a password were actually stolen, and the user placed an Escrovery challenge, the new owner could still respond to the Escrovery challenge, since they now own the account.

More research is needed to determine how well Escrovery would work in practice. The costs, time durations, escrow payout fractions, user interface, and user awareness would all determine its effectiveness. Although this paper argues that attackers and malicious miners couldn't use Escrovery to steal active accounts, it is still an open question how well this would work in practice.

6 References

[1] J. Bonneau, E. Bursztein, I. Caron, R. Jackson, and M. Williamson, "Secrets, Lies, and Account Recovery: Lessons from the Use of Personal Knowledge Questions at Google," in Proceedings of the 24th International Conference on World Wide Web - WWW '15, Florence, Italy, 2015

[2] Chainalysis. "Bitcoin's $30 billion sell-off" [Online]. Available: https://blog.chainalysis.com/reports/money-supply

[3] S. Eskandari, D. Barrera, E. Stobert, and J. Clark, "A First Look at the Usability of Bitcoin Key Management," in Proceedings 2015 Workshop on Usable Security, San Diego, CA, 2015.

[4] M. Vasek, J. Bonneau, R. Castellucci, C. Keith, and T. Moore, "The Bitcoin Brain Drain: A Short Paper on the Use and Abuse of Bitcoin Brain Wallets"

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

[6] Business Insider, "A password for the Hawaii emergency agency was hiding in a public photo, written on a Post-it note" [Online]. Available: http://www.businessinsider.com/hawaii-emergency-agency-password-discovered-in-photo-sparks-security-criticism-2018-1

[7] M. Ali, R. Shea, J. Nelson, M. Freedman, "Blockstack Technical Whitepaper," [Online]. Available: https://blockstack.org/whitepaper.pdf

[8] H. Kalodner, M. Carlsten, P. Ellenbogen, J. Bonneau, and A. Narayanan, "An empirical study of Namecoin and lessons for decentralized namespace design"

[9] C. Allen, G. Maxwell, P. Todd, R. Shea, P. Wuille, J. Bonneau, J. Poon, T. Close, "Smart Signatures" Rebooting the Web of Trust. https://github.com/WebOfTrustInfo/rebooting-the-web-of-trust/blob/master/final-documents/smart-signatures.pdf