# Decentralized Storage of Crypto Assets via Hierarchical Shamir's Secret Sharing

Max Skibinsky[†], Dr. Yevgeniy Dodis[†],
Terence Spies[†], Wasim Ahmad[†]

January 2018

**Abstract**

Our most precious digital assets today include cryptocurrencies and the security of these assets is already at an unprecedented threat level. To protect these assets, we need a new cryptographic security platform − one that does not leave security centralized in a single place, with a single person, on a single device or in a single organization.

In this paper, we describe a new *cryptostorage platform* that is based on hierarchical Shamir's secret sharing. This platform enables new security and storage models, which are designed specifically for owners of crypto assets. We will show how decentralized cryptography and a decentralized storage network come together to form infrastructure that protects cryptocurrencies with full owners control, complete privacy, reliability and high availability. Owners of crypto assets can quickly setup distributed *Vaults* that are highly resilient and impenetrable to attacks on any part of the cryptostorage infrastructure. To power the *cryptostorage platforms'* day-to-day operations, we introduced a new *ERC20* token, *Vault Guardian Token* (VGT), which is designed to incentivize all actors in the platforms' ecosystem.

## 1. Introduction

In a world where our most precious digital assets include cryptocurrencies, the security of these assets is at an unprecedented threat level. Whether it is natural disasters, organized crime rings, malicious governments, hacked accounts or the

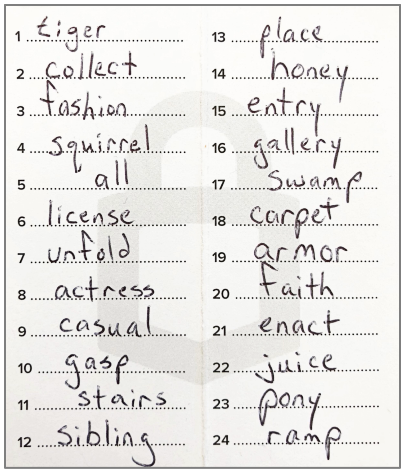
[†] founders@vault12.io

very fabric of our computer chips[1], the default security of our digital landscape is broken. And yet, cryptocurrencies themselves are just at the beginning of a new, world-wide shift towards empowering individual ownership secured by fundamental cryptography. In a few years, it won't be just cryptocurrencies. It will be house keys, car keys, real estate titles and all sorts of personal property that are secured by cryptographic keys. In this paper, we will introduce a radical new approach for storing all current and future, high-value cryptographic assets.

## 1.1 New Security Model for Crypto Assets

We routinely store our cryptocurrency assets, for day-to-day use, in hardware/software wallets or in a centralized, online account. However, both of these approaches have significant weaknesses.

In the case of wallets, if the owner loses the device, he risks permanent loss of all the assets[2]. Wallet vendors provide 12/24-word passphrases as means of "last resort" backup. Owners store these passphrases as pieces of paper — sometimes in a bank safe deposit box, sometimes in a sock drawer. That is a pretty odd medium to use for the highest level of security for digital money. This approach not only shifts the risk of attack on keys into the recovery phrase[3], but it is also subject to a whole new set of risks – earthquakes, burglaries, mudslides and fires to mention a few factors in California. The paper passphrase, unfortunately, becomes an unacceptably risky single point of failure.



1 tiger
2 collect
3 fashion
4 squirrel
5 all
6 license
7 unfold
8 actress
9 casual
10 gasp
11 stairs
12 sibling
13 place
14 honey
15 entry
16 gallery
17 swamp
18 carpet
19 armor
20 faith
21 enact
22 juice
23 pony
24 ramp

In the case of centralized online accounts, all of the account holders are at catastrophic risk when dedicated criminal hacker organizations target the online

[1] https://thehackernews.com/2018/03/amd-processor-vulnerabilities.html

[2] http://www.wired.co.uk/article/bitcoin-lost-newport-landfill

[3] https://www.wired.com/story/i-forgot-my-pin-an-epic-tale-of-losing-dollar30000-in-bitcoin/

storage providers, which was the case with *MtGox* (2014)[4], *Bitstamp* (2015)[5], *Bitfinex* (2016)[6] and *Coincheck* (2018)[7]. The delegation leaves users without direct ownership of their funds, negates one of the key principles of crypto assets and effectively forces users to fully *trust*[8] a centralized storage vendor. That relegates crypto assets to same outdated setup as the legacy banking system.

As cryptocurrency investors start to invest in more and more coins, the number of accounts and wallets needed will also increase. Given the threat landscape, protecting this currency, whether for the short term or the long haul, is critical. A better solution is needed to prevent more high-profile hacks and the subsequent loss of assets — some of it irretrievably.

## 1.2 The New Platform

To address these challenges, we introduced a new *cryptographic security platform* that does not leave security centralized in a single place, with a single person, on a single device or in a single organization. This new platform offers a decentralized way to create redundant security around the globe. It protects against hackers, organized crime and hostile government actors, including future ones that might be equipped with quantum computers.

Even with the new cryptography, the question remains – where to store the bits and bytes that physically represent crypto assets. What is needed is a new mechanism for storing digital crypto assets that can be redundant and highly resilient as well as be set up quickly and impenetrable to attacks on any part of the storage infrastructure.

[4] https://medium.com/@jimmysong/mt-gox-hack-technical-explanation-37ea5549f715

[5] https://arstechnica.com/information-technology/2015/01/bitcoin-exchange-bitstamp-claims-hack-siphoned-up-to-5-2-million/

[6] https://www.theguardian.com/technology/2016/aug/07/bitfinex-exchange-customers-receive-36-percent-loss-tokens

[7] http://time.com/money/5123018/coinchec-nem-hack-how-the-hackers-pulled-it-off/

[8] Szabo, Nick (2001). "Trusted Third Parties Are Security Holes"

## 1.3 Decentralized Cryptostorage

> *"The Winklevosses came up with an elaborate system to store and secure their private keys. They cut up printouts of their private keys into pieces and then distributed them in envelopes to safe deposit boxes around the country, so if one envelope were stolen the thief would not have the entire key."*
>
> *"How the Winklevoss Twins Found Vindication in a Bitcoin Fortune"* [9]
> by Nathaniel Popper, New York Times, December 19, 2017

Imagine if crypto owners could protect their cryptocurrency assets with encryption that cannot be cracked – even by a quantum computer. Imagine if they could store crypto assets in a storage system that is not located on any cloud server, but only on a distributed mesh of devices that they directly control. Imagine if owners could split their assets up in the way the Winklevoss brothers[10] have, but with all the convenience of modern, mobile, digital tools.

[9] https://www.nytimes.com/2017/12/19/technology/bitcoin-winklevoss-twins.html

[10] Winklevoss Capital is an investor in Vault12, Inc.

# 2. Technical Overview

We have introduced a fully-private, self-managed and highly-reliable crypto asset storage system that uses an approach invented by Adi Shamir[11], one of the world's foremost cryptographers and co-inventor of the RSA algorithm. The cryptographic algorithm of *Hierarchical Shamir's secret sharing*[12]*(H3S)* combines the personal control and complete privacy of self-managed cryptostorage with the reliability, high redundancy and elimination of a single point of failure associated with delegated storage.

## 2.1 Key Concepts

The core of our approach harnesses each crypto owner's trusted circle as a mesh network of mobile devices for distributed storage and, at the same time, as a method of social verification of each owner's identity.

The mobile devices of an owner's most trusted friends and family form an encrypted, distributed and decentralized storage network, which stores Shamir's secret sharing[13] *shards* that contain crypto assets key values — private keys, recovery phrases or any other critical artifacts. With this approach, we retain privacy and direct control of self-ownership with the high redundancy of distributed storage.

- Shamir's secret sharing allows any critical data (such as crypto asset keys) to be split into *m* number of *shards*, out of which *n shards* are required to restore the data. This is generally referred as *n/m* schemas[14]. Each *shard* does not disclose any information about the underlying asset, so potential compromises of individual devices are harmless. In fact, threshold cryptography is provably secure, because if an attacker obtains *n-1 shards*, even with unlimited computational resources, the attacker cannot decrypt any data from these *shards* — even with quantum computational resources.

[11] https://en.wikipedia.org/wiki/Adi_Shamir

[12] Tassa, Tamir. *"Hierarchical Threshold Secret Sharing"*

[13] Shamir, A., *"How to share a secret"*: http://cs.jhu.edu/~sdoshi/crypto/papers/shamirturing.pdf

[14] For example, *3/7 schema* means that we need at least *3 shards,* out of *7 shards* we created initially, to decrypt the secret.

- A larger number of *shards* provides redundancy and resilience of storage layer, since any *n shards* will be sufficient to recover the original data of our larger set of the m *shards* that were initially created. We will typically deal with policies in a range of 3—4 *shards* out of 5-10 created. Policies that require extreme redundancy can be created as *10/100* — even *20/200* are fully supported by the *cryptographic security platform.*

- The encrypted file storage inside an uncompromised iOS mobile device is one of the most secure digital storage options in existence today. External access, to a locked iOS phone internal storage, is close to impossible.

- Close friends, family and colleagues are in the best position to authenticate owner's real identity and would be the hardest audience to fool using any social impersonation attack.

- Hierarchical extension, using *hierarchical secret sharing*[15]*,* enables us to define roles of each device. For example, some operations are only possible with the participation of specific devices.

We define three tiers of trust as follows.

1. The highest tier is the *owner* himself. We will refer to *owners* when we talk about social verification of identity and *master device* when referring to *owner's* phone.

2. A next tier is a group, selected by the *owner*. We call them *Guardians.* They are family members and best friends who deserve a high degree of trust. An *owner* can also add his additional personal mobile devices as *Guardian* devices[16].

[15] Tassa,Tamir. *"Hierarchical Threshold Secret Sharing"*

[16] The only downside of adding more personal devices is that, if they are co-located in the same location, they will be subject to same risks in case of fire or earthquake. Distributed social *Guardians* provide more robustness against natural disasters.

3. The last tier consists of the *owner's* casual connections – a large number of acquaintances that the *owner* can easily contact, yet who do not have any implied strong trust. That tier provides additional redundancy and availability, since the number of weak connections will greatly outnumber high-trust *Guardians.* We will call all of the *owner's* connections involved in safeguarding his assets *Custodians.*

The typical process of securing crypto assets works as follows. *The owner* recruits his *Custodians* - a few trusted *Guardians* and a number of casual friends - whose phones become distributed storage nodes. Then he opens a *Vault* of distributed data storage powered by these nodes. The app on the *master device* converts critical data into threshold schema *shards*, which are then encrypted using master device PKI keys. These are immediately distributed to the *Custodian* devices for permanent storage. Once *shards* are transmitted out, the application verifies storage of the *shards* in a trusted circle and monitors the health of the *Vault* over time. An active *Vault* can dynamically generate new *shards*, if the *owner* invites new *Guardians* and increases the size of *Guardian* circle.

When the *owner* unlocks his *Vault*, he uses social verification to request threshold *shards* from *Custodian* devices. *Owners* can opt in to use personal U2F, HSM or biometric devices when authenticating *shard* requests with *Custodians.* After collecting a threshold number of *shards*, the *master device* reconstructs crypto asset data allowing the *owner* to access and use them. Depending on policy selected during *Vault* creation, one or more *Guardian* devices are required by hierarchical schema to complete reconstruction.

Private keys, setup on the *master device*, are also automatically *sharded* into the *Vault*, but these *shards* are sent only to *Guardian* devices. If the *owner's master device* is lost or destroyed in an accident, the *owner* can contact his *Guardians* for higher-level social verification and request *restoration shards.* This would allow him to restore *master device* setup on a new phone. *Custodian* devices play the role of passive, extra storage that increases the number of backup *shards* for this critical information, yet they do not have the *Guardian*-level hierarchical *shards* to restore that data nor encryption keys to decrypt each *shard.*

Finally, our method offers maximum protection to the *owner* if his *master device* is lost or stolen. The *master device* itself stores no crypto assets. Therefore, there is simply nothing of value to recover from the *master device* even if internal storage is somehow penetrated. Any unauthorized attempt to unlock or restore operation will require impersonating *owner* on voice or video call to some *Guardians* who know the *owner* very well and can immediately warn the *owner* if somebody is attempting an unauthorized operation with the stolen phone.

## 2.2 Design Goals

**Full Decentralization:** The platform should minimize dependency on central authority or central services to the maximum extent possible. The platform should be fully usable by any individual who simply installs or runs the open source software – similar to the operations of the decentralized network of Bitcoin full nodes.

**No Single Point of Failure:** Multiple devices in the *owner's* trusted circle provide storage redundancy and geographical separation. The *owner* should be able to easily restore the *master device*, if it is lost or stolen. Destruction of the majority of *Custodian* devices does not lead to any loss of funds. We physically distribute H3S *shards* across multiple devices to create extreme robustness of storage.

**Scalability:** Additional storage devices bring extra redundancy and reliability and secret sharing platform can easily handle hundreds of devices. This is a natural property of threshold secret sharing that allows hundreds of *shards* to be generated.

**Incremental Access:** Assets encrypted with the less complicated H3S policy are easier to access than more sensitive documents. Full access to maximum security assets requires maximum social verification. We achieve this goal by setting up different storage policies of hierarchical secret sharing as described below.

**Dynamic Membership:** The *owner* should add/remove *Custodians* at will without impacting stored documents. This is a natural property of threshold secret sharing, which allows the *owner* to generate new *shards* by following same policy created at the beginning of the H3S process.

**Full Privacy:** The trusted circle should have no visibility to any of *owner's* assets, nor should they see any changes during the lifetime of secured assets. This is a critical differentiator when compared with multi-sig, co-ownership of crypto assets – where every co-signer has full visibility to the amount of funds and transactions passing through the multi-sig address. Full, theoretical security of H3S ensures that any number of *shards* below the threshold value does not provide any information about the content of assets they are securing.

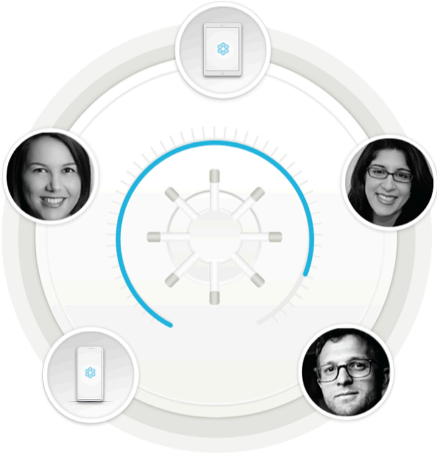
**Social Verification:** The *owner* can enforce the policy unlocks higher-level security policies. However, one or more *Guardians* must verify *owner's* identity. This ensures that access to the secured asset is cryptographically impossible without *Guardians* approval. This is a natural property of adding hierarchical tiers to threshold secret sharing, as we demonstrate below.

**Hardware Verification:** The *owner* can opt-in to use U2F, HSM or zero-knowledge biometric devices as a verification channel when proving his identity to *Custodians.* This provides a faster verification cycle for *owners* willing to invest in the purchase and setup of additional hardware. In this document, we assume social verification is the default option available to all *owners*, which they can augment or replace with U2F/HSM hardware.

**Collusion-Proof:** Bad actors, among lower trust *Custodians,* should not be able to restore assets by themselves – even if they form a group to perform a collusion attack. It should be cryptographically impossible for them to restore any secrets using only the *shards* from the non-*Guardian* storage tier. This is natural property of adding hierarchical layers to plain threshold secret sharing, as we describe in the H3S policies below.

# 3. Trusted Circle of Guardians

The *owner's* trusted circle consists of high-trust *Guardians* and low-trust *Custodians.* Adding Custodians is optional but recommended as they add additional redundancy to the storage system.



## 3.1 Fully-Trustworthy Guardians

Close friends or immediate family are *fully-trustworthy Guardians.* These would be a relatively small group of people (between 5—10) who are characterized by strong personal connection and permanent, long-term ties and who have a high level of personal trustworthiness, such as parents, direct siblings, BFFs, spouse, *et cetera.*

## 3.2 Personal Devices

*Owner's* additional personal devices (e.g., iPads, spare phones) can play the role of extra *Guardians* under the *owner's* personal control. This gives the *owner* immediate access to his assets. However, these devices are subject to the same risk as the *owner's master device,* since they are usually co-located in the same place.

## 3.3 Casual Friends

*Casual friends* are easy to reach via Facebook, Twitter and phone contacts. They include co-workers, random friends, etc. These would be 100–200 social contacts with a low level of trust.

## 3.4 Roles in Trust Circle

*Guardians* provide trust and social verification for the *owner.* The rest of the *Custodians* provide geographically distributed, massively redundant storage. The hierarchical sharing policy makes a friend-collusion attack impossible. This is because the information contained in *Custodian shards* is insufficient as it requires either the *master device* or *Guardians* to fully restore secret documents.

In normal operations, the *owner* can tap personal devices or any *Custodian* storage nodes (from hundreds available) to recover *shards* via social messaging, as needed. The abundance of storage nodes improves both storage redundancy and access speed. In the case of catastrophic failures, such as complete device destruction, the *owner* can recover critical information from his *Guardians* and rebuild his *Vault* on a new device.

# 4. Storage System

Secure storage of crypto assets, such as *Bitcoin*, *Ethereum* and other cryptography-based tokens, presents a unique challenge. In case of permanent loss of key values (such as private keys or derivation seeds), the assets become permanently unrecoverable and could result in a substantial financial loss. At the same time, creating multiple backups of key values creates multiple avenues of attack that can be exploited by malicious hackers and result in complete loss of crypto assets. Crypto assets storage requires strong redundancy, structured in a way that does not lead to increased risk of security compromise.

## 4.1 Legacy Cryptostorage Solutions

There are two mainstream methods of cryptostorage – personal self-storage of crypto asset by hardware/software wallets or delegation of storage and security responsibility to a central provider.

Creating personal cryptostorage, using software or hardware solutions, is technically challenging and is above the skill level of most non-technical users. Additionally, native self-storage setups often suffer from a single point of failure and are easy for sophisticated attackers to bypass.

The critical weakness of crypto wallets is that the loss of a physical hardware device or software wallet data is equivalent to losing the crypto assets stored on them. To address that risk, vendors provide standard 12/24-word recovery phrase. For all practical purposes, possession of the 24-word recovery phrase for the best hardware wallet is the equivalent to possessing all current and future funds managed by that wallet. This policy effectively shifts the risk from an attack on the keys to an attack on the recovery phrase, which is often stored as a piece of paper and is subject to all sorts of additional risks ranging from common burglary to fires and earthquakes.

Another popular option is the delegation of storage to an external provider. Unfortunately, such centralized storage solutions expose a large number of users to catastrophic failures, if the provider is targeted by a dedicated attack. The delegation leaves users without direct ownership of their funds, negating one of the key features of crypto assets. Delegation of storage effectively forces crypto asset *owners* back to the same outdated setup as the legacy banking system with an external provider serving the role of the crypto bank.

Our new cryptostorage model leverages commonly available mobile devices and can be set up quickly. It is also redundant, highly resilient and impenetrable to attacks. The *Vault cryptostorage* system is not located on a cloud server. It is located on a distributed mesh of mobile devices, which the *owner* carefully selects for his trust circle.

## 4.2 Storage Tiers

*The owner's* assets are each assigned a storage tier based on required security. In general, more secure assets require more social tax in order to be restored with implied stronger social confirmation of restoration request.

As an example, we can imagine a setup with four storage tiers, separated between one *quick access* tier and three long-term storage tiers.

- **Quick Access:** Stores low-value information that the *owner* needs to access frequently and that does not require any security. Assets are stored locally on the *master device* and are distributed across *Custodians* only for backup.
- **Easy Access:** Stores low-value hot wallets.
- **Secure Storage:** Stores wallet passwords and high-value wallets.
- **Ultra-Secure Storage:** Stores recovery phrases for hardware devices and cold wallets.

When we refer to all assets stored with the same policy, we will call these *storage containers.*

Now, let's map these policies to H3S *shards* for distributed storage.

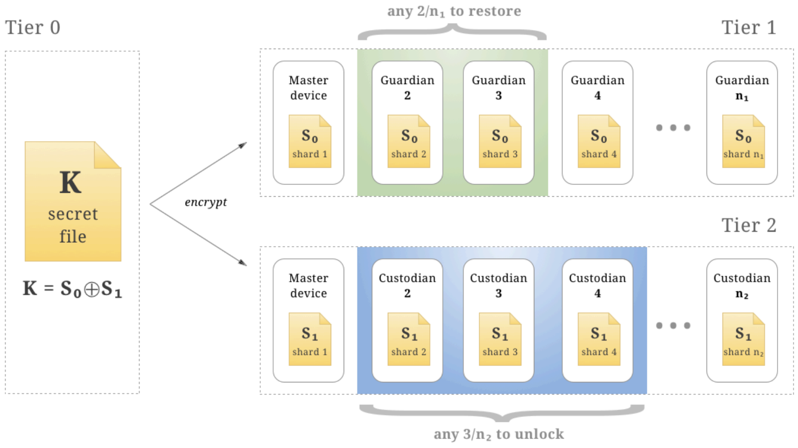
## 4.3 Hierarchical Shamir's Secret Sharing

Higher security policies have higher social requirements to be recombined. A common-sense requirement is to make restoration of higher-level secrets possible only with the participation of higher-trust tiers – *owner's master device* or *Guardians.* We will also pair high-order policy access with high trust confirmation methods, such as live phone calls or video chat.

To construct the tiered access policy, we use the hierarchical extension of Shamir's secret sharing.

Given *n* storage tiers and store secret document **S**, when required, we can generate *n-1* random and independent secrets $\mathbf{S_i}$ of the same size[17]. Then, we can define[18] $\mathbf{S_0} = \mathbf{S} \oplus \mathbf{S_1} \oplus \mathbf{S_2} \oplus \dots \mathbf{S_{n-1}}$ for every tier that must be present during the restoration of **S**. Afterward, each secret $\mathbf{S_i}$ is distributed between devices in a given tier using standard $(n_i,m_i)$ – Shamir's secret sharing[19]. Obviously, restoring **S** back will be a simple $\mathbf{S} = \mathbf{S_0} \oplus \mathbf{S_1} \oplus \mathbf{S_2} \oplus \dots \mathbf{S_{n-1}}$ operation, where $\mathbf{S_i}$ of each tier is restored from Shamir *shards*.

One way to think about this structure is that the logical operation between vertical storage tiers is a logical *AND*, while Shamir's *sharding* of each tier is logical *OR*. Vertical tiers require the presence of two or more secrets $\mathbf{S_i}$ from each tier for final restoration, while Shamir's $(n_i,m_i)$ *sharding* of specific $\mathbf{S_i}$ is horizontal *OR* operation that specifies a minimal threshold of *shards* required to restore that tier. Using *AND* and *OR* as our primitives allows us to create any *social security* policy for the *owner*, with next to unlimited vertical AND layers ($\mathbf{S_i}$ pads) and specific $(n_i,m_i)$ policy on each horizontal level (policy to *shard*/restore each $S_i$ pad).



*Example of 2-tier storage policy*

[17] These are one-time pads for each tier and same size as original document.

[18] ⊕ is XOR operation.

[19] $(n_i,m_i)$ schema means that $n_i$ shards (threshold value) are required for reconstruction of the secret data out of $m_i$ created shards of the tier i.

As a simple example, let's say we want a two-tier policy where secret file **K** is *sharded* over three *Custodians* (out of $n_2$ tier-2 *Custodians*), yet we want at least one of them to be a *Guardian* (out of $n_1$ tier-1 *Guardians*), and the *owner's* device is a *Guardian*[20] as well. Hierarchical *sharding* will give us the following algorithm.

- Create a random pad $\mathbf{S}_1$ of the same size as secret file **K**.
- Create masked pad as $\mathbf{S}_0$=$\mathbf{S}_1$⊕**K.**
- Tier 2: *Shard 3/*$n_2$ masked pad $\mathbf{S}_0$ and send these *shards*, $\mathbf{S}_{0i,}$ to all $n_2$ *Custodians* and *Guardians.*
- Tier 1: *Shard 2/*$n_1$ the pad $\mathbf{S}_1$; keep one *shard* $\mathbf{S}_{10}$ on the *master device*; and send the other *shard* $\mathbf{S}_{1j}$ to $n_1$*-1 Guardians.*

If we want to unlock file in this two-tiered schema, then the response of one *Guardian* is mandatory. If the response contains only *Custodians*, we can recombine three *shards* back to $\mathbf{S}_0$. Yet, it will be impossible to restore original key without the mask $\mathbf{S}_1$. As soon as one of the *Guardians* responds, we recombine the mask from two *shards* (using $\mathbf{S}_{10}$ local *shard* on the *owner's* device) and recover the original key by $\mathbf{S}_1$⊕$\mathbf{S}_0$. If the *owner's* device is lost, he can contact two *Guardians* to recombine $\mathbf{S}_1$ when restoring on a new phone. Then, he can reconstruct **K** by contacting one extra *Custodian.* (We already got two *Custodian*-tier *shards* from two *Guardians* who store one *shard* for each role.)

### 4.3.1 True Entropy for One Time Pads

Theoretical security of H3S is guaranteed only if the one-time pads, generated for each tier, come from a genuinely unpredictable entropy source. Unfortunately, that is especially hard to guarantee on an arbitrary mobile device. Additionally, random-number generator failures related to cryptocurrency use are especially catastrophic[20]. Although */dev/unrandom* device on iOS is one of the best, state-of-the-art CPRNGs on the market today, there is no way of knowing about currently undiscovered weaknesses in a mobile PRNG stack if any exist. Especially troubling, we cannot know about future weaknesses that might be introduced by buggy libraries or OS updates. As the name implies, one-time pads are for one-time use. If they are to be generated by weak PRNG, this weakness will persist as long as a given one-time pad is used.

[20] https://bitcoin.org/en/alert/2013-08-11-android

To radically safeguard all current and future one-time pads generated by our platform, we developed a new hardware random source that leverages omnipresent thermal noise in the iPhone camera. The simplified version[21] of this algorithm was open sourced[22] for cryptographic community review last year. This hardware source, unique for each individual device at the moment of entropy generation, is utterly unpredictable for external observers and is mixed in with regular entropy stream from */dev/unrandom* when generating one-time pads required by the H3S algorithm. Using this true physical phenomenon as part of overall entropy generation makes our pads truly unpredictable even in the case of future PRNG compromises.

### 4.3.2 Adding New Custodians

One property of threshold secret sharing is the option to generate new *shards* using threshold restoration of pre-existing *shards*. Formally, each Shamir's *shard* is the solution to polynomial at points *f(1), f(2), f(3),…* and the solution to *f(0)* is the secret itself. Adding new *shards* requires simply requesting threshold number of *shards* back. However, instead of solving for *f(0)*, we solve the polynomial for *f(n+1)*, which is a new *n+1 shard*. Because of that property, any policy can generate additional *shards* for each horizontal layer if the threshold number of *Custodians* at that layer approves the creation of new *shard*. In social terms, that means adding one more member into *Custodian* or *Guardian* circle, which is done by threshold approval of existing policy.

## 4.4 Creating Storage Policies

H3S allows us to create any configuration of *AND/OR* secret sharing matrix in any number of policies. Let's review how we can map four storage tiers specified to such a secret sharing matrix. We will refer to *n/m* threshold for that specific policy request – *n Custodians* to confirm access or restoration request out of *m* created *Custodians* for given policy layer. Confirmation channel refers to the method *owner* uses to send his access request and verify his identity to *Custodians*, such as phone or video call, SMS, email, instant messaging apps, Slack, Skype or any other direct communication tool.

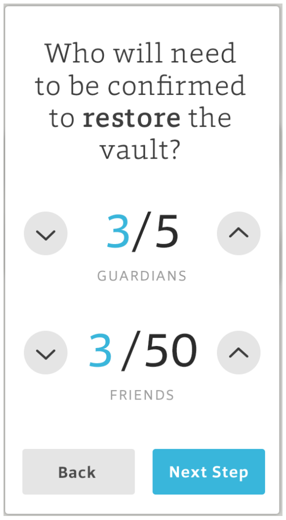
[21] Vault12, Inc. (2017). “How to get true randomness from your Apple device with particle physics and thermal entropy”

[22] https://github.com/vault12/TrueEntropy

### 4.4.1 Quick Access

Commonly, lowest security assets are stored on the phone locally and are instantly accessible to the *owner*. If the *master device* is lost with documents in the *quick access* container, the *owner* can recover them since *Custodian* devices will contain *shards* of that storage container. If a malicious party discovers the lost device before the restoration, these documents can only be accessed if malicious party succeeds in unlocking the phone. For *quick access* storage containers, H3S provides distributed backup hosted on other people's phones, but it does not provide any additional storage security.

*The quick access* container will provide the *owner* with the following storage profile.



Who will need to be confirmed to **restore** the vault?

3/5 GUARDIANS

3/50 FRIENDS

- **Maximum ease of use**: *The owner* can access any assets immediately with no social verification. The only protection of these assets is local, device-level encryption tied to the physical device's unlock process.
- **Social backup & restoration**: *The owner* can contact *Custodians* to restore his assets. *The owner* completely removes social confirmations in this policy, yet still keeps the benefit of *social backup* and redundancy on a MIST network. This allows him to restore his assets socially in case of device loss.
- **Maximum security risk**: Possession of the *master device* with unlock code is equivalent to the possession of all assets stored in *quick access* container.

This policy might be acceptable for *owners* who always keep the *master device* in a controlled, secured environment, such as hardened safe in corporation central office, yet need distributed social backup to protect from natural disasters. However, if the *master device* runs the risk of being lost or stolen in public places, such a device should be setup only with secured policies.

### 4.4.2 Secured Policies

- **Easy access**[23]: This is the first policy and it governs more sensitive assets with limited value, such as a small-denomination hot wallet for day-to-day expenses. It is the simplest of the storage containers to unlock socially. It requires only one *Custodian shard* in addition to the *shard* stored on the *master device.* This policy allows for weakest and fastest social confirmation channels, such as SMS, chat, IM and email.
- **Secure Storage**: This policy is for valuable assets, such as high-value wallets and wallet passwords. It requires that at least two *Guardians* must always authenticate user identity for any asset.
- **Ultra-Secure**: This policy is for top-secret documents/data, such as cold wallets and 12/24-word recovery phrases. It requires at least three *Guardians* to always authenticate user identity for accessing the data.

## 4.5 Mapping Policy to H3S

Now, knowing all of the containers' roles, we can formulate a few security policy presets using the *AND* (∧) and *OR* (∨) terms of H3S. We will use square brackets for grouping of sets that constitute full secret restoration.

We will call the act of sending a *shard* back to the *owner* a *social confirmation.* For weaker containers, they can be as simple as responding to a text or a chat message. For higher order containers, default policy will require a phone or video call before a *shard* is released.

We will construct a three-tiered system with the *master device* **D** as a single node, *Guardians* **G,** and remote connections **F**. *Custodians* **C** represent *Guardians* and remote connections combined[24]. We will refer to *quick access* container **QB** and storage containers for other policies as **B1-B3**.

[23] For this and any other storage policies, we always assume that *owner* can re-configure policy requirements to different *n/m* thresholds and different confirmation channels.

[24] Bear in mind that Custodians **C** never receive *shards* for private keys used to PKI encrypt all *shards* on *master device*, so they cannot participate in any collusion attack. *Guardians* can participate in a collusion attack by restoring PKI keys from their *restoration shards*, and then decrypting data shards with these keys. That's why it is crucial to choose good *Guardians.*

| Container | Policy | Properties |
|---|---|---|
| **QB** | [**D**] *OR* [3x**G**] | The *master device* stores these documents locally. If the device is lost, they can be restored from 3x*Guardians.* |
| **B1** | [D∧**C**] *OR* [3x**G**] *OR* [**G**∧3x**F**] | The *owner* needs one confirmation from any *Custodian* to recombine secrets. If the device is lost, **B1** documents can be restored from 3x*Guardians.* If only one *Guardian* is available, they can be restored by additional three *Custodians.* |
| **B2** | [**D**∧2x**G**] *OR* [3x**G**] *OR* [2x**G**∧5x**F**] | The *owner* needs two *Guardian* confirmations to recombine the assets. If the device is lost, **B2** assets can be restored from 3x*Guardians.* If only two *Guardians* are available, assets can be restored by additional confirmations of five *Custodians.* |
| **B3** | [**D**∧2x**G**] *OR* [3x**G**] *OR* [2x**G**∧10x**F**] | The *owner* needs 2x*Guardians* confirmations to access the container. If the device is lost, **B3** documents can be restored from 3x*Guardians.* If only two *Guardians* are available, seed can be restored by additional confirmations of 10 *Custodians.* |

Now let's consider different design forces acting on policy selection. For example, an *owner* prefers the maximum ease of use and prefers to avoid the full social friction of *Guardian* phone calls, while still enjoying reliability and redundancy of social storage. For such *owners*, an *easy access* policy might be configured like this.

| **Containers** | **Policy** |
|---|---|
| QB | [**D**] ∨ [3x**G**] |
| B1 | [**D**∧**C**] ∨ [3x**G**] ∨ [**G**∧3x**F**] |
| B2 | [**D**∧**C**] ∨ [3x**G**] ∨ [2x**G**∧5x**F**] |
| B3 | [**D**∧2x**C**] ∨ [3x**G**] ∨ [2x**G**∧10x**F**] |

In this policy, the *owner* needs only one or two confirmations from any *Custodian* using texts or emails during regular file access. The *owner* can, however, still restore his *Vault* from three *Guardians* or additional *Custodians*, verified via phone calls as required by restoration policy.

Obviously, any policy allowing restoration by *Guardians* and remote connections represents a collusion risk, if the *owner* chose the *Custodians* poorly and they actively conspire against him. An *owner*, wishing to eliminate the risk of *Custodian* collusion, might create an extra secure schema. An *Extra Secure*" policy might look like following.

| **Containers** | **Policy** |
|---|---|
| **QB** | [**D**] ∨ [5x**G**] |
| **B1** | [**D**∧**C**] ∨ [5x**G**] |
| **B2** | [**D**∧[**G**∧**C**]] ∨ [5x**G**] |
| **B3** | [**D**∧[2x**G**∧**C**]] ∨ [5x**G**] |

In this policy, the *owner* still gets the benefits of higher *Custodian* availability for **B1-B2** unlocks[25] while making five *Guardians* the only option to restore the *Vault*.

These sample schemas demonstrate that hierarchical storage policies will allow *owners* to configure a wide variety of practical policy selections based on which is best suited for a specific *owner's* needs. In general, we can provide any number of hierarchical or parallel *AND/OR* groups to satisfy specific policy requirements and different *owner* needs as well as construct crypto asset storage policy for any specific security profile.

## 4.6 Multi-device Setup

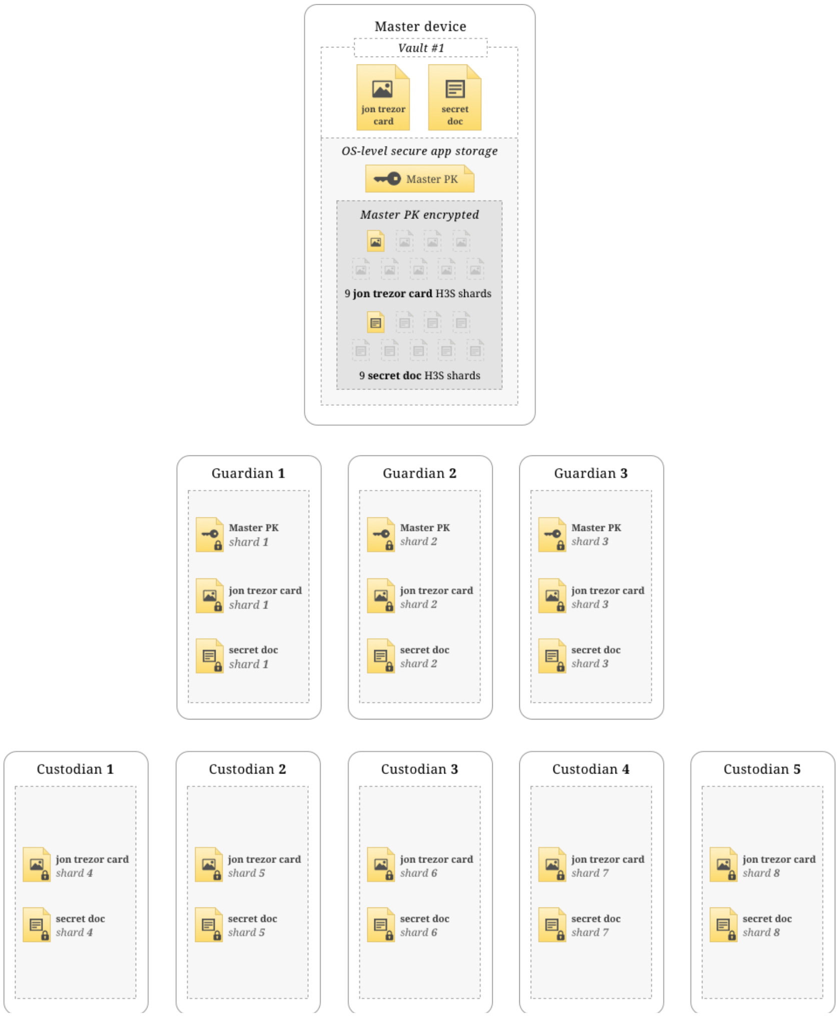
Users might utilize multiple mobile devices to reduce social requirements by adding multiple *Guardians*. Additional devices can also be added as independent *Guardians*. Replacing *Guardians* with *owners'* personal devices brings its own set of tradeoffs.

- **Convenience**: The *owner* can access assets from multiple devices right away without waiting for social confirmations.

[25] Since there are numerically more *Custodians* than *Guardians*, it should be quicker to find the required threshold number of *Custodians* online.

- **No geographical separation**: If the *owner's* main device is destroyed in a natural disaster event, the same event is likely to destroy all of the other *owner's* devices in the same location.
- **Device co-location**: If the *owner* keeps his mobile phone, laptop and tablet in the same bag, theft of the bag will give potential attackers access to all three devices.

When used with reasonable precautions, such as avoiding co-location of *owner* devices, the multi-device approach can reduce the number of social verifications needed to unlock crypto assets and increase the overall speed of operations.



*Example of distributed storage*

# 5. The Vault Cryptostorage Platform

Let's review our *cryptostorage platform* as a whole. The platform’s first priority is to serve cryptocurrency *owners*, who require sophisticated storage and security policies to safeguard their assets from catastrophic failures. The platform can also serve various organizations, such as law firms, corporations and investment firms that require full, distributed security for financial, legal and other kinds of critical documents and digital credentials. It enables anyone with a smartphone to create *cryptostorage Vaults,* containing high-value assets[26], which are safeguarded by the *owner’s* trusted people or devices – also known as *Guardians. Owner's Guardians* can be adjusted in real-time, so that the appropriate number is always guarding the *owner’s* cryptocurrency assets. No one can access or view *owner's* crypto assets and only the *owner* gets access, when needed.

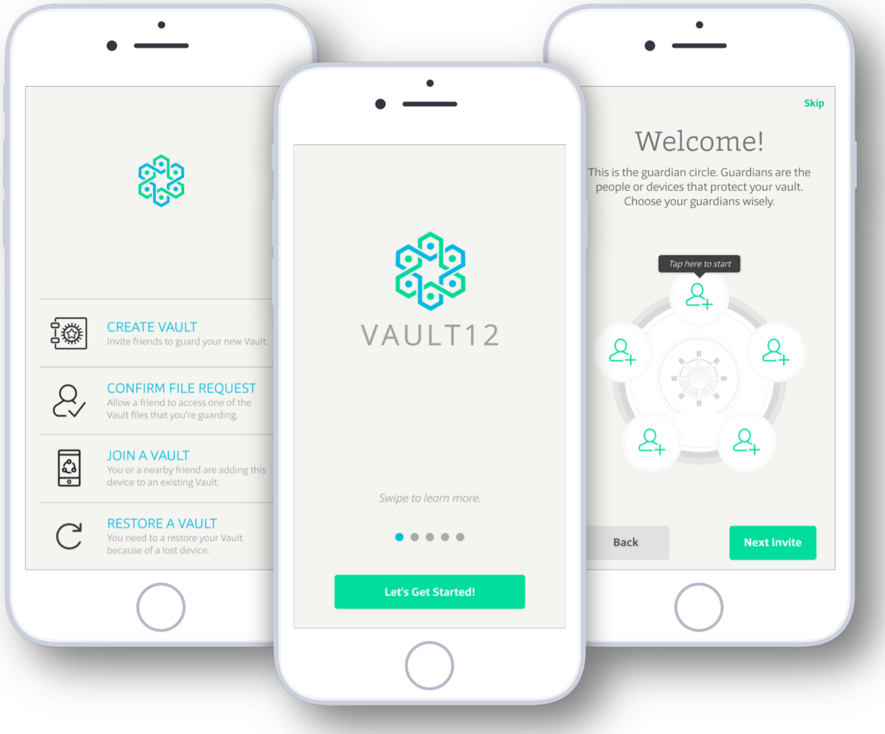
All data in the *Vaults* is secured using *Hierarchical Shamir's Secret Sharing*, which is so secure that not even an attack via a quantum computer can unlock protected assets. The only part of metadata exposed to *Custodians* is the filename, which is provided by the *owner* when storing his assets in the *Vault.* When asking *Custodians* for access, the *owner* refers to the asset by this filename. However, the internal content of any asset is absolutely opaque to any *Custodian* and is impossible for them to recover. *Owners* can keep their existing wallets and just add critical recovery information to their *Vault.*

## 5.1 Vault Platform Applications

Anyone can create *cryptostorage applications* using APIs exposed by the Vault *Cryptostorage platform's* open source software stack. As a reference implementation, we have built the first platform application to run on mobile devices — phones and tablets. The *Vault12* mobile app[27] can be used by *owners* and *Guardians* alike and creates the mesh networks used to store *owner's* cryptocurrencies. *Guardians* can create their own cryptostorage *Vaults* that will work side by side with whatever assets they are guarding. Any mobile device can be a *master device* for its *owner*, while also being a *Guardian* for any number of friends and family.

[26] That can take any form, including pictures of recovery phrase, QR codes, private keys, documents, files, et cetera.

[27] http://vimeo.com/vault12/vault12video



Invite friends to guard your new Vault.

Allow a friend to access one of the Vault files that you're guarding.

You or a nearby friend are adding this device to an existing Vault.

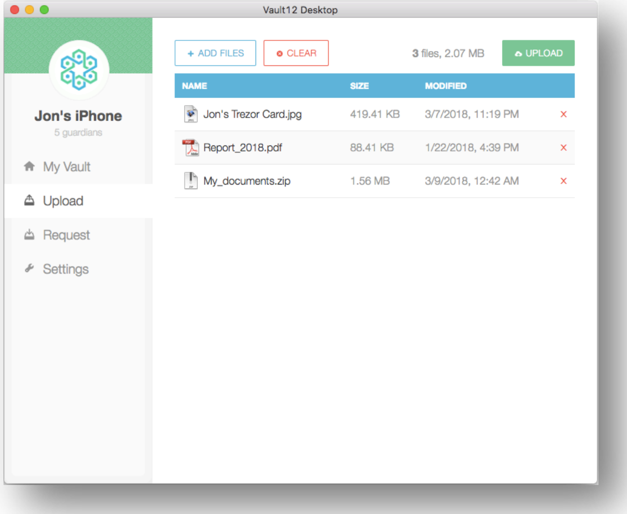
You need to a restore your Vault because of a lost device.

Swipe to learn more.

This is the guardian circle. Guardians are the people or devices that protect your vault. Choose your guardians wisely.

*User interface of Vault12 mobile app.*

*The owner* may need to secure or restore assets to/from their desktop computer. Our second platform application is the *Vault12 desktop client* for Windows and OS X. *The desktop client* pairs directly with the *master device* to establish a secure, unbreakable communications channel. Any asset can be securely *sharded* or restored into the *owner's Vault* using any number of paired desktop clients.



3 files, 2.07 MB

| NAME | SIZE | MODIFIED |
|---|---|---|
| Jon's Trezor Card.jpg | 419.41 KB | 3/7/2018, 11:19 PM |
| Report_2018.pdf | 88.41 KB | 1/22/2018, 4:39 PM |
| My_documents.zip | 1.56 MB | 3/9/2018, 12:42 AM |

*Vault12 desktop app pairs directly with the* master device *to establish a secure, unbreakable communications channel.*

## 5.2 Vault Cryptostorage Platform Stack

### 5.2.1 MIST Network

*The Mesh Information Storage (MIST)* Network automatically tracks the storage and health of all *shards* of all information stored in *cryptostorage Vaults.* If any *shards* are lost (for example, if unmotivated *Custodian* deletes the app), it will warn the *owner* that redundancy of storage of for a specific *Vault* is decreasing. Each *MIST* network is created individually, for each cryptostorage *Vault,* and is only accessible to its *owner.* After the *owner's* files are *sharded* into the *Vault,* there is no single device (and therefore no attack vector on the device) that contains a copy of these files. They only exist as distributed *MIST* spread across *Custodian* devices.

### 5.2.2 Secure Relay Network

Due to battery power conservation, mobile devices are *asleep* most of the time and can only occasionally check for incoming traffic. To ensure that the *owner's* crypto assets can be accessed quickly and efficiently, we need a reliable, always-on, secure, asynchronous messaging network of communication relays.

For that purpose, we specifically designed and built strong, cryptographic, open source "*Zax*" relays[28]. All *shards* are moved between the *owner's master device* and *Custodians'* using these relays. This asynchronous messaging network provides all the communications necessary to connect *owner* and *Guardian* devices around the world. This allows each device in the relay network to just send a short burst of encrypted communications to a few nearby relays and go back into *sleep* mode to conserve battery power.

The relays offer total anonymity to all parties. Not only it is theoretically impossible for relays to recover the content of communications between two devices, but relays also do not know the identities of devices participating in communication sessions. This is due to the special hardened design of the cryptographic protocol powering "*Zax*"[29] relays. Relays offer *perfect forward secrecy*[30] to all communicating parties. All PKI keys, used

[28] M. Skibinsky, Y. Dodis, (2015). "Asynchronous Mobile Peer-to-Peer Relay," https://s3-us-west-1.amazonaws.com/vault12/crypto_relay.pdf

[29] "Zax" relay source code. https://github.com/vault12/zax#zax-

[30] https://en.wikipedia.org/wiki/Forward_secrecy

in communication sessions, are created only for a given session and are permanently destroyed a few minutes after each session.

If *owners* desire additional security, the open source relay enables them to set up their own private, secure relay network. This ensures that *owner's* assets and *shard* traffic are always transferred in a controlled environment. It also enables a new class of service providers to deliver best-in-class relay networks with the highest degrees of availability and security.

### 5.2.3 Secret Management Module

*Secret Management Module* is responsible for converting any assets or documents, given to the platform applications, into H3S *shards* according to the policies setup by the *owner*. It is also responsible for reconstructing original documents from given *shards*. *Secret Management Module* also manages all of the private and public keys required for the low-level cryptographic operations necessary to convert documents that are submitted for processing. Other vendors can build different *cryptostorage platform* applications by providing different UX or functionality based on information queries sent to the *Secret Management Module* of given agent (*Guardian*, *Custodian*, etc.).

### 5.2.4 Authentication Module

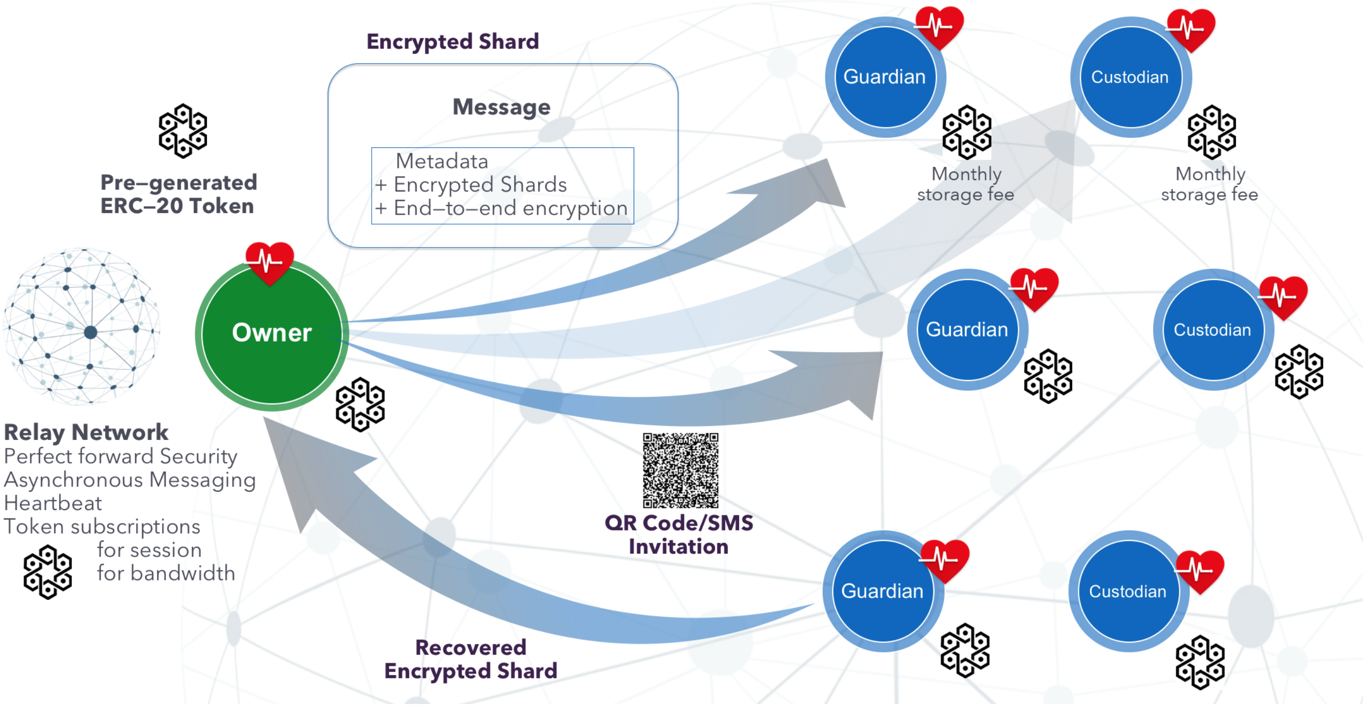
The *Authentication Module* is responsible for verifying the *owner's* identity to the trust circle he established during the *Vault* setup process. The default option is social verification via phone call, video chat or another direct communication channel. The *owner* can select a specific social verification policy – for example, requiring only FaceTime video call verification when accessing high-security containers. The *Authentication Module* will enforce given policy requirements for each *Custodian* device when *shards* are requested by the *owner*.

*Owners* can opt-in to use 2FA/U2F/HSM devices for faster authentication. In such cases, during *vault* setup process, the *Authentication Module* queries 2FA/U2F/HSM devices to create a shared authentication credential/keypair for each *Custodian* in the trust circle. Later, when the *owner* requests a shard back, his 2FA/U2F/HSM devices adds a time-dependent verification block to the request. If that block is successfully verified against authentication credentials by *Custodian* device, it can automatically release the *shard* back to the *owner* without the *Custodian's* direct involvement[31].

[31] We not using the *owner's* phone itself as 2FA device, because, if that phone is lost, it defeats whole purpose of distributed security.

# 6. Vault Guardian Token: VGT

So far, we simply assumed that both *Custodians* and *relay network operators* are always available to facilitate storage and transmission of an *owner's* encrypted *shards*. In the real-world, we cannot expect for these services to be free or to have a high degree of reliability. Although some people might be willing to participate for the altruistic reasons (such as helping a friend safeguard their *Vault* or doing a public service by running a free node), in general, the platform will be far more reliable and have wider avaiablity if we align incentives of *Custodians* and *relay* operators with incentives of the *owners*. To provide the settlement mechanism for all parties to reward activities that benefit *owners* and the platform economy as whole, we have created *Vault Guardian Tokens (VGT)*.



*Vault cryptostorage platform token economy.*

## 6.1 Custodian Incentives

*Custodians* are giving *owner's* free storage space on their phones, which will grow as *owners* add more files to their *Vaults*. To motivate *Custodians* to safeguard the *owner's shards*, respond to *owner's* requests and to demotivate *Custodians* from deleting platform storage application, *owners* can opt-in to pay a mothly storage fee. This will be proportionate to the amount of assets stored and level of service expected from a given *Custodian.*

### 6.1.1 Professional Custodian Services

Any *owner* can easily setup his *Vault* using friends and family. *Custodians* can always be recruited from the *owner's* personal network. However, while casual *Custodians* are perfectly suitable to secure consumer wallets, they might not be the best option to secure assets of high-net-worth individuals or extremely sensitive financial documents. To address this market, high-net-worth *owners* will require a *professional custodian service* (PCS). A PCS is organized by individual attorneys or law firms to offer premium services for additional monthly fees. These premium services include:

- Advanced security for safeguarding *shards* at rest, such as storage of *Custodian* devices only in access-controlled facilities in a security safe.
- Additional safe-guards for verifying the *owner's* identity, such as providing *owner* with hardware 2FA device, verifying *owner's* government-issued ID in person and other methods of secure authentication.
- Full insurance for *owner's* funds under protection.

Because all of these services have significant additional costs, *Custodians* will be able to advertise for and charge additional mothly fees for these services. As an extreme case, high-valueadd *PCS* providers may even require payments as large as few basis points of total amount of assets under custodianship[32].

### 6.1.2 Custodian Storage Payments

When creating a new *Vault,* the *owner* will see the prices, duration (usually 6 to 12 months) and services in the *Custodian Smart Contract* advertised by each invited *Custodian.* Some *Custodians* can remain free (such as *owner's* personal devices), while others can be individuals or *PCS.* Once *owner* activates his *Vault*, he accepts all smart contracts advertised by his *Custodians*, which transfers the tokens required to satisfy contract's duration from *owner's* app to the escrow account. Every month *Custodians* are paid from the escrow account according to smart contract terms. *Custodians* who do not pass a daily shard health check are proportionally compensated less and reminder of escrow account reverts back to the *owner* at the end of contract period. *An owner* and

[32] Given the perfect secrecy of H3S cryptography, *Custodians* never know the exact amount owner's assets. However, if *owner* decides to opt-in into a high-value service (for example, an insurance coverage of stored funds), they will have to voluntarily share this information with the PCS.

*Custodians* can opt-in into automatic roll-over of the *Vaults* subject to the same conditons.

## 6.2 Relay Operator Incentives

The service profile of a *relay operator* is simular to profile of commercial website. *Relay operators* incure hosting, bandwidth and maintance costs for keeping relays running. To offset these expenses, *relay* operators should be able to charge fees to offset these costs. Additionally, it provides *relay* operators with a convineint tool to defend against spam and DOS attacks.

Different *relays* might require different payment structures. On the platform level, we will provide a comprehensive list of token payments that *relay operator* can configure out of the box.

- Fixed token fee for each request
- Fixed token fee for each session
- Proprotional token fee per unit of bandwidth
- Proprotional token fee per unit of storage
- Unlimited and limited subscription fee per billing period

*Relay operators* will have full freedom to mix and match any of these *relay* options to create a billing policy best suited to their needs.

## 6.3 Platform Application Vendors

Distributed cryptostorage offers broad range of use cases, ranging from casual users securing small crypto investments to crypto funds that have to secure hundreds of millions of dollars' worth of assets. It is unlikely a single application with perfectly serve all use cases for all customers. Therefore, different *application vendors* can explore different solutions and tradeoffs by offering various applications built on the *Vault Cryptostorage Platform.*

*Application vendors* can opt to accept payments for installation, monthly use and unlocking advanced features by accepting token payments from *owners.*

## 6.4 Vault Guardian Tokens

Vault12, Inc. created the *Vault Guardian Token* (VGT) to power all services and applications deployed on *Vault Cryptostorage Platform.* The *VGT* is intended to be used for the following purposes.

- O*wners* can incentivize their *Custodians* by either offering direct, periodic payments or by accepting a *smart contract* from a professional custodian service. For example, let's say Jon has five friends and family, casual *Guardians.* Jon will transfer 600 *Guardian tokens* to his platform storage app and assign them to be paid as *10* tokens per month to each *Custodian,* during the upcoming year. Advanced payment options for *professional custodian service* will require Jon to accept *smart contracts* offered by these services.
- *Relay operators* can establish a specific token charge policy for *relay* usage. For example, Jon connects to *relays* that offer higher speed and guarantee 24/7 reliability in major countries. He transfers an additional 100 Guardian tokens to his platform storage app and sets this amount as bandwidth budget for premium relays as 10 VGTs per session.
- *Owners can p*ay for installation, monthly use and unlocking of additional functionality in *Vault Platform Applications.* The reference *Vault12* mobile application will follow the same policy and will be fully accessible for *VGT* payments.

## 6.5 Token Issue and Management Policy

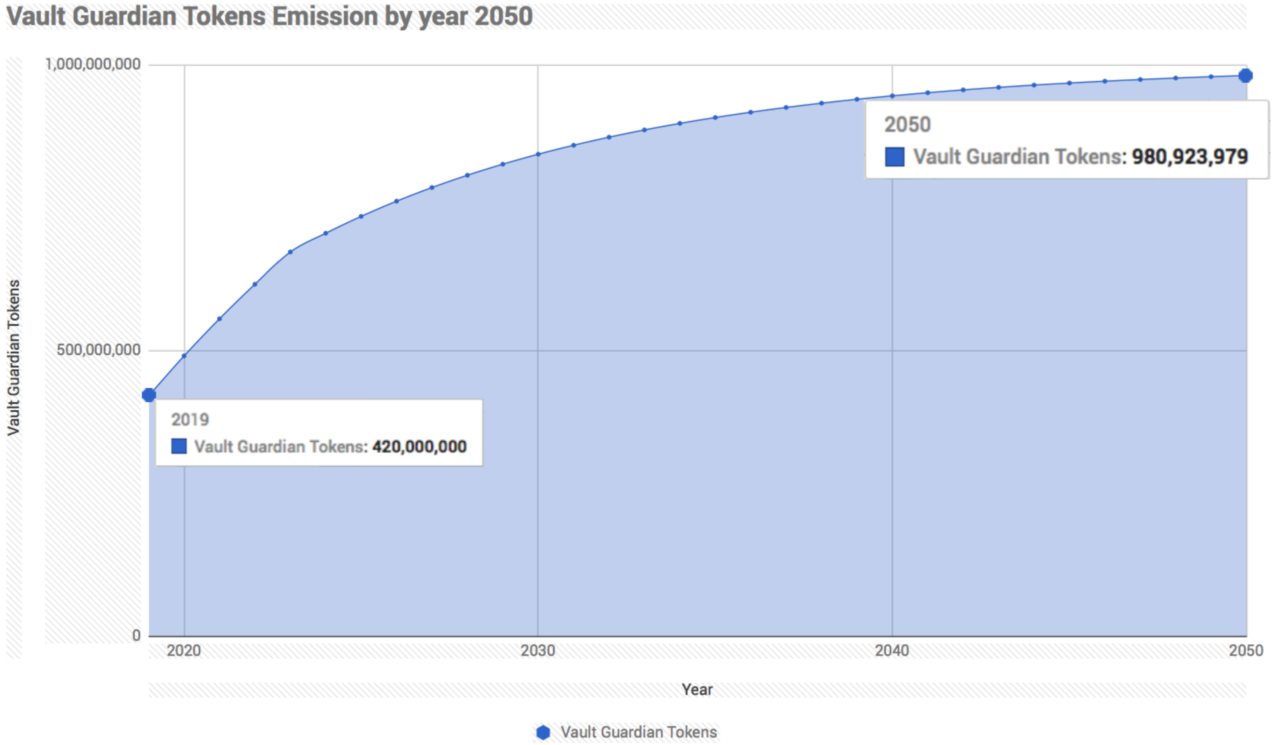
To bootstrap operations of the *Vault Cryptostorage Platform,* Vault12, Inc. will create a fixed supply of 1 billion ($1x10^9$) *Guardian* tokens following *ERC20* standard and will maintain the ledger. There will be no mechanism for supply to be increased.

In the initial token sale, Vault12, Inc. will sell 40% of all issued tokens. 10% will be allocated for a community reward pool – to be awarded, at Vault12, Inc.'s discretion, to individuals and organizations for outstanding, high-quality open source contributions to the *Vault Cryptostorage Platform* and codebase. 50% will remain locked in company inventory for one year after the end date of the initial sale. After the initial one-year lock-in period, Vault12, Inc. will never sell more than 10% of remaining locked token supply within calendar year[33]. Every proprietary application published by Vault12, Inc. will offer all options for payments and subscriptions via *VGT*.

# 7. Token Economy Model

To estimate yearly economic activity in the *Vault Cryptostorage Platform*, we need to estimate two key factors: total number of *Vault Guardian Tokens* in circulation, and the economic value transferred due to platform activities.

## 7.1 Token Emission



Vault Guardian token supply will start with an initial sale of $4x10^8$ (400 million) tokens in 2018. Additionally, $1x10^8$ (100 million) tokens are reserved as community rewards. Let's assume about 20 million tokens of community rewards are awarded each year, and after five years all community rewards are distributed since the platform software development will be largely complete.

[33] In other words, no more than 5% in 2nd year, 4.5% in 3rd year, etc.

In such case, token supply will start with $4.2 \times 10^8$ (420 million) tokens in 2019. Every year token issue will be the total of *negative* 10% compound interest on remaining balance of locked tokens, starting from $5 \times 10^8$ (500 million). Therefore, next year there will be $2 \times 10^7$ community rewards and $5 \times 10^7$ newly unlocked tokens added to the circulation, resulting in total $4.2 \times 10^8 + 0.7 \times 10^8$ (new supply) $= 4.9 \times 10^8$ (490 millions) of released tokens by the year 2020, and so forth. The graph above models token emission until the year 2050 when there will be slightly less than 981 million tokens in circulation.

## 7.2 Market Size

Initial customers for the *Vault Cryptostorage Platform* will be various segments of crypto owners, which we rank by assets value – from top tier crypto funds all the way down to casual crypto owners. We assume professional funds will require most stringent form of protection with the most distributed *Guardians* (for geographical distribution and backup robustness) and significant number of *Professional Custodians* (for increased security and authentication). In contrast, casual crypto owners will be satisfied with as little as five *Guardians* supplemented on a few personal mobile devices.

After few years of growing the platform and application solely for cryptocurrencies owners, the applications will develop to serve broader demographics. At that moment, many other organizations (law firms, real estate, family funds, intellectual property corporations) that need to protect critical documents or trade secrets can start leveraging the *Vault Cryptostorage Platform.*

## 7.3 Methodology

To project market size for the **Vault Guardian** token (VGT), we first segment holders of cryptocurrency wallets and based on wallet size, make assumptions on numbers of Guardians, security preferences and penetration. Secondly, we look at the growing category of digital assets found in real estate, family fund management, law offices and the protection of trade secrets. The projection also takes into account the viral factor of Guardians deciding to use the platform to protect their assets.

Here are all the assumptions we used in our model:

- Casual investors use *5 Guardians* and reward each with *$10* payments per month. They also pay *$100* per month as application subscription. Casual investors do not setup private networks, nor do they use premium relays. Within 5 years *Vault Cryptostorage platform* is used by 2.5% of casual crypto investors.

- More established crypto investors use *7 Guardians* and reward each with *$10* payments per month. They also pay *$100* per month as application subscription. Crypto investors do not setup private networks, but they sign up for a few premium relays that cost them *$1500* per month. Within 5 years *Vault Cryptostorage Platform* is used by 5% of crypto investors.

- Professional investors use *10 Guardians*. One third of these Guardians are professional service firms that charge *$3,000* per month for advanced authentication services and shard protection. The rest are rewarded with the usual fee of *$10* per month. They also pay *$100* per month as application subscription. Professional investors do not setup private networks, but they sign up for a few premium relays that cost them *$3000* per month. Within 5 years *Vault Cryptostorage Platform* is used by *5%* of professional crypto investors.

- Top-tier investors use *25-50 Guardians*. One third of these Guardians are professional service firms that charge *$3,000* per month for advanced authentication services and shard protection. The rest are rewarded with the usual *$10* per month. They also pay *$100* per month as application subscription. Top tier investors either setup their own private networks, or alternatively they sign up for a few premium relays that cost them *$3000* per month. Within 5 years *Vault Cryptostorage Platform* is used by *5%* of top-tier crypto investors.

- Starting in 2020-2021 other organizations such as law firms will start leveraging Vault Cryptostorage Platform.  They will use 5-10 Guardians for the usual $10 per month and may use premium relays. Their setup is similar to that of mid-tier crypto investors. Within 5 years Vault Cryptostorage Platform is used by 1%-2.5% of non-crypto organizations.

- Vault Cryptostorage Platform operations require a big number of guardians to install the storage application every year. This represents strong viral channel for the platform distribution. We estimate *20% of installed Guardians* will convert into «*casual investors*» in a given calendar year.

# 8. Summary

The *Vault Cryptostorage Platform* is a new, open-source initiative to leverage decentralized cryptography and storage to protect our most precious digital assets. Today, this means cryptocurrencies. In the future, this could include titles, passports, house keys, car keys and all kinds of other digital credentials.

Powered by *Hierarchical Shamir's Secret Sharing* (H3S) cryptographic algorithm with perfect theoretical security, the *Vault Cryptostorage Platform* distributes secrets across a trusted mesh network of personal mobile devices designated by cryptocurrency *owners*.

Designed to be used alongside traditional hardware, software and online wallets, the *Vault Cryptostorage Platform* gives cryptocurrency *owners* the peace of mind that their crypto assets not only remain backed up and cryptographically secure, but also accessible regardless of well-known and emerging threat vectors. The *Vault Cryptostorage Platform* takes an entirely new, decentralized cryptographic approach that can be easily deployed on existing mobile devices and combines it with a distributed storage network, which is made possible with the advent of decentralized cryptography.

For more information and to view our live product demo, visit https://vault12.com.

# 9. Reference Links

1. Shamir, A. (1979). *"How to Share a Secret,"* http://cs.jhu.edu/~sdoshi/crypto/papers/shamirturing.pdf
2. Tassa, Tamir. *"Hierarchical Threshold Secret Sharing,"* https://www.openu.ac.il/lists/mediaserver_documents/personalsites/tamirtassa/hss_conf.pdf
3. Szabo, Nick (2001). *"Trusted Third Parties Are Security Holes,"* http://nakamotoinstitute.org/trusted-third-parties/
4. M. Skibinsky, Y. Dodis (2015). "*Asynchronous Mobile Peer-to-Peer Relay,"* https://s3-us-west-1.amazonaws.com/vault12/crypto_relay.pdf
5. Vault12, Inc. (2015). "*Zax*" relay source code, https://github.com/vault12/zax#zax-
6. Vault12, Inc. (2017). *"How to get true randomness from your Apple device with particle physics and thermal entropy,"* https://medium.com/vault12/how-to-get-true-randomness-,from-your-apple-device-with-particle-physics-and-thermal-entropy-a9d47ca80c9b
7. Vault12, Inc. (2018). Cryptostorage mobile app demo, https://vimeo.com/258424334
8. Vault12, Inc. open source projects, https://github.com/vault12

# 10. Table of Contents