Bitkey Recovery Features

Bitkey Overview

Bitkey is a self-custody bitcoin wallet that includes a mobile application, hardware device, and recovery tools. Bitkey is a 2-of-3 “multisignature” wallet — two keys are required to move money. Customers control two of the three keys using the mobile app and the hardware device. Block, the company that builds Bitkey, manages the third key in order to help customers regain access to their funds when they lose one — or even both — of their keys, and to cosign transactions up to a limit that a customer has pre-authorized, so that customers can keep their hardware and phone separate for small transactions.

This multisignature design aims to:

- Ensure that customers can move their money without Block
- Ensure that Block cannot move money on its own
- Insulate customers from losing one (and in some cases, two) components of their wallet
- Provide and prioritize resilience to accidental loss while maintaining a high degree of security against attackers
- Provide customers the convenience and flexibility of transacting with only a phone, without the friction and fees that come with managing a separate hot wallet

Bitkey’s approach to recovery includes 4 primary capabilities:

- **Cloud Backup**, for when you lose access to the key stored on your phone, for example if your phone is lost or stolen - or you just get an upgrade
- **Delay and Notify**, for when you lose your hardware, or when you lose both the key stored on your phone and your cloud backup
- **Recovery Contacts**, an optional feature for when you lose both your hardware and the key stored locally on your phone
- **Emergency Exit Kit**, for if you no longer can or no longer want to use the current version of the Bitkey app to access your funds

The remainder of this document describes the intent and design of these features in detail.

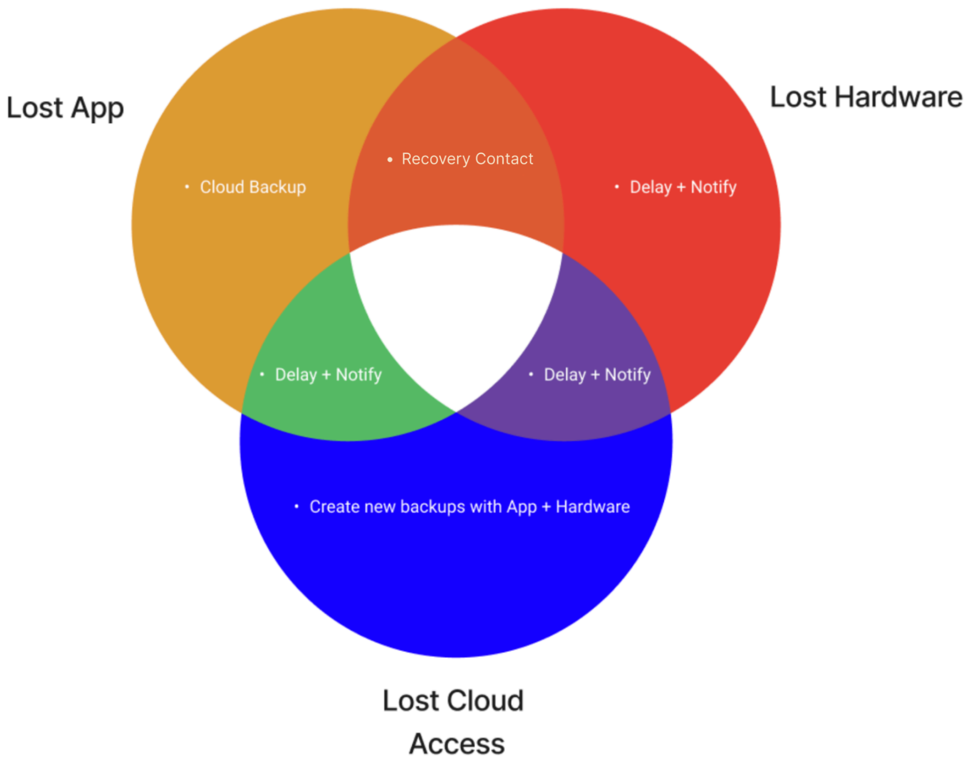

Figure 1: Recovery tools used for different loss scenarios.

Cloud Backup

Bitkey is focused on ease of use for a wide audience, so we want to make the most common cases of accidental loss incredibly easy to recover from. One of the most common situations people encounter in 2023 is losing or replacing their phone.

To allow people to quickly and seamlessly regain access to their funds on a new phone, we've implemented a cloud backup feature for the App Key stored on each customer's phone. When a customer sets up their wallet, the Bitkey mobile app stores an encrypted copy of the App Key in the customer's cloud storage associated with the Apple or Google account they used during setup of the Bitkey app. The backup can only be decrypted using a key that is stored in — and never leaves — the customer's Bitkey hardware device. This process requires customers to grant permission to the app to store data in their connected cloud account — and it is a required step during Bitkey setup. If creation of the cloud backup fails at any point during onboarding, the Bitkey mobile application will not allow a customer to use the wallet — this is an important safety feature that helps ensure customers maintain control over their bitcoin.

When a customer later wants to regain access to their funds on a new phone — say, after losing a phone or upgrading to a new device — they simply need to log into their cloud account they used when initially setting up their Bitkey wallet and re-install the Bitkey mobile app. The Bitkey mobile app will automatically download the cloud backup and prompt the customer to unlock their hardware device and tap it to their phone in order to decrypt the cloud backup. At this point, the customer has regained access to their App Key, and together this App Key alongside their hardware key means the customer once again holds full control over their funds.

Cloud Backup: Platform Storage Mechanisms

Bitkey's cloud backup feature of the App Key relies on Apple and Google-provided storage mechanisms. On iOS, our current implementation uses iCloud key-value store. On Android, we use app-specific storage backed by Google Drive. These storage mechanisms are used for the encrypted cloud backup, which contains an encrypted copy of the App Key. The App Key is also stored locally on the customer's phone, using the local Keychain on iOS and EncryptedSharedPreferences (which relies on the local Keystore) on Android.

In our implementation of cloud backup on both platforms:

- **Bitkey cloud backups are hardware-encrypted**
  - Before storing backups in the customer's cloud storage, the Bitkey mobile application encrypts the backup with a process that requires the customer's Bitkey hardware device (which holds one of the other keys involved in the wallet). This means that if an attacker gains control of a customer's cloud storage account, they cannot decrypt the customer's backup without the customer's unlocked hardware device - and thus do not gain access to any of the three keys that comprise the customer's wallet.
- **Even if platform vendors could access backups, they're encrypted**

- Cloud platform vendors may be able to access the customer's encrypted cloud backup, either in the course of their normal business operations as defined through their terms of service, or via a compromise by a rogue employee or attacker who gains access to their systems. Because Bitkey cloud backups are encrypted, and because even the decrypted version is only 1 of 2 keys required to move funds, such access does not provide an attacker with access to funds stored in Bitkey.

Cloud Backup: Inner Workings

Cloud backups contain "keysets" — groupings of BIP-32 extended keys for 3 public keys corresponding to the phone, hardware, and server — as well as the seed for the phone's key. The backups may contain both "active" and "inactive" keysets. Keysets become "inactive" when a customer loses a key and conducts a recovery operation that results in moving funds into a new keyset. Bitkey stores these inactive keysets in order to recover funds that may be sent at a later time to an address a customer gave to a sending party prior to conducting a recovery.

Prior to storage in the customer's iCloud or Google Drive, backups are serialized into json and encrypted. The backup is first encrypted with a 128-bit cloud storage encryption key ("CSEK") generated by the customer's Bitkey mobile application. Then, the CSEK used to encrypt the backup is transferred via NFC to the customer's Bitkey hardware, which encrypts it using a unique key and transfers the encrypted CSEK back to the Bitkey app. The Bitkey app then stores the encrypted CSEK alongside the encrypted backup in each customer's cloud storage.

The keys involved in the cloud backup process are detailed in the table below, and the setup process is depicted below in Figure 2.

Name	Description	Storage
PKMat	Private key material: App Key to be encrypted and backed up using Cloud Backup	Mobile Device
CSEK	Cloud Storage Encryption Key, used to encrypt the private key material, PKMat. Generated by the Bitkey app.	Ephemeral
HWDEK	Hardware Data Encryption Key, used to encrypt the CSEK so that only the customer's Bitkey hardware can be used to decrypt the cloud backup material	Bitkey Hardware

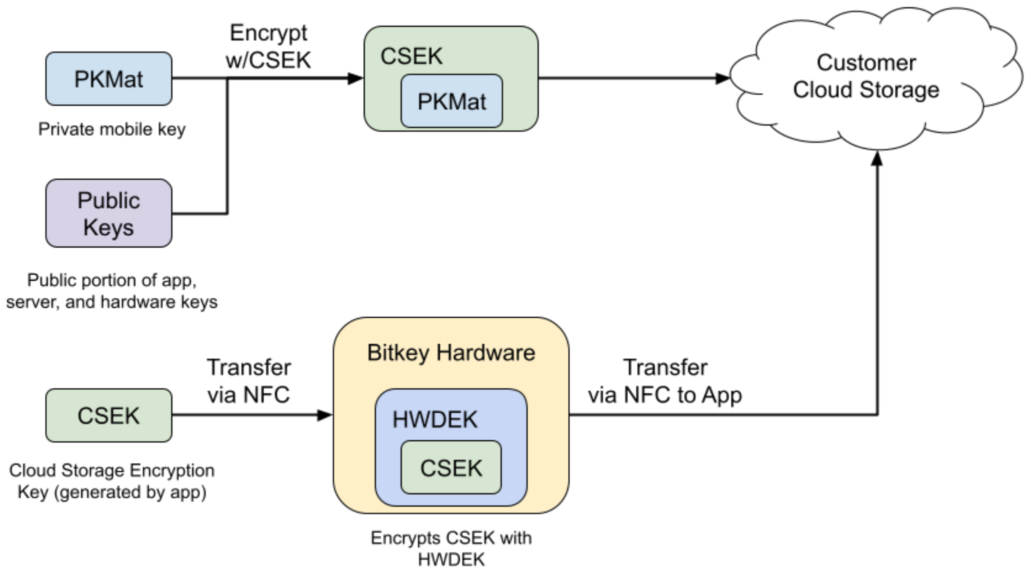

Figure 2: Cloud backups are encrypted using keys known only to the customer's phone and hardware device.

Cloud Backup: Limitations

The cloud backup feature provides one-tap setup and recovery for most customers in a number of common situations — without adding a new account or password for customers to manage and without exposing customers to loss of funds in the event their cloud account is compromised. However, as with every recovery-related feature, cloud backups cannot address all situations. The most notable limitations of this feature include:

- Cannot handle platform switches
  - A customer who switches from Android to iOS or vice versa will not be able access their cloud backup on their new device, because these operating systems do not provide a mechanism to share private App Key backups across platforms.
- Must be used with a cloud account that actually contains the backup
  - A customer who is unable to sign in to the Apple or Google account that they used to set up the Bitkey app will not be able to access the cloud backup. Customers who've forgotten their password or have lost all of their two-factor authentication devices and backup codes must use another recovery method to regain access to their funds (we cover these other alternatives below).
- System settings can delete backups or interfere with the app's ability to access or update backups
  - After an encrypted cloud backup is created by the Bitkey mobile application and stored in the customer's cloud storage, it is possible for customers to revoke the Bitkey app's access to cloud storage, which would interfere with the ability of the app to restore from backup and to update the backup during subsequent recovery events, or even result in deleting the backup. For example, on iOS,

customers may disable iCloud Drive after having set up their Bitkey wallet and stored a cloud backup there, resulting in wiping the cloud backup from the customer’s iCloud storage. On Android, customers can revoke permission for the Bitkey application to store data there. The Bitkey app will periodically check for presence of the cloud backup (and ability to update it), alert customers if it cannot detect the backup, and recommend remediation steps — but in the rare cases in which a backup cannot be detected, customers will have to take specific actions to restore it, including fixing a mobile OS system setting and tapping their hardware to generate a new backup.

Cloud Backup: Cheat Sheet

	When it helps you	**How you use it**	**How it works**
Cloud Backup	When you get a new phone or delete the app and need to reinstall it.	Install the Bitkey mobile app on your new phone and log into your cloud account used to initially create your wallet. The Bitkey app will direct you to unlock and tap your hardware to regain access to your wallet.	The Bitkey mobile app stores an encrypted copy of the App Key in the customer’s cloud storage account. The copy is encrypted by the customer’s Bitkey hardware prior to storage, and must be decrypted by the customer’s Bitkey hardware during recovery.

Delay and Notify

Features like Cloud Backup help customers recover easily from common situations like losing their phone or getting a new one. But what happens when a customer has lost their hardware device, or lost access to both their phone *and* their Cloud Backup? While these scenarios may be less likely to occur than simply getting a new phone, we nonetheless want to help customers protect themselves from these other loss scenarios — using a feature we're currently calling "Delay and Notify."

Delay and Notify allows a customer who has access to only one of their two keys to begin a recovery process that will ultimately let them move funds to a new set of Bitkey keys protecting their wallet balance — after a predefined period of time referred to in this document as the "security waiting period."

During the security waiting period, Bitkey servers will notify the customer that the recovery is in progress via notification methods that the customer configured during wallet setup (push notification, SMS, and/or email), and will provide a link that can be used to cancel the recovery process in case they did not request the recovery. If they indeed were the person who generated this recovery, then they can take no action and simply wait for this security period to expire. In the case where the customer indeed needed and requested this recovery, they might be a little frustrated to have to wait for this period to pass but that delay is there for their protection. This approach protects customers from potential security issues discussed below in the Contested Recovery section.

When the security waiting period elapses, essentially signaling to Bitkey servers that the customer indeed wanted to complete this recovery, Bitkey servers will co-sign a transaction to move the wallet's funds to a new set of keys generated by the Bitkey app and hardware device — a process that requires a signature from the customer's remaining key in order to meet the requirement that 2-of-3 keys have signed a transaction for acceptance by the bitcoin network. Once this recovery transaction is confirmed on the blockchain, the customer has control of 2 new customer keys that now control their funds.

The Delay and Notify feature is enabled for all Bitkey wallets by default, and cannot be disabled. The security waiting period is not configurable by customers — in our external beta, it will be set to 7 days, and this number may change before the product is available to the public. During the security waiting period, Bitkey servers will send notifications about the ongoing recovery to all of a customer's communications channels (e.g. push, email, SMS) every 2 days, including after the security waiting period has expired, until the recovery is resolved. For a security waiting period of 7 days, inclusive of the notifications at the start and end of the security waiting period, this results in at least 5 notifications.

Delay and Notify represents a tradeoff: rather than requiring customers to always keep control of 2 keys, we chose to allow customers a path to recovery when they're down to 1 key – while this may introduce more attack surface, we believe that providing resilience to accidental loss leads to more safety in practice. The remainder of this section covers how the Bitkey system strikes this tradeoff.

Delay and Notify: It's a Co-Signing Policy

The Delay and Notify feature is a mechanism implemented by Block that defines, in code, the conditions under which Bitkey servers will co-sign a Recovery Transaction — this is effectively a "co-signing policy."

A fundamental tenet behind Bitkey and self-custody is that **keys represent control and ownership** — not personal information, government IDs, selfies, biometric databases, or any other hallmarks of invasive bureaucracy that put companies, and not customers, in control. As a result, in order to co-sign a recovery transaction, Bitkey servers require:

- a security waiting period to elapse, and
- the person requesting recovery to prove control of one of the two customer keys, and
- in the event that both customer keys are being used to request recovery (e.g. if an attacker has successfully stolen one key), proof of control of the customer communications channel (email or SMS) can be used to break the tie and proceed, as described below in the Contested Recovery section.

Bitkey's co-signing policy is expressed in code that runs on Bitkey servers, and is written and deployed by Block engineers. This code will be changed over time by Block engineers to improve Bitkey's recovery services. While it could be changed by a malicious actor if they were able to sufficiently compromise Block's infrastructure, such modifications do not immediately lead to funds loss because this code and environment manages 1 key for customers, with the other 2 keys held in customer hands.[1] Prior to general availability of Bitkey, this code will be published to GitHub as part of our open development approach. Although it will not yet be possible to verify that this code matches exactly what is running on Bitkey servers, publishing this code will help invite scrutiny on any changes the Bitkey team makes to the properties of features like Delay and Notify.

Delay and Notify: The Security Waiting Period and Contested Recovery

In many scenarios, using the Delay and Notify feature - and the security waiting period in particular - will be uneventful for customers. Consider Alice, who dropped her phone overboard while boating, and when buying a new phone, realized that she doesn't remember the password to her Google account - and has no other way to regain access to her Google account. This means she can't get to her Cloud Backup - so she uses her new phone and her Bitkey hardware device to begin the Delay and Notify process. With her phone at the bottom of the lake, she's the only one in control of either customer key. Throughout the security waiting period, Bitkey

[1] Security guarantees for the backend "Wallet Security Module" that manages the third key for customers will be covered in future published security materials.

systems will send alerts to the email or SMS she provided during wallet setup, and nobody will cancel the recovery using the links in those messages (because only Alice controls her communications channel) nor attempt to start another recovery with the other customer key (because only Alice controls a key). The security waiting period will expire and Alice will safely recover to a new wallet, which she has associated with a new Google account she controls. In the scenario covered above, the security waiting period represents friction for Alice -- for security purposes, she must wait until its expiry in order to regain access to her funds. Why exactly did Bitkey make her wait? Well, in case it hadn't really been her who requested that recovery - we'll illustrate this with our next example.

Consider Bob, whose hardware device was stolen by an attacker who was also somehow able to hack past the fingerprint lock. One day, Bob receives an email and push notification that someone has started the Delay and Notify recovery process for his wallet - because the attacker used Bob's unlocked hardware to do so. Bob taps the push notification and uses the Bitkey mobile application to tell Bitkey servers that he didn't start the recovery and doesn't want it to proceed.

At this point in Bob's story, we call the situation a Contested Recovery – Bitkey servers are no longer sure which of the two customer keys (the attacker's access to Bob's hardware key, or Bob's control of his own App Key) to co-sign a recovery transaction with. To resolve this contest, we defer to the communications channel provided for recovery purposes during onboarding. In practice, when this type of contest arises:

- Any pending Delay and Notify recovery process is canceled
- Restarting any Delay and Notify recovery process will require confirming a verification code sent to the email or SMS associated with the wallet (to use Bitkey, customers must configure at least one of these communications channels)
- Any previously enabled 'mobile pay' spending limit will be disabled, so that no spends are possible using only the App Key

Delay and Notify: Inner Workings

Authentication Keys

The Bitkey mobile application and hardware device manage keys used for spending bitcoin, as well as the seeds they are derived from. The app and hardware also manage 'authentication keys' – keys derived from the same seed used to derive spending keys – but which are used only to interact with Bitkey servers and never to directly secure or move bitcoin on the blockchain. Authentication keys are used to prove - to Bitkey servers - possession of a key derived from the mobile or hardware seed in order to conduct security-sensitive actions like beginning the Delay and Notify recovery process. When a customer completes a Delay and Notify recovery, their authentication keys are replaced by the new authentication keys their app and hardware provided during initiation of the recovery process.

Signing Policy Logic

Bitkey servers sign transactions for customers in two broad settings: recovery, and spends below the customer-configured daily limit that enables a customer to use just their App Key to spend money. Prior to signing, Bitkey servers verify a range of properties prior to signing transactions, which follow the approximate logic outlined below:

1. Check that the request is authenticated by an authentication key that:
    - Shows possession of phone or hardware for a wallet Bitkey servers hold the 3rd key for
    - Is the 'correct' factor for the type of request (e.g. phone for mobile pay)
    - If the request was not authenticated with the correct authentication key, don't co-sign

2. Check if there is a Delay and Notify recovery pending for this wallet
    - If there is, and this request was made using the other factor (e.g. hardware when someone already started Delay and Notify using the phone), suspend the existing recovery, don't co-sign, and require verification using email or SMS to begin recovery again

3. If there is a pending Delay and Notify recovery, check if the security waiting period has elapsed
    - If it has not elapsed, don't co-sign

4. Check if the PSBT outputs are entirely owned by the wallet in question
    - If they are NOT, then verify that the cumulative outflows today are less than the customer-configured spending limit
    - If the limit would be exceeded, don't co-sign

5. If we got to this point in the logic, co-sign

In practice, the logic above is implemented across multiple APIs. For implementation details at this level of granularity, we refer readers to Bitkey's forthcoming Github repository.

Delay and Notify: Limitations

- Implementing this approach involves complexity
    - Managing the 3rd key in Bitkey's 2-of-3 multisignature wallet architecture - and in particular a feature like Delay and Notify - means managing state across a hardware device, mobile app, and backend servers in order to provide a safe and smooth experience that protects customers from accidental loss. This means more code that needs to be vetted for correctness, analyzed for security vulnerabilities, and maintained as Bitkey's product and feature set grows. We take this on in service of our customers – we're prioritizing their needs and making self-custody a realistic option for more people, even if it means making our lives more challenging
- Recoveries require network fees

    - When a Delay and Notify recovery is completed, a Recovery Transaction is broadcast to the bitcoin network in order to sweep funds to a new wallet. A customer must pay bitcoin network fees in order for this transaction to be accepted, as well as wait for confirmation of the transaction to the blockchain
- Customer must buy a new Bitkey device when they've lost their hardware
    - In our current implementation, we don't support recovering directly to an arbitrary address
- Security waiting period is not currently customer-configurable
    - In our beta implementation, this delay is set to 7 days, but may change before general availability; and may change over time as well as we learn about customers' needs and preferences
- SMS delivery isn't guaranteed - which is why we also encourage a secondary comms channel like push notifications and email for customers who choose SMS
    - This means that customers may not receive every text message sent to them by Bitkey servers to notify a customer that there is a recovery in process for their wallet. Although Bitkey servers can and do send multiple messages, SMS does not provide a mechanism to confirm delivery and is not failsafe. We also find in our early research that many customers may prefer to select multiple communications channels - for example, both email and SMS - to help ensure they react quickly to important messages
- Recoveries involve a sweep transaction that includes all UTXOs owned by the wallet
    - This can detract from wallet privacy by associating UTXOs with each other via the common-input assumption
- Customers can lose access to their communications channel(s)
    - If a customer has lost their hardware *and* the communications channel(s) they configured for recovery, or has lost their app and cloud backup *and* communications channel(s), *and* an attacker actively controls the lost key and attempts recovery with it, then funds can be permanently deadlocked (and, from a customer point of view, this is equivalent to lost)
- An attacker with control of a communications channel (e.g. email or SMS) can block a customers' recovery attempts in the case they independently lose another part of their wallet
    - This is because Bitkey servers send notifications about - and an opportunity to cancel - recovery actions. An attacker who controls a customer's email or SMS cannot steal funds in this case, but they can cancel recoveries initiated by, for example, a customer who lost their phone or hardware device and is attempting to replace it.
- Bitkey server code that implements the co-signing policy is not yet verifiable
    - We will open-source the code prior to general availability - but this code runs in an Amazon Web Services environment managed by Block whose properties customers cannot directly verify. While we may be able to provide verifiability through AWS' Nitro Enclave Attestation, we have not yet evaluated this approach nor implemented it

Delay and Notify: Cheat Sheet

	When it helps you	How you use it	How it works
Delay and Notify	When you lose access to either: • your hardware device, or • both the cloud account you used to set up Bitkey and the App Key stored on your phone	Begin the recovery process by reporting a lost hardware device from your existing Bitkey app, or if you lost your phone and cloud backup, by using your hardware device with the Bitkey app on your new phone. If you've lost your hardware device, you'll need to order a new one. In both cases, after the security waiting period elapses, assuming you don't cancel the request for any reason (like you found it again), you'll have access to your funds in a new Bitkey wallet.	After a delay, Bitkey servers co-sign a transaction along with a customer-held key to move funds to a new wallet, alongside whichever key you still have control of. Bitkey servers may require you to prove control of the customer communications channel you used during signup (email or SMS), in the rare case that there are two separate customer keys attempting recovery at the same time.

Recovery Contacts

Bitkey’s Cloud Backup and Delay and Notify features work to keep customers in control of their mobile and hardware keys, and to help them regain access to their funds when they’ve lost one of their keys. Bitkey’s optional Recovery Contacts feature protects customers when they’ve lost *both* their phone and their Bitkey hardware device during the same time period, but are still able to access their cloud account.

To protect themselves using Recovery Contacts, customers enroll one or more friends, family members, colleagues, or others as their Recovery Contacts (RCs). The role of these RCs is to verify during a recovery that it is the customer, and not an attacker, requesting the recovery, and to communicate this information to Bitkey servers.

During enrollment, RCs receive an invite link from their friend, family member, or colleague who is a Bitkey customer and download the Bitkey app. Once they’ve downloaded the Bitkey app and accepted the invite, the app generates an encryption key and stores it locally on the RC's phone and, with the RC's permission, automatically backs it up to the RC's cloud storage, using the same storage mechanisms described in the Cloud Backup section above (iCloud key-value store for iOS, app-specific storage on Google Drive for Android). The public portion of this key is used by the customer’s app to encrypt a copy of the customer’s mobile app key, which is stored in the customer’s cloud storage – the exact details of this are covered in the sections below.

When a customer has lost their mobile app and their hardware device, they can start Recovery Contacts, which consists of the following steps:

1. Customer downloads the Bitkey app on their new phone and logs into the cloud account they’d previously used with Bitkey
2. When the customer begins Recovery Contacts, the Bitkey app will show the customer a six-digit alphanumeric code
3. The customer reaches out to their Recovery Contact(s) (RCs) and asks them to enter this code into the RC's instance of the Bitkey mobile app. RCs will not receive a notification that a recovery has been initiated – it is the responsibility of the customer to contact them
4. The RC confirms that they are speaking to the customer, not an impostor, prior to entering the code
5. Once the RC has entered the code into their Bitkey app, the RC's app uses a secret key stored on the RC's mobile device (and in their cloud storage) to inform Bitkey servers that the Recovery Contacts should proceed
6. Bitkey servers facilitate communication between the customer and RC apps, allowing the customer’s app to get the information needed to decrypt the backup of their own customer mobile app key that was created during Recovery Contacts setup.
7. At this point, the customer has now regained control of one of the two customer keys (their App Key), and can use the Delay and Notify feature to regain access to their second and final key (the hardware key) to regain access to their funds

Recovery Contacts: Design Properties

- **Relies on cryptographic keys, not secrets humans need to remember**
  - Recovery Contacts' verification of a customer leverages a secret stored in their Bitkey app (and their cloud)
- **No reliance on personal information for "verification" purposes**
  - In contrast to traditional financial products and many attempts at similar solutions in the bitcoin and crypto space, this design lets a customer choose who will verify them, rather than asking or forcing them to provide extensive identity information to Block
- **No reliance on a manual "verification" process that could be bypassed by Block**
  - Recovery Contacts hold the information needed to decrypt Recovery Contacts App Key backups that the customer has possession of; Block is just a communications facilitator
- **Block never sees the customer's App Key**
  - Although Block is involved in facilitating communication of cryptographic material between a Bitkey customer and their Recovery Contacts, Block never sees the App Key that the customer is trying to recover using Recovery Contacts.
- **No individual cloud account (customer or RC) is a single point of failure**
  - An attacker who compromises a customer's cloud account cannot move funds without the cooperation of one of their RCs or the compromise of one of the RC's cloud accounts. An attacker who compromises a RC's cloud account cannot move funds without the cooperation of the customer or a subsequent compromise of the customer's cloud account.

Recovery Contacts: Inner Workings

The Recovery Contacts process consists of the following major steps: Setup, Customer Verification, and Restoration:

- During the Setup step of Recovery Contacts, the customer's Bitkey app generates an encrypted backup of their App Key and stores it in their cloud storage -- this process is similar to the Cloud Backup process, but instead of conducting the encryption with a key held in the customer's Bitkey hardware, it is conducted using the public portion of an asymmetric key generated by each Recovery Contact's Bitkey app.
- During the Customer Verification step, Recovery Contacts verify that it is the customer, and not an impostor, requesting recovery.
- During the Restoration step, the customer's Bitkey app and their Recovery Contact's Bitkey app work together to decrypt the backup.

Setup

The desired result of the setup process is that:

- the customer's cloud contains an encrypted backup of their App Key, along with encrypted copies of the key needed to decrypt the backup

- their Recovery Contacts' cloud accounts each contain a unique key that the customer can use to - through several steps - decrypt the backup of their App Key

The keys involved in the setup process are detailed in the table below, and the setup process is depicted below in Figure 3.

Name	Description	Storage
PKMat	Private key material: Customer's App Key to be backed up and subsequently decrypted using Recovery Contacts	Local Keychain / Keystore
PKEK	Private Key Encryption Key, used to encrypt the private key material, PKMat. It's then encrypted by the public portion of the keypair of each Recovery Contact participating in the recovery (RCEK_i).	Local Keychain / Keystore
RCEK_i	An asymmetric keypair generated by each Recovery Contact. The public portion of this keypair is used to encrypt the backup material during Recovery Contacts setup, and the unique private key held by each Recovery Contact is used to decrypt backup material during Recovery Contacts restoration. Customers may choose more than one Recovery Contact; we use RCEK_i to denote the i'th Recovery Contact's key.	RC Cloud

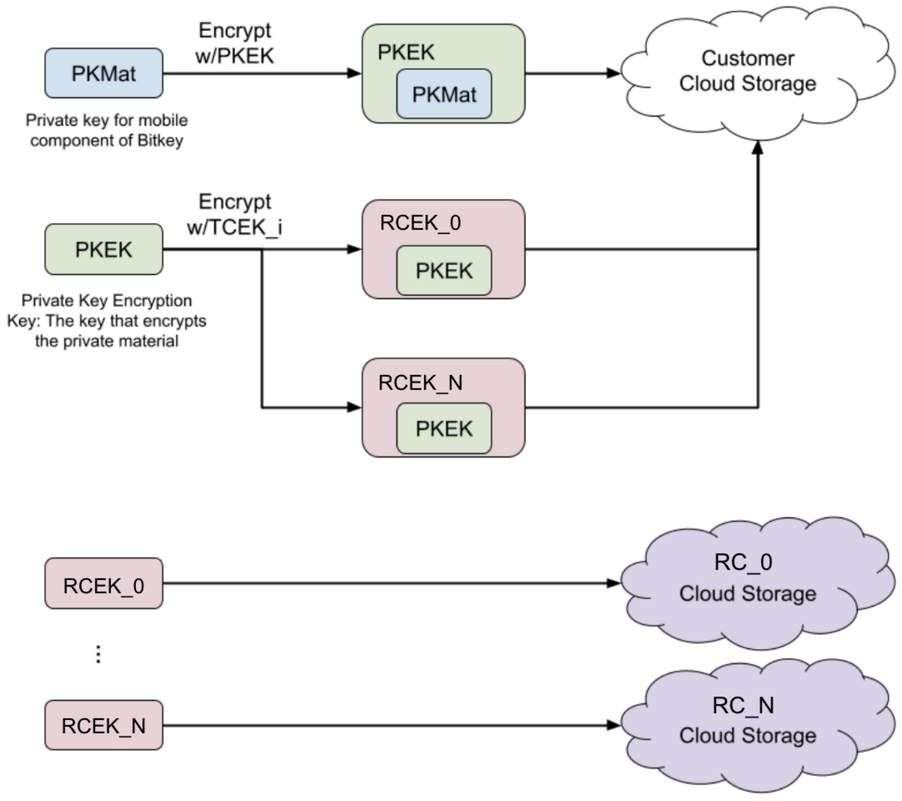

Figure 3: keys and data involved in Recovery Contacts setup.

Restoration

The aim of the restoration process is for the customer's Bitkey app to decrypt the App Key backup with assistance from one of their Recovery Contacts' Bitkey apps. The process consists of the following steps:

- Customer downloads the following from the cloud:
    - `PKEK(PKMat)`
    - `RCEK_0(PKEK) … RCEK_N(PKEK)`
- Customer sends `RCEK_0(PKEK)...RCEK_N(PKEK)` to Block.
- For each Recovery Contact involved in the recovery (just one RC in our current implementation):
    - Block delivers `RCEK_i(PKEK)` to RC
    - RC decrypts `RCEK_i(PKEK)` resulting in `PKEK`
    - RC delivers `PKEK` to Block.
    - Block delivers `PKEK` to the Customer.
- The customer decrypts `PKEK(PKMat)` with `PKEK` resulting in the fully restored `PKMat`.

Future updates to Bitkey documentation will elaborate on how enrollment works and on the details of how the customer's app and their Recovery Contacts' apps interact with Block servers.

Recovery Contacts: Limitations

An attacker who gains access to the customer's cloud account can begin Recovery Contacts (provided they'd chosen to use this optional feature), but can't complete the recovery process without compromising additional assets – they must also gain access to a secret held by a Recovery Contact, for example by ***also*** having:

- tricked or coerced a Recovery Contact into entering the code into their Bitkey app, or
- gained control/contents of the Recovery Contact's cloud account

This means an attacker with access to the customer's cloud account *and* access to the Recovery Contact's secret can conduct a Recovery Contacts and gain access to one of the two customer keys, or can spend funds up to any optional mobile limit configured by a customer – but they are still not yet able to access any funds above a mobile limit if one was configured, and are not able to access any funds at all if no mobile limit was configured.

At this point in the attack chain, an attacker must either:

1. Gain direct access to Block's backend keys (e.g. by compromising Block infrastructure, or by successfully compelling Block to share or misuse a key via a legal process)
2. Attempt to use the Delay and Notify recovery process (using the one key they've gained access to) in order to move the customer's funds to a new, attacker-controlled wallet

Using the Delay and Notify feature will result in notifications to the communications channel (email or SMS) that the customer provided during recovery, and an opportunity for the customer to use their hardware key and communications channel to regain sole control of their wallet.

An attacker with access to the customer's cloud account, access to the Recovery Contact's secret (either via their phone or cloud storage), a new Bitkey hardware device, *and* the ability to disable notifications for the Delay and Notify feature (e.g. by compelling Block or its employees) can steal funds.

To illustrate the above, consider the following initial attack scenarios -- none of which is sufficient to immediately steal funds:

- Large batch of compromised cloud accounts exposes accounts of both customer and one of their RCs to compromise
- Government actor who can (legally or otherwise) either request access to both customer and RC cloud accounts (or gain the participation of the RC)
- Rogue employee(s) at one or more cloud providers who manage to circumvent internal controls to gain control of both customer and RC cloud accounts

To successfully steal all of a customer's funds in any of these example scenarios, an attacker must subsequently use the Delay and Notify recovery process without detection by the customer. This could happen for example if a customer's only notification method was an email that is now attacker-controlled, if a rogue employee at Block disables notifications for the Delay and Notify feature, or if a government successfully compels Block to reduce the security of the Delay and Notify feature.

Note that we're continuing to develop in this area, both in service of further preventing accidental loss for a broad set of customers and in service of further hardening against sophisticated attack scenarios. If you want to provide feedback to the Bitkey team on the tradeoffs we're striking in this area (or any areas of our design), reach us at bitkey@block.xyz

Recovery Contacts: Cheat Sheet

	When it helps you	How you use it	How it works
Recovery Contacts	When you lose access to *both*: ● your hardware device ● the key stored locally on your phone Note: this feature is not enabled by default, you must opt-in to use it.	Install the Bitkey mobile app on your new phone and log into your cloud account. Reach out to your Recovery Contact(s) to provide them with a recovery code – once they enter this code, you'll regain access to your App Key. Finally, use the Delay and Notify feature with a new Bitkey hardware device to regain access to your wallet.	During Recovery Contacts setup, the Bitkey mobile app stores an encrypted backup in your cloud account, and stores the key to decrypt it with your Recovery Contact(s). When you need to use Recovery Contacts, your Recovery Contact(s) will confirm your identity by speaking or seeing you (e.g. phone, video, in-person) and Bitkey servers will facilitate communication between your apps to help you decrypt the backup and regain access to your App Key.

Emergency Exit Mode

Using the Bitkey mobile app is an easy and convenient way to own and manage your bitcoin – but installing the mobile app from the Google Play Store or Apple App Store can't be the *only* way to move your money, or you could lose access to your funds if Block's app were ever unavailable, which would undercut the control provided by self-custody. We want to ensure that customers can move their bitcoin even if:

- The Bitkey mobile application is removed from Apple or Google app stores in their region *at the same time* the customer happened to delete their app and thus cannot re-download it at that point
- Customers cannot use or no longer trust Block's mobile application for any reason, and thus do not want to use it to move their funds to another platform

These goals are unique to self-custody; planning for them in advance is key to making sure that customers are in control in many different scenarios, however unlikely they might seem.

To achieve the goals above, we are including a feature we currently refer to as **Emergency Exit Mode**, which allows customers to move their funds by constructing and signing a transaction without relying on downloading the Bitkey app from the Google Play Store or Apple App Store. We don't want customers to have to use this, but we know that some may need to (or want to).

During the onboarding process when a customer sets up their wallet, the Bitkey app automatically creates an Emergency Exit Kit ("EEK") and stores it in the customer's cloud storage, using iCloud Drive and Google Drive storage mechanisms that result in files visible to the user, unlike the other backup functionality discussed in this document.[2] This kit contains an encrypted backup of the App Key as well as information needed to move funds without needing to download the official app store Bitkey app. Unlike with a seed phrase, an attacker who sees the contents of the EEK cannot move funds – the EEK must be used along with the customer's unlocked Bitkey hardware device, through the process described below, in order to move funds.

If Apple or Google remove the Bitkey app from the App Store or Play Store in a given country - or even globally - some affected customers may want to stop storing their funds using Bitkey. Any customers who still have the Bitkey app installed on their mobile device at this point in time could use it to move funds to another wallet. However, any customers who no longer have the app on their mobile device (e.g. because they deleted it or because the mobile platform operating system removed it for any reason) would need to use Emergency Exit Mode in order to move their funds.

[2] This is because the storage mechanisms used for Cloud Backup and Recovery Contacts can only be accessed by the Bitkey app, and thus cannot be used for Emergency Exit Mode as the official app store release of the Bitkey app is not involved.

Emergency Exit Mode: Process

The primary purpose of the Emergency Exit Mode process is to provide a substitute for using the officially-distributed Bitkey mobile app to construct a transaction and sign it using the App Key. This is accomplished by:

- storing an encrypted copy of the App Key in the Emergency Exit Kit,
- providing transaction construction and signing capability in a sideloaded version of the Bitkey mobile application that is available through mechanisms other than mobile providers' app stores,
- relying on the same functionality used by the Bitkey hardware device to sign a transaction under normal operation

Additionally, the Bitkey app automatically creates and stores a new Emergency Exit Kit whenever a customer's App Key changes – this happens for example when a customer goes through a Delay and Notify recovery process after losing both their phone and access to their cloud backup. The Emergency Exit process can be used multiple times, if needed, and the Bitkey hardware involved can subsequently be used to interact with the officially-distributed Bitkey app again.

The Emergency Exit Kit contains:

1. A QR code containing a link to sources where historical APK/IPA files of the Bitkey app are hosted, independent of the Bitkey team / Block (and a printed text copy of this link)
2. A reference to a specific older version of the Bitkey App compatible with the Emergency Exit Kit, with a hash to use to verify the contents of a matching APK/IPA to be used in sideloading.
3. An encrypted Cloud Backup as described earlier in this document (i.e. contains an encrypted copy of the App Key, along with extended public keys for all 3 keys involved in the multisig)

To use Emergency Exit Mode, customers:

1. Must have their Emergency Exit Kit, as well as their Bitkey hardware and the ability to unlock it
2. Locate a source for the Bitkey app in APK (Android) or IPA (iOS) format. The EEK will link to at least one location where these can be found. Follow this link to download the file directly to the mobile device it is to be installed on, which can later be found through the native "Files" application (iOS and Android)
3. (Strongly Encouraged but Optional) Verify the contents of the downloaded APK/IPA against a hash of the contents of the same app version at time of EEK creation (included in the EEK), using tools like Hashdroid or QuickHash.
4. Prepare their mobile device to sideload an application
    a. Android - Enable (at least temporarily) installation from unknown sources in the Android settings menu
    b. iOS - Use a PC to install an application such as AltStore, which can assist with managing further sideloading app installs. If a PC is not available or not

preferred, use an Android device and follow the Android instructions for using Emergency Exit Mode.

5. Install the Bitkey App downloaded and verified earlier in this process by navigating to the application file using the native “Files” application and selecting the file
    a. Android - Use “App viewer” to view and then launch the application
    b. iOS - Use the installed “AltStore” application to open the file and install the Bitkey app
6. Open the Bitkey app, and proceed to restore from backup by importing the App Key from a scan of the QR included in the Emergency Exit Kit
7. Decrypt this backup using a tap from the unlocked Bitkey hardware
8. If desired or necessary, use the Bitkey app settings menu to configure a different customer electrum server (e.g. if default Block-provided and public sources are unavailable)
9. Use the sideloaded Bitkey app to create,sign, and send a transaction for all available funds, signed using customer App Key and Hardware Key, to an arbitrary destination outside of Bitkey
10. Wait for confirmation of the transaction. The transaction will initially be displayed as ‘pending’ in the Bitkey app - this indicator will be automatically removed when the transaction has been confirmed.

This process is what we are starting with – but it is not the only way, and others may choose to build additional mechanisms that customers could use when they need to use Emergency Exit Mode. In our forthcoming GitHub repository, we will publish technical information that will help others build additional ways for people to move money using a EEK and Bitkey hardware device.

Emergency Exit Mode: Limitations

No solution, including Bitkey’s, covers every possible threat perfectly. In the case of Bitkey’s Emergency Exit Mode, we note the following limitations:

- Mobile platform operators and handset manufacturers could theoretically interfere with people’s ability to conduct the Emergency Exit process
    - For example, by blocking applications from using the NFC interface, filtering out NFC data specific to Bitkey, or attempting to prevent people from accessing the Emergency Exit Mode Interface code whether it takes the form of a sideloaded application as described in this section or other forms in the future. But their ability to interfere greatly diminished when the methods used for Emergency Exit Mode are generic, can be deployed by many entities other than Block, and can be accessed by customers using platforms other than mobile handset hardware if necessary.
- Customers must have unlocked Bitkey hardware
    - If customers lose access to their Bitkey hardware *and* are not able to install/reinstall the Bitkey app, they cannot access their funds

- Relies on the customer's cloud storage
    - While customers could in theory make digital or physical copies of the EEK, the default behavior is to rely on general cloud storage. These materials could be deleted either accidentally by a customer, or possibly intentionally by a cloud storage operator. Customers could make additional copies of these materials, but would need to update their copy any time they lose both their App Key and their cloud backup (because then the App Key, which is included in the EEK, will change)
- Funds sent to the wallet after using Emergency Exit Mode can only be recovered using the same Bitkey hardware and Emergency Exit Kit
    - This is only an issue if an address associated with the wallet is reused either by the customer or by someone they previously shared the address with
- Using Emergency Exit Mode consists of a potentially confusing, technical process
    - While some customers who may need to use Emergency Exit Mode will already be familiar with sideloading an app, others may be doing this for the first time – making this hard to use even with numerous high quality public tutorials about the topic. Bitkey's Emergency Exit approach attempts to limit the use of difficult workarounds to the rare situations they're intended for, rather than subject every customer to complicated processes and additional responsibility up front.

Emergency Exit Mode: Cheat Sheet

	When it helps you	How you use it	How it works
Emergency Exit Mode	When you want to migrate off of Bitkey because you can no longer: • Access the Bitkey mobile app on app stores and you don't have the app already on your phone when this happens • Trust Block's mobile app, and thus do not want to use it to move funds to another platform Note: you must have your Bitkey hardware in order to use this feature.	Retrieve an Emergency Exit Kit file that was automatically stored in your cloud storage during wallet setup. Follow the instructions it contains, which will direct you to install ("sideload") a version of the Bitkey mobile application from a source other than the mobile platform app store you're typically using on your phone. Use this application along with your Bitkey hardware to generate a transaction that will move your funds to a destination bitcoin address of your choosing.	This process uses the same transaction signing and cloud backup decryption capabilities provided by the hardware during normal operation. The Emergency Exit Kit contains an encrypted backup of the App Key, which is decrypted using the hardware so that both customer keys can be used to sign a transaction.

Recovery Features Cheat Sheet

	When it helps you	How you use it	How it works
Cloud Backup	When you get a new phone or delete the app and need to reinstall it.	Install the Bitkey mobile app on your new phone and log into your cloud account used to initially create your wallet. The Bitkey app will direct you to unlock and tap your hardware to regain access to your wallet.	The Bitkey mobile app stores an encrypted copy of the App Key in the customer's cloud storage account. The copy is encrypted by the customer's Bitkey hardware prior to storage, and must be decrypted by the customer's Bitkey hardware during recovery.
Delay and Notify	When you lose access to either: ● your hardware device, or ● both the cloud account you used to set up Bitkey and the App Key stored on your phone	Begin the recovery process by reporting a lost hardware device from your existing Bitkey app, or if you lost your phone and cloud backup, by using your hardware device with the Bitkey app on your new phone. If you've lost your hardware device, you'll need to order a new one. In both cases, after the security waiting period elapses, assuming you don't cancel the request for any reason (like you found it again), you'll have access to your funds in a new Bitkey wallet.	After a delay, Bitkey servers co-sign a transaction to move funds to a new wallet, alongside whichever key you still have control of. Bitkey servers may require you to prove control of the customer communications channel you used during signup (email or SMS), in the rare case that there are two separate customer keys attempting recovery at the same time.
Recovery Contacts	When you lose access to *both*: ● your hardware device ● the key stored locally on your phone Note: this feature is not enabled by default, you must opt-in to use it.	Install the Bitkey mobile app on your new phone and log into your cloud account. Reach out to your Recovery Contact(s) to provide them with a recovery code – once they enter this code, you'll regain access to your App Key. Finally, use the Delay and Notify feature with a new Bitkey hardware device to regain access to your wallet.	During Recovery Contacts setup, the Bitkey mobile app stores an encrypted backup in your cloud account, and stores the key to decrypt it with your Recovery Contact(s). When you need to use Recovery Contacts, your Recovery Contact(s) will confirm your identity by speaking or seeing you (e.g. phone, video, in-person) and Bitkey servers will facilitate communication between your apps to help you decrypt the backup and regain access to your App Key.
Emergency Exit Mode	When you want to migrate off of Bitkey because you can no longer: ● Access the Bitkey mobile app on app stores and you don't have the app already on your phone when this happens ● Trust Block's mobile app, and thus do not want to use it to move funds to another platform Note: you must have your Bitkey hardware in order to use this feature.	Retrieve an Emergency Exit Kit file that was automatically stored in your cloud storage during wallet setup. Follow the instructions it contains, which will direct you to install ("sideload") a version of the Bitkey mobile application from a source other than the mobile platform app store you're typically using on your phone. Use this application along with your Bitkey hardware to generate a transaction that will move your funds to a destination bitcoin address of your choosing.	This process uses the same transaction signing and cloud backup decryption capabilities provided by the hardware during normal operation. The Emergency Exit Kit contains an encrypted backup of the App Key, which is decrypted using the hardware so that both customer keys can be used to sign a transaction.