Dark Crystal Web3 is a proposal sourced from an initial post by Vitalik Buterin in 2021, that Magma Collective is currently working on developing into an MVP. It's a social recovery protocol for which we identified three main important affordances: The app doesn't require the recovery partner to remember anything except the access information for their main ethereum wallet. The app should be secret-agnostic, meaning it can backup any kind of secret, including but not limited to an ethereum private key. The app should ideally allow for both the anonymity of the secret owner AND the recovery partners. Because the app is handling secrets, and perhaps secrets with a high value, validating the security assumptions and requirements for trust in our project was crucial to examine.

As part of our initial phase of MVP development, we undertook user research with core tools and protocols in the Web3 and Ethereum communities, conducting interviews with representatives of WalletConnect, 3Box, Metamask, Gnosis Safe, Rainbow and Tally as well as circulating an anonymous survey on Twitter and Discord. Our goals were not only to reflect on the usability of our project, but also the current thinking of important protocols on social recovery, and potential needed tooling. What we learned was endlessly fascinating, and summarized here for the benefit of the community.

Summary

- The role of cold storage
- How technical are users?
- What kind of "identities" are already in-use on chain and preferences regarding anonymity
- Role of onchain vs centralized storage
- Most important roadmap considerations

The role of cold storage

Besides the significant confirmation that usability should be prioritized, and a number of proposals for what that might include, perhaps our most significant finding is that the cold storage user is rare. Cold storage refers to keeping a key in a way that it is never connected to the internet – this could include a paper wallet, or a hard drive that is only connected to offline computers. It's widely considered the most secure way to store a secret key. However, transacting with a cold storage key presents significant usability hurdles: usually a wallet program must be downloaded and built on an offline computer, which packages and signs the transaction, which is then transferred to and broadcast from another device. Since our app was handling secrets, we wanted to know how likely this kind of process would be for our users?

Perhaps surprisingly, we found this kind of security practice was not common. Or more accurately, the overlapping subset between keys in cold storage and keys users would back up with our app is very small. In all our synchronous conversations, we heard that though it was important to support the cold storage pathway, this person would not use it themselves. From interviewees who did have a cold storage key, we heard that they would only use a system like ours for smaller-value hot wallet keys. Survey respondents were somewhat more favorable to cold storage – with 33% saying they would require their key to stay in cold storage at all times. With a longer research timeline, it would be useful to discuss with this user segment further, and discover how they differ from our interviewees. We also received feedback that the most security conscious users likely had their own backup systems, and would be unlikely to trust ours. Considering instead the person who needed social recovery the most was non-technical: "Anyone who can use a command line doesn't need social recovery."

How technical are users?

Indeed, what kind of technical literacy we could require from the user was another area of discussion – what tools, wallets, etc are familiar or required by our potential users? Architecturally speaking, Dark Crystal Web3 involves a hosted browser-based dapp, and a rust command-line component with a web frontend that can be served locally. Both our survey respondents and interviewees were very technical, with 80% saying they would be comfortable using a command line application if necessary. The considerations that led to our current architecture plan were wrapped up in the desire to support cold storage, but based on feedback we suspect that if a fully hosted version of our app was available, only a smaller subset of significantly motivated users would use it any other way. The feedback was so strong that, though we've worked from cold storage to hosted in terms of design and execution, it's worth considering that it perhaps should be the opposite.

Though we have targeted and mostly discussed a browser-based dapp as the main point of entry, we received strong feedback from 2 of 6 interviewees that a downloadable app was a more trustworthy and preferable design pattern, due to the frequency of phishing attacks. Overwhelmingly tools like Metamask, Walletconnect, mobile wallets, and block explorers presented no problems for interviewees or survey respondents. Though we have attempted to keep the protocol agnostic to what kind of secret is being backed up, we believe all of our potential secret owners are fully onboarded to Ethereum. We heard explicitly many times that a user who is not already an Ethereum user would be unlikely to choose our social recovery protocol. That said, these users expressed the desire to have much less technical recovery partners, who perhaps did not yet have wallets. There may be a useful feature to be considered here.

What kind of "identities" are already in-use on chain and preferences regarding anonymity

One of our core questions was how to support anonymity of the participants: how much is required, and how to maintain anonymity while also being able to find the recovery data reliably later? There are a few potential solutions, all involving storing the recovery data with some kind of lookup key, which could be hashed or determined from a user's main wallet. However, we wondered if user usage patterns would support this. Do people keep a persistent wallet, or do they derive many from the same seed and switch often? Do they generate new seeds? All in-person calls confirmed that our typical user would have a "main ethereum address" that they would be unlikely to forget under any circumstances. Interestingly however, half of survey respondents said it was possible or certain that the user had forgotten this, or never had one. The frequency of the "main ethereum address" as an identity-concept is a potential area for future research.

In general, feedback was very favorable for anonymity, but we heard repeatedly that once a significant threshold of users were part of the app, the "needle in a haystack" version of anonymity would be sufficient in many cases. How to bootstrap until this period is reached is not clear, or specifically which threshold of use would enable this. We discussed with several interviewees whether there was a public protocol for mapping ethereum addresses to public encryption keys that might provide a starter pack for this kind of anonymity, but have not identified an existing project that could fill this role. We also found that anonymity of the recovery partners was rated more strongly than anonymity of the secret owner, as seen below:

[

901×441 14.5 KB

](https://ethresear.ch/uploads/default/original/2X/e/e1dee1ab89145b94aa658890912aea868bd860f5.png)

Three survey respondents raised the concern that they might potentially forget who the recovery partners are, and we also received the feature request to be able to change recovery partners several times.

Role of onchain vs centralized storage

We received criticism in two cases for storing the encrypted secret onchain, one coming from the perspective of cost, and one coming from concern over the quantum threat to cryptography. It is certainly true that if any of the cryptography we use is someday broken, all secrets stored by our protocol would be compromised. In terms of cost, both survey respondents and interviewees consensed around $200 as an upper bound for the app's transaction costs, with lower being desirable. We heard that, for our use case, there is no advantage to being on mainnet over another ethereum-like network, such as polygon or gnosischain. Finally, it was suggested that we consider a centralized pathway for the app, where a user could opt out of onchain storage, which would of course present other potential problems (ongoing maintenance, potential for leaks) but might still meaningfully increase accessibility. 37% of survey respondents said they would feel more comfortable with offchain storage. Relatedly, long term maintenance of the protocol was flagged as a concern several times: how can potential users trust that the recovery process will still work far into the future, after our team has potentially disbanded or significantly changed? Relatedly, is there any way that the secret owner can verify that the secret is recoverable, without going through the whole process?

Most important roadmap considerations

One of the important aspects of secret recovery that we have so far descoped from MVP is the need for communication between the secret owner and the recovery partner. The initial set up information, as well as recovery data will need to be sent between parties. This is most likely the biggest security weakness of the system. 100% of survey respondents believed it was possible that some set of secret shards would be leaked by insecure out-of-band transfer, our in-person calls echoed this belief strongly. Because so many of our interviewees were protocol developers, we asked them about encrypted wallet-to-wallet messaging or existing roadmap features that might help communication. A closely related set of features has to do with notifications for secret owners and recovery partners at various stages of the recovery process. Many of the wallet developers we spoke to did have this on their roadmap, though without confirmed release dates. Further work in this direction seems very important to the ecosystem and integrating with one of these solutions is most likely central to Dark Crystal Web3 in the long term.