Collaborative Seed Recovery

A New Methodology for Smart Custody

By Christopher Allen, Shannon Appelcline & Wolf McNally of Blockchain Commons

The Problem: Seed Fragility

Seed loss is one of the biggest dangers when practicing self-sovereign control of digital assets. For years, the newspapers have been full of stories of lost hard drives[1], forgotten PINs[2], and other mistakes that have led to the potential loss of very valuable digital assets. The problem with self-sovereign control of digital assets is that you're the one in control, with no one to back you up. Single-points-of-failure, such as the loss of a key, can cost you everything.

To make self-sovereign control of digital assets truly work, especially in a world where your very identity might be a digital asset, we have to make self-sovereign custody *smarter*. We need to empower the average user to protect their digital assets' seeds in a way that's very easy, that doesn't expose single-points-of-failure, and that doesn't cause process fatigue.

The Possibilities: Multisigs & Secret Sharing

Cryptocurrencies currently offer three different methodologies where you can divide your secret into multiple parts, and then can access your assets using just *some* of those parts. This adds considerable resilience to your custody solution: though you now have more secrets (or shares of secrets) to track, you can also lose some of those secrets (or shares) without losing your assets. There are also numerous other advantages to these methodologies, the most notable of which is additional protection against compromise: using a standard setup for dividing up secrets, an attacker will need multiple secrets (or shares) to steal your assets.

These three methodologies are:

Multisigs. This is the standard methodology for locking digital assets with a number of keys. Some or all of them may be required to unlock the assets. When only some keys are required, which is an m-of-n multisig, then more resilience is created. It's ultimately a balance, as the fewer keys that are required to unlock an asset (reducing single-points-of-failure) corresponds to the fewer keys that are required to steal the asset (creating single-points-of-compromise).

Multisigs would be our preferred methodology for protecting seeds if not for some inadequacies in the modern landscape of digital custody. The biggest is that

[1] Max, D.T. 2021. "Half a Billion in Bitcoin, Lost in the Dump!". *The New Yorker. https://www.news18.com/news/buzz/uk-man-who-accidentally-dumped-bitcoin-worth-rs-3000-crore-is-still-looking-for-it-4560359.html.*

[2] Zetter, Kim. 2022. "Cracking a $2 Million Crypto Wallet". *The Verge. https://www.theverge.com/2022/1/24/22898712/crypto-hardware-wallet-hacking-lost-bitcoin-ethereum-nft.*

they're not supported by all blockchains. Ethereum offers the biggest barrier, as multisigs are not part of the standard custody paradigm for Ethereum (which more often depends on single master keys) and though they are usable through some smart contracts such as Gnosis[3], their use can be very expensive. Even on blockchains where multisigs are cheaply available, they remain hard to use. We recently authored a multisig self-custody scenario[4] and we believe the complexity of transferring data back and forth between multiple devices is beyond what most users are willing to do. Beyond this, current identity standards don't offer any support for multisigs, so the immediate future will only see identities protected by singular keys.

Secret Sharing. The other current option is to back up a secret with a secret-sharing scheme, where a single key is used to control the secret, but a backup of the key is sharded, such that the key can be reconstructed from some of those shares, but not necessarily all of them. The standard methodology to do so is Shamir's Secret Sharing[5], but Verifiable Secret Sharing[6] is another option that's quickly maturing and that offers new features.

Fundamentally, secret sharing schemes aren't as safe as multisig methodologies, because they rely on a single key that can be easily compromised. How to protect the key as it's reconstructed is a particular challenge[7]. However, secret sharing has notable advantages in that it can be much easier to use in the modern market and it doesn't depend on support from a blockchain. As a result, we currently believe that secret sharing is a prime technology for improving self-sovereign custody, where a user controls his own digital assets in a decentralized way.

(There are also possibilities to combine multisigs with secret sharing, as is done in the Blockchain Commons multisig self-custody scenario, but since those necessarily embrace usability issues with multisigs, we're ignoring those as a separate option for the moment.)

Collaborative Key Generation & Usage. This is a next-generation solution, where machines hold individual secrets that can be used to generate a secret that may not exist in any form before the second that it's needed. This technology is still very new, though it's in use with architectures such as FROST[8] and

[3]Uncredited. Retrieved 2022. *Gnosis Safe. https://gnosis-safe.io/*.

[4]Blockchain Commons. 2022. "Multi-Sig Self-Custody Scenario". *GitHub SmartCustody Repo. https://github.com/BlockchainCommons/SmartCustody/blob/master/Docs/Scenario-Multisig.md*.

[5]Keyless Technologies. 2020. "A Beginner's Guide to Shamir's Secret Sharing". *Medium. https://medium.com/@keylesstech/a-beginners-guide-to-shamir-s-secret-sharing-e864efbf3648*.

[6]Feldman, Paul. 1987. "A Practical Scheme for Non-Interactive Verifiable Secret Sharing". *IEEE. https://ieeexplore.ieee.org/document/4568297*.

[7]Blockchain Commons. 2021. "The Dangers of Secret-Sharing Schemes V0.2.0". *GitHub SmartCustody Repo. https://github.com/BlockchainCommons/SmartCustody/blob/master/Docs/SSKR-Dangers.md*.

[8]Uncredited. 2020. "Flexible Round-Optimized Schnorr Threshold Signatures". *CrySP". https://crysp.uwaterloo.ca/software/frost/*.

Torus[9]. We don't believe it's mature enough for our usage, though any work on addressing the problem of self-sovereign key fragility must future-proof their architecture to allow it to support Collaborative Key Generation & Usage.

The Development: Our Work So Far

Because of the current advantages of secret-sharing over multisig for solutions that support a variety of blockchains, secret-sharing has been a focus for our Smart Custody work in the last few years. The following articles from recent Web of Trust events detail some of our additional thinking over the years:

- **"A New Approach to Social Key Recovery"** (RWOT8 Advance Reading)[10]. An investigation into creating multi-level secret-sharing schemes to reduce the possibility of collusion and thus compromise.
- **"Publicly Verifiable Split-Key Schemes for Hybrid Secret Sharing and Multisig Authorization"** (RWOT9 Advance Reading)[11]. A discussion of the connection between social-key recovery and multisig schemes.
- **"Evaluating Social Schemes for Key Recovery"** (RWOT8 Finished Paper)[12]. A rubric for evaluating social-key recovery methods.
- **"Shamir's Secret Sharing: An Overview"** (RWOT8+9 Unfinished Paper)[13]. An unfinished (but extensive) paper from RWOT8 & 9 that discusses best practices, terminology, and more.

The work on secret-sharing schemes at RWOT 8 & 9 has also led to Blockchain Commons' development of an open-source secret-sharing library called SSKR[14]. It currently supports Shamir, but we hope to expand it to VSS in the future. It has also received a security review[15].

---

[9] Uncredited. Retrieved 2022. *Torus Website. https://tor.us/*.

[10] Allen, Christopher and Mark Friedenbach. 2019. "A New Approach to Social Key Recovery". *GitHub RWOT8 Advance Readings. https://github.com/WebOfTrustInfo/rwot8-barcelona/blob/master/topics-and-advance-readings/social-key-recovery.md*.

[11] Friedenbach, Mark and Christopher Allen. 2019. "Publicly Verifiable Split-Key Schemes for Hybrid Secret Sharing and Multisig Authorization". *GitHub RWOT9 Advance Readings. https://github.com/WebOfTrustInfo/rwot9-prague/blob/master/topics-and-advance-readings/verifiable-secret-sharing.md*.

[12] Gilligan, Sean, Gregory Jones, Adin Schmahmann, Andrew Hughes & Christopher Allen. 2019. "Evaluating Social Schemes for Key Recovery". *RWOT8. https://github.com/WebOfTrustInfo/rwot8-barcelona/blob/master/final-documents/evaluating-social-recovery.pdf*.

[13] Allen, Christopher, Bryan Bishop, Mark Friedenbach, Chris Howe, Andrew Kozlik, and Daan Sprenkels. 2021. "Shamir's Secret Sharing: An Overview". *Google Docs. https://docs.google.com/document/d/1rZJlFZcftrCM_KaxFnHUIskJKlSQzF0zFn4WIRQGDLU/edit#*.

[14] McNally, Wolf and Christopher Allen. Retrieved 2022. "Blockchain Commons SSKR". *GitHub Repo. https://github.com/BlockchainCommons/bc-sskr*.

[15] Radically Open Security. 2021. "Code Audit Report v1.1". *GitHub Repo bc-sskr. https://github.com/BlockchainCommons/bc-sskr/blob/master/reviews/2021-security-review.pdf*.

The Solution: Collaborative Seed Recovery

Producing secure code for secret-sharing is just one step in resolving the core problem of key fragility. As shown in part by past articles, there's also a need to consider both human usability and the best practices for actually using secret-sharing schemes. Blockchain Commons is now working on a new smart-custody solution that we call Collaborative Seed Recovery, which we hope will answer some of these questions.

The key features are:

- Use of Shamir's Secret Sharing to shard secrets.
- New capabilities to protect larger amounts of data than Shamir can natively, thanks to use of "Envelopes", which encrypt any amount of data through a symmetric key, which is what is actually sharded.
- Use of platform cloud backup system as a storage locale for the first share as well as to hold metadata about where other shares are.
- Design of share-servers, which hold shares as part of an ecosystem of secret sharing.
- Automation so that the user doesn't initially have to worry about the secret-sharing methodology at all.
- Ability for the user to later choose which share servers he prefers to use to protect his digital assets.
- Support for a variety of authentication methods on share servers, to ensure that shares are only returned to the authorized user and that the seed is protected during the vulnerable time of reconstruction.

We expect the standard storage workflow to be as follows:

1. User uploads secret data to a CSR app.
2. Data is encrypted with a symmetric key.
3. Symmetric key is sharded.
4. Envelopes are constructed, each containing encrypted data and one share.
5. First share is stored in cloud backup associated with app's platform.
6. Other shares are stored with random seed servers.
7. Information on share locations is stored with first share.

We expect the standard reconstruction workflow to be as follows:

1. User initiates reconstruction.
2. CSR app retrieves first share from platform cloud backup.
3. CSR app learns where other shares are from platform cloud backup.
4. CSR app retrieves sufficient shares from other share servers, asking user to authenticate as required.
5. Seed is reconstructed.

More information is available from our *article* describing CSR[16] as well as

[16]Blockchain Commons. 2022. "Collaborative Seed Recovery (CSR)". *HackMD.* [*https://github.com/BlockchainCommons/Gordian/blob/master/Docs/CSR.md*).

our videos on crypto-envelope, the foundation of CSR (the "crypto-envelope" presentation from July 15 is the first). Our *poster* also overviews much of this information.

We believe this approach meets the needs of usability that are currently absent from other solutions and that careful design of seed servers can ensure that best practices are met. Blockchain Commons is currently working with several companies to turn the CSR architecture into a reality. We welcome additional comments and development of this architecture here at RWOT11.

Related Topics

We also welcome discussions of related topics that have also been posted as RWOT11 advance readings:

Self Custody Risk Analysis

- by Eric Schuh, Legendary Requirements, USA
- A software framework to enable the choice of how to self custody digital assets
- #recovery #wallet #threat-model

Social Wallet Recovery

- by Timo Glastra, Animo Solutions, The Netherlands
- Social recovery of wallet data and keys by leveraging sharding.
- #recovery #wallet #sharding