

Bitcoin Self-Custody Guide

Self-Custody Guidelines Built From Real Incidents and Best Practices

Received: 2026-07-28

Accepted: 2026-09-15

About

Metanomia is a think tank
studying crypto markets and technology.

© 2026 Metanomia. All rights reserved.

Researcher

Jeewan Lee

Corresponding Author

Taemin Oh

Contents

01	Two Incidents	You Can Lose Your Funds Even When You Do Nothing Wrong Poor Mnemonic Management Can Lead to Loss Even Without Hacking Who Needs Self-Custody and Its Value	3
02	Where Do Incidents Occur?	Generation Phase: Insufficient Randomness Weakens a Wallet From the Start Backup Phase: The Recording Method Becomes the Exposure Vector Storage Phase: Physical Location Defines the Attack Surface Signing Phase: Computer Screens Are Not Ground Truth Recovery Phase: An Unverified Backup Is Not a Backup Human and Physical Realities	6
03	How Institutions and Governments Manage Keys	Lesson 1: Control the Environment at the Moment of Key Generation Lesson 2: Make Sure No Single Person Can Move Assets Alone Lesson 3: Never Concentrate Assets in a Single Location Lesson 4: Turn Critical Operations Into Scripts, and Keep Records That Prove They Were Followed Lesson 5: Verify Against External Standards, Not Your Own Judgment	10
04	Individual Principles of Self-Custody		13
05	Practical Measures for Individuals	Choosing a Setup: Passphrases and Multisig Generation Phase: The Environment Where You Make the Key Backup Phase: How to Record the Mnemonic Storage Phase: Where to Keep Things Signing Phase: What to Verify Recovery Phase: Test the Full Procedure With a Small Amount Using AI as a Verification Aid A Response Plan for Leaks and Loss	14
	Conclusion		31
	Notes		32

Two Incidents

In 2026, two separate incidents highlighted the risks of self-custody from different angles. One involved a flaw in a key generated by a hardware wallet; the other involved a government agency that ended up disclosing the recovery phrase of a wallet it had seized.

Together, these two incidents illustrate a fundamental reality of self-custody: all responsibility for managing the wallet falls on the owner. Whether funds are lost through a personal mistake or a manufacturer defect, there is no company to reimburse the owner, and there is virtually no insurance that broadly covers personal self-custody. Yet there are people who have no choice but to use self-custody, and this guide is written for them.

Self-custody cannot necessarily be considered safer than entrusting assets to a trusted institution. With that in mind, this guide looks at how institutions and governments are addressing the challenge of safeguarding crypto assets and uses those approaches as a reference point for outlining practical safeguards that individuals can realistically manage. Readers who are already using self-custody and need an immediate checklist can start with Chapter 5.

You Can Lose Your Funds Even When You Do Nothing Wrong

On July 30, 2026, Bitcoin began disappearing from the wallets of Coldcard hardware-wallet users. Within just 25 minutes, approximately 594 BTC, worth about \$38 million at the time, disappeared from around 500 wallets. Three more attacks followed the initial attack, resulting in a total of four thefts. In all, approximately 1,816 BTC, worth about \$116 million at the time, disappeared from more than 5,200 addresses.¹

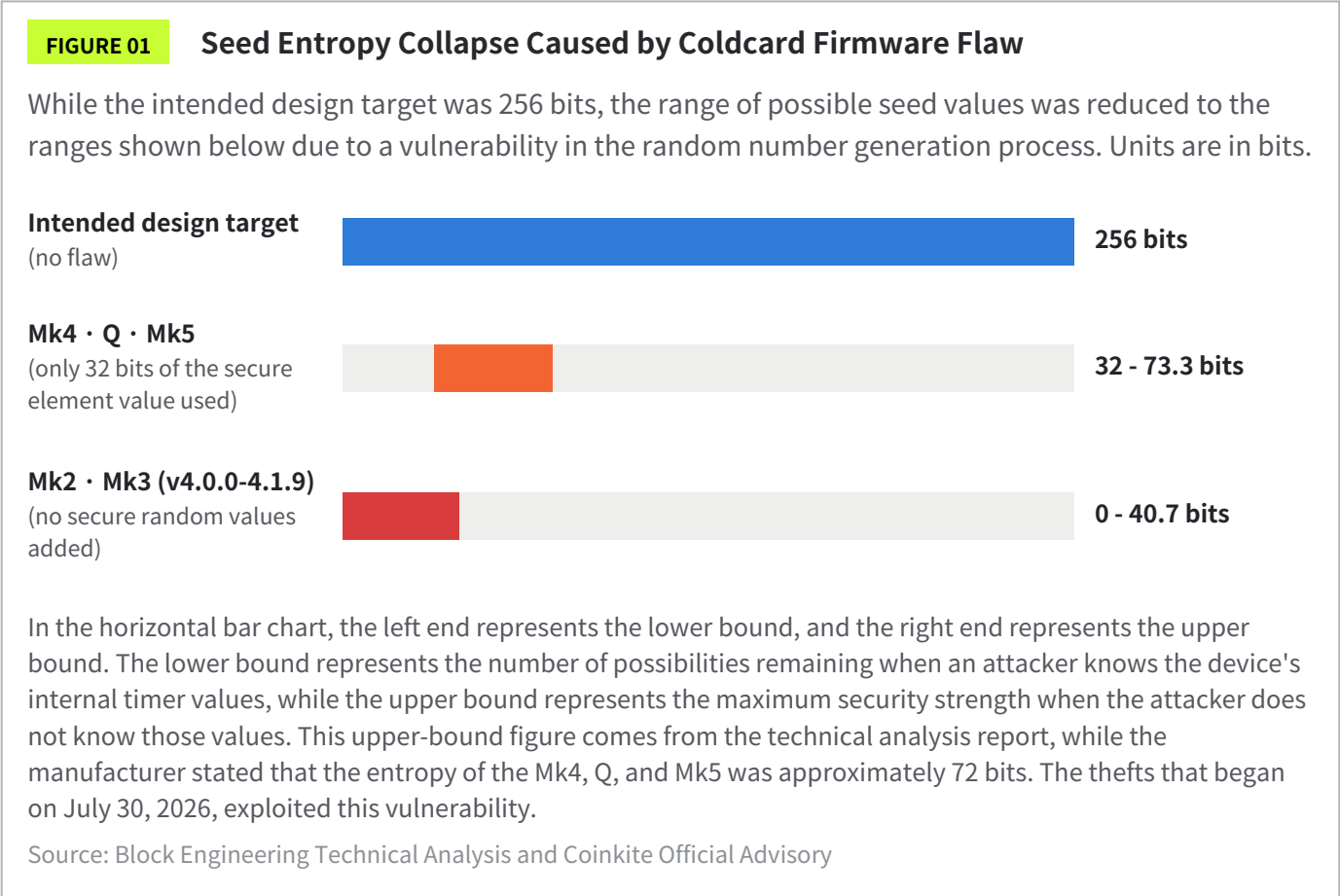
The cause was not user error. A flaw in the seed-generation process was introduced in firmware 4.0.0, released on March 17, 2021. As a result, the range of seeds that could be generated was significantly narrower than the intended 2^{256} possibilities.

According to an analysis by fintech company Block, Coldcard's older models, Mk2 and Mk3, failed to add cryptographically secure random values when generating seeds. The subsequent models, Mk4, Q, and Mk5, also failed to make sufficient use of the values obtained from their secure elements, incorporating only 32 bits of them into the random-number generator. Although 32 bits can distinguish roughly 4.3 billion possibilities, this is extremely small by cryptographic standards.

The actual difficulty of the search varied depending on information known to the attackers, including device identifiers, boot times, and the history of random-number-generator calls. The attackers reproduced the vulnerable random-number-generation process on their own computers to generate candidate seeds, then compared the addresses derived from those seeds with wallet addresses publicly available on the blockchain. Once they found a matching address, they could use the corresponding seed to move the funds

in the wallet. Because the flaw went undetected for more than five years, users who had generated seeds with affected devices during that period had little way of knowing whether they were exposed to the risk.²

Ironically, Coldcard had long been known as a security-focused brand, and many individuals who trusted that reputation ended up losing their crypto. Coldcard’s manufacturer, Coinkite, had promoted transparency by making its firmware source code public, and the vulnerable firmware had remained publicly available since March 2021. Yet users had no tool to check whether their own firmware was vulnerable. Instead, the publicly available code gave hackers material they could use to search for vulnerabilities. There was even speculation that the attack may have been carried out not by a human but by an AI language model.³



Poor Mnemonic Management Can Lead to Loss Even Without Hacking

On February 26 of the same year, South Korea’s National Tax Service released a press statement highlighting its achievements in collecting back taxes from taxpayers with large outstanding tax debts. The press release included a photograph of a seized cryptocurrency wallet. Visible in the photograph, without any redaction, were the 24 words required to recover the wallet, known as its mnemonic phrase.⁴ Within a day of the material being made public, all 4 million PRTG tokens in the wallet, worth approximately KRW 6.9 billion (approx. USD 5.2 million) based on the exchange-listed price at the time, were transferred to another wallet. On February 28, an individual claiming responsibility returned the entire amount, but about two and a half hours later, the tokens were transferred again to another wallet.⁵

No sophisticated hacking techniques were involved in this incident. All it took was reading the publicly released photograph. Although the incident involved an Ethereum-based token rather than Bitcoin, the importance of proper mnemonic management is no different for Bitcoin. Even a government agency can

make this kind of mistake. The possibility of individuals making the same mistake should be taken even more seriously.

Who Needs Self-Custody and Its Value

Even so, there are people who have no choice but to use self-custody. They include people who want to hold their assets directly rather than entrust them to anyone else, those who have circumstances that make it difficult to complete the Know Your Customer (KYC) procedures required by exchanges, and those who use Decentralized Finance (DeFi) services that require them to sign transactions directly with their own wallets. Babylon staking while keeping Bitcoin in one's own wallet, as well as trading Ordinals and Runes, which requires signing transactions directly with one's own Bitcoin wallet, cannot even be started without self-custody.⁶ These services are still relatively small markets, but for people who want to use them, self-custody is not optional.

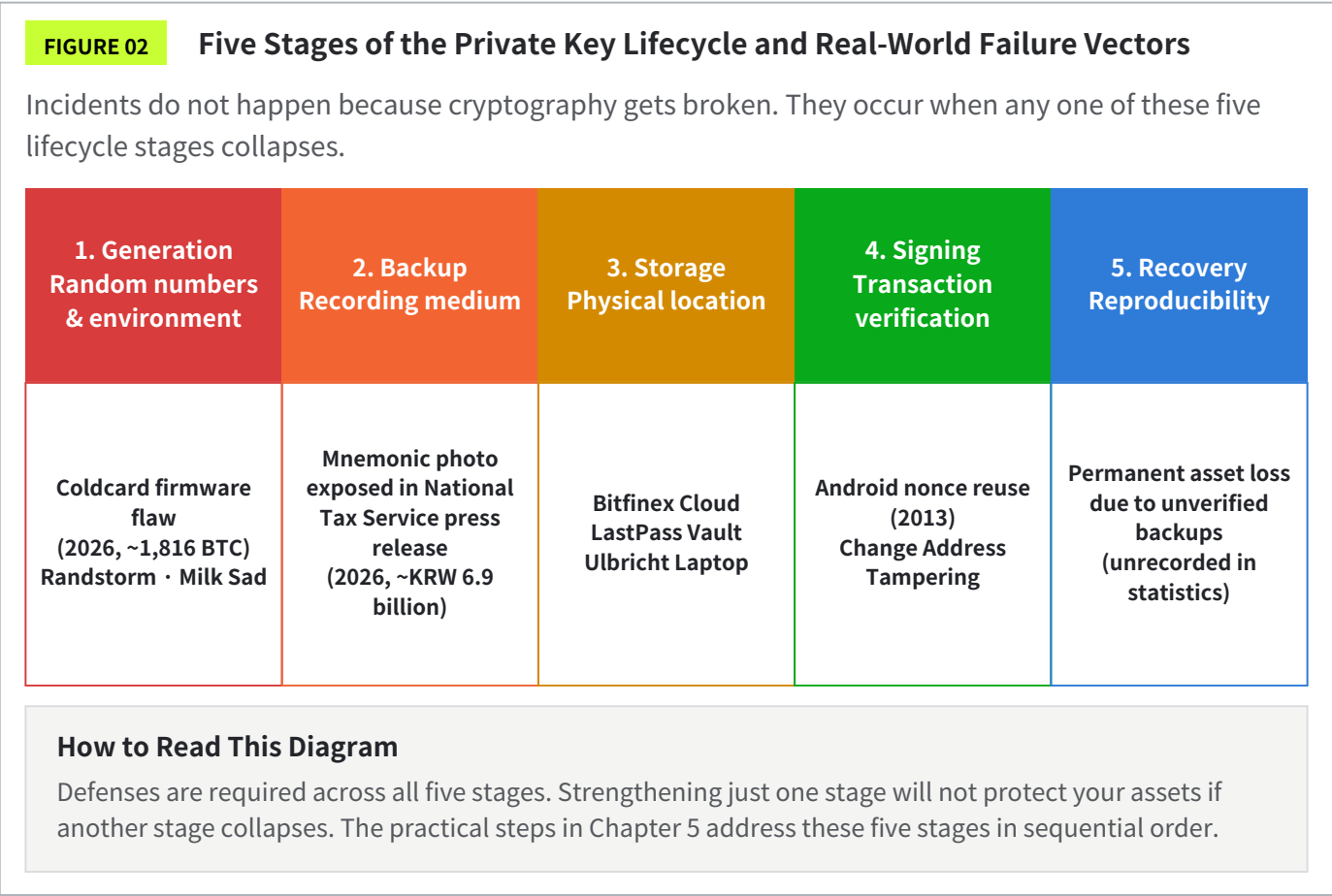
Self-custody also provides clear benefits to individuals. Cryptocurrency held on an exchange or with a custodial company is simply a balance shown on a screen—in other words, a claim against that company. If the company collapses, suspends withdrawals, or freezes an account, the balance may remain on the screen even though the crypto cannot be accessed. In recent years, the collapse of major exchanges has repeatedly left users unable to recover their assets. By holding the keys yourself, you alone have the authority to move your assets. The fact that your assets are not dependent on another party's operations or policy changes is the practical benefit of self-custody.

Self-custody also has significance for the ecosystem as a whole. When Bitcoin becomes concentrated in a small number of large companies, those companies become new single points of failure. The greater the concentration, the wider the impact when one of them is breached or collapses, while large amounts of Bitcoin gathered in one place become major targets in themselves. Conversely, when holdings are distributed among multiple individuals, an incident at one location does not spread across the entire ecosystem. This is also why the institutions discussed in Chapter 3 avoid concentrating their assets in one place.

These benefits, however, do not provide a reason to recommend self-custody. Starting without adequate preparation increases the risk of losing assets, and that loss falls entirely on the individual. This guide does not recommend self-custody; it is written to help those who decide to use it minimize their risk of loss.

Where Do Incidents Occur?

The incidents discussed in Chapter 1 are, in fact, nothing new in the history of Bitcoin. They are simply examples of the types of incidents that have occurred repeatedly in the past. A private key has its own lifecycle: it is generated, recorded, stored, used for signing, and restored when needed. Almost all known self-custody incidents have occurred at some point in these five stages. There are no known cases of assets being lost because Bitcoin’s cryptographic algorithms themselves were broken.



Generation Phase: Insufficient Randomness Weakens a Wallet From the Start

The security of a private key rests on the premise that its value is unpredictable and random. If this premise fails, subsequent defenses weaken along with it. The Coldcard incident described in Chapter 1 is exactly this type of failure. Even if a user kept the device strictly offline and backed up the seed on a metal plate, those precautions alone could not protect the assets if randomness was lacking at the moment of seed generation. Adding a passphrase (an extra password known only to the user, appended to the seed) can make an attack

harder in proportion to its strength, but it does not eliminate the underlying vulnerability of the seed itself. Because security then comes to depend on the passphrase alone, the overall setup remains fragile.

Similar problems have occurred before. The randomness vulnerability that arose when certain web wallets generated private keys inside the browser, between 2011 and 2015, is called Randstorm. The JavaScript library used at the time relied on a weak random function in some browsers, narrowing the range of possible private keys. As a result, some wallets created during this period were found to have private keys that attackers could reconstruct.⁷

A similar problem was later found in another program. The vulnerability discovered in the seed-generation feature of Libbitcoin Explorer, a cryptocurrency wallet tool, is called Milk Sad (CVE-2023-39910). When generating a seed, this program converted the computer's system clock into a 32-bit value and used it as the starting value for the random number generator. As a result, the number of possible seeds was limited to 2^{32} — about 4.3 billion — a small enough space to brute-force with a high-performance personal computer.⁸

What Individuals Can Do at This Stage

There is no way for an individual to verify that a device's random number generator is working properly. Instead, using a device that lets you manually input random values from dice or coin flips means that the randomness you provide remains intact even if the device's RNG is broken. In the Coldcard incident, users who rolled dice enough times to generate their seed were excluded from those at risk. This is why dice input is recommended in Section 5.2.

Backup Phase: The Recording Method Becomes the Exposure Vector

The moment a seed is recorded is the second weakness. The National Tax Service incident discussed in Chapter 1 is an extreme example: assets were drained simply because a photograph of paper with a mnemonic phrase written on it was published in an official press release.

For individuals, a more common exposure path is storing the seed on digital devices. In the 2022 LastPass breach, attackers got hold of encrypted password vaults for millions of users, then cracked master passwords one at a time. What the victims had in common was that they had written their seed phrases in the notes field of their online vaults. U.S. law enforcement concluded that this breach led to roughly \$150 million in cryptocurrency theft, and seized approximately \$24 million worth of assets in March 2025.⁹

Keeping a photo of the seed is also dangerous. The SparkCat malware disclosed by Kaspersky searches smartphone photo galleries, reading text to find seed phrases. This malware was distributed without being caught by the review process of official app stores, and used character-recognition models across multiple languages, including Korean, to hunt for strings resembling mnemonics.¹⁰

Storage Phase: Physical Location Defines the Attack Surface

Where keys are physically kept is the third point of vulnerability. In October 2013, when the U.S. Federal Bureau of Investigation (FBI) arrested the operator of Silk Road, agents secured his laptop while it was still powered on and logged in. Because it was seized before locking, investigators gained immediate access to his wallets.¹¹ In another instance involving funds stolen in the 2016 Bitfinex hack, the perpetrator stored

private keys in a cloud account; law enforcement accessed that account in February 2022, leading to the seizure of approximately 94,000 bitcoins.¹²

These cases share one thing in common: the moment keys reside on an online device or in the cloud, their security no longer depends on cryptography, but on the lock protecting that storage space. And a cloud account can be opened either through legal process or through hacking.

Signing Phase: Computer Screens Are Not Ground Truth

The fourth point of vulnerability is when signing transactions. Bitcoin signatures use a one-time random number called a nonce; if the same nonce is used twice, an attacker can calculate the private key from the public signatures alone. In August 2013, the Bitcoin project issued a warning that a flaw in Android's random number generator caused certain wallet apps to repeatedly generate the same nonce, exposing funds to theft.¹³ Following this incident, support for deterministic nonce generation (RFC 6979) became an important criterion when choosing a hardware wallet, since this method prevents the same nonce from being reused across different transactions.

Another threat is change-address tampering. In Bitcoin, leftover funds from a transaction return to your wallet as change. If malware on an infected computer swaps out this return path with the wrong value, you won't be able to retrieve the funds even if you hold the key, so long as you don't know the correct path.¹⁴ The principle of defense is simple: verify transaction details before signing on the hardware wallet's screen, not the computer screen.

Recovery Phase: An Unverified Backup Is Not a Backup

The final point of vulnerability is recovery. Incidents at this stage rarely make the news, since there's only one victim and no perpetrator. This is when someone loses access to their own funds because they recorded the seed words incorrectly, misremembered a passphrase, or lost their multisig configuration information. It doesn't show up in statistics, but this may well be the most common way people actually lose assets in self-custody.

The nasty part of this stage is that years can pass between when the mistake happens and when it's discovered. Someone who recorded their backup incorrectly only finds out when they actually need to recover — and by then it's too late. This is exactly what the full-procedure testing with small amounts in Section 5.6 is meant to catch.

Human and Physical Realities

There is risk beyond these five stages as well. In June 2020, the online store database of hardware wallet maker Ledger was breached, exposing detailed records — including customers' names, addresses, and phone numbers — for about 272,000 people, along with roughly 1 million email addresses. Ledger disclosed the breach in July, but didn't correct and confirm the full scale of the exposed records until that December.¹⁵ Private keys themselves weren't compromised. But the fact that criminals now had the real names and home addresses of people known to hold crypto assets created a foothold for physical threats.

Once it's known that someone holds crypto, threats that have nothing to do with technology begin. No cryptographic protection helps once someone threatens you directly, in person, to hand over the seed. There are two responses: managing things so that the fact of holding crypto is never linked to your identity and address, and having a decoy wallet ready for use if you're ever threatened. The decoy wallet leads into the passphrase discussion in Section 5.1.

How Institutions and Governments Manage Keys

While the previous chapter examined where security failures occur, this chapter draws lessons for individuals from institutional and governmental practices. Because of audits and certification, institutional key management practices are extensively documented in public disclosures. Rather than listing every technical specification, this chapter selects five lessons that individuals can adapt, pairing each with the case study that best illustrates it. While an individual cannot match institutional resources, the core intent behind them can be fully adopted.

Lesson 1: Control the Environment at the Moment of Key Generation

According to a 2018 report, Coinbase generates its private keys (the secret key that can move the assets in a wallet) inside a tent that blocks electromagnetic signals. This cube-shaped tent, measuring roughly 2.4 meters on each side, is wrapped in signal-blocking mesh to function as a Faraday cage. The setup location is randomly chosen from among several secure areas, and shielded power sources are used to prevent information from leaking out through the power lines as well. Inside the tent are only two laptops that have never once connected to the internet, a folding table, lights, and a printer. A coin toss decides which laptop is used, and once the work is done, both laptops are destroyed. The whole process takes nearly a full day.¹⁶

What this procedure is meant to prevent is clear: no signal should leak out of that space at the moment the key is created, and the device used to generate the key is never reused. An individual can't go as far as setting up a tent like Coinbase does. But the underlying practice can be followed directly — generate the key in a room with every device that transmits signals, records, or takes photos switched off, and leave behind no notes taken during the process. The specific method is in Section 5.2.

Lesson 2: Make Sure No Single Person Can Move Assets Alone

At institutions, moving assets requires approval from multiple people. Generating, backing up, and recovering keys is done by several people gathered together following a predetermined script — a procedure called a key ceremony. Withdrawals also require approval from a set number of people acting together, and important operations are always handled by two or more people together. No one can move assets alone.

Most individuals don't have colleagues available to co-approve. But the core of this principle isn't the number of people — it's eliminating the single point of failure: the one spot that, if breached, brings everything down. Individuals can achieve this with multiple keys — specifically, a multisig (multi-signature) setup, where two of three independent keys must come together to move assets. The specifics are covered in Section 5.1.

Lesson 3: Never Concentrate Assets in a Single Location

El Salvador moved its national Bitcoin holdings into physical vaults disconnected from the internet, then split reserves that had been concentrated in a single wallet across multiple addresses. Reports indicate that approximately 6,284 BTC was split across 14 addresses, with no single address holding more than 500 BTC.¹⁷ The reasons were strengthening security and reducing single points of failure.

For individuals, separate from address diversification, there's a need for a principle of splitting backups, keys, and assets. One part is keeping backups and keys in multiple locations, physically separated from each other; the other is not putting your entire holdings into a single wallet. If you keep a small everyday wallet separate from a long-term storage wallet, a mistake on the wallet you handle often stays contained there.

Lesson 4: Turn Critical Operations Into Scripts, and Keep Records That Prove They Were Followed

Institutions and governments don't leave critical operations to human memory — they manage them by manual. The Swiss Crypto Valley Association's 2020 Trusted Key Ceremony Guidelines recommends turning key-generation procedures into a script. The script records who performs which role and even the version and serial number of the equipment used, and beyond the ceremony leader and operators, it calls for at least one independent witness, such as an auditor or notary. Components are sealed in tamper-evident bags, and evidence that can be checked later — video, photos, participant signatures — is kept. Notably, the guidelines call for scripting even outwardly invisible checks, such as the auditor confirming serial numbers were recorded correctly, and state plainly that any deviation from procedure is logged by the independent witness.¹⁸

Institutions and governments feed the lessons of incidents back into their manuals to strengthen security. South Korea went through the same process after the National Tax Service incident in Chapter 1. The National Tax Service announced it would introduce internal controls to review materials before public release, alongside its apology, and the Board of Audit and Inspection examined how seized assets were managed across four agencies: the Prosecution Service, the National Police Agency, the National Tax Service, and the Korea Customs Service. In April 2026, the relevant ministries jointly issued improvement measures. These include moving seized assets on the spot into institutional wallets disconnected from the internet, having private keys and recovery phrases managed by two or more people, equipping storage locations with safes and CCTV, and running at least one simulated breach drill per year.¹⁹

At the individual level, these examples translate into two practices. First, make a script: for creating a wallet, retrieving a backup, or moving a large sum, number the steps in order ahead of time, and do only what's written. Second, keep records: log the type of hardware wallet used and the date. If something happens that falls outside the plan, don't improvise on the spot — first write down what happened, then decide how to respond.

Lesson 5: Verify Against External Standards, Not Your Own Judgment

Rather than trusting their own belief that they're secure, institutions get verified against outside standards and audits. The leading example is the CryptoCurrency Security Standard (CCSS). Maintained by the non-profit CryptoCurrency Certification Consortium (C4), this standard applies 41 checkpoints to

self-custody systems — examining things like whether keys are generated with sufficient randomness, whether procedures and records are kept from generation through use and disposal, and whether backups and authority are split so that the failure of one person or one point doesn't lead to loss of assets, among other criteria. Added to this are FIPS 140-2 and 140-3, which verify cryptographic modules; SOC 1, which covers controls related to financial reporting; and SOC 2, which covers a service organization's controls over security, availability, and similar areas.²⁰ The scope of application and verification differs across these, but they share one thing: none of them rely solely on internal judgment — all are verified against external standards.

Individuals can't obtain institutional-level certification. The core idea is: don't trust your own judgment alone — put a separate procedure in place to verify it. Once you set up a wallet, always test a small deposit and withdrawal, check whether you can actually recover using the backup alone, recalculate the address shown on your device with a separate tool to confirm it matches, and compare your firmware's hash against the manufacturer's official documentation.

Most of Chapter 5 is devoted to putting this principle into practice.

Individual Principles of Self-Custody

Drawing on the incidents from Chapter 2 and the lessons from Chapter 3, eight principles for individuals have been distilled. The endnotes map which incident and lesson each principle comes from, and which guideline in Chapter 5 it leads to.²¹ These principles lead into the practical measures in Chapter 5.

- **Principle 1. Don't create a single point of failure.** Avoid any setup where the loss of one key, one location, one manufacturer, or one memory wipes out your entire holdings.
- **Principle 2. Control the environment where keys are made.** Generate keys with every signal-transmitting, photo-capable, or recording device switched off, and add randomness you create yourself, such as with dice.
- **Principle 3. Backups should leave no digital trace.** Photos, cloud storage, password managers, email, and messaging apps — all of these are paths where real, large-scale incidents have actually happened.
- **Principle 4. Keep backups and keys geographically separated.** Don't keep the device and the backup securing the same seed in the same location.
- **Principle 5. Confirm transaction details on the hardware wallet's screen — never the computer screen.** Confirm the destination address, amount, fee, and other transaction details directly on the device.
- **Principle 6. Test the full procedure with a small amount before moving a large sum.** A backup you haven't tested by recovering from it isn't really a backup.
- **Principle 7. Verify with independent means, not your own judgment.** Recalculate the values your device shows using a separate tool to confirm they match, and get your reference values from the manufacturer's official documentation.
- **Principle 8. Write your own wallet-management rules and key-management manual.** Keep a record of your wallet usage.

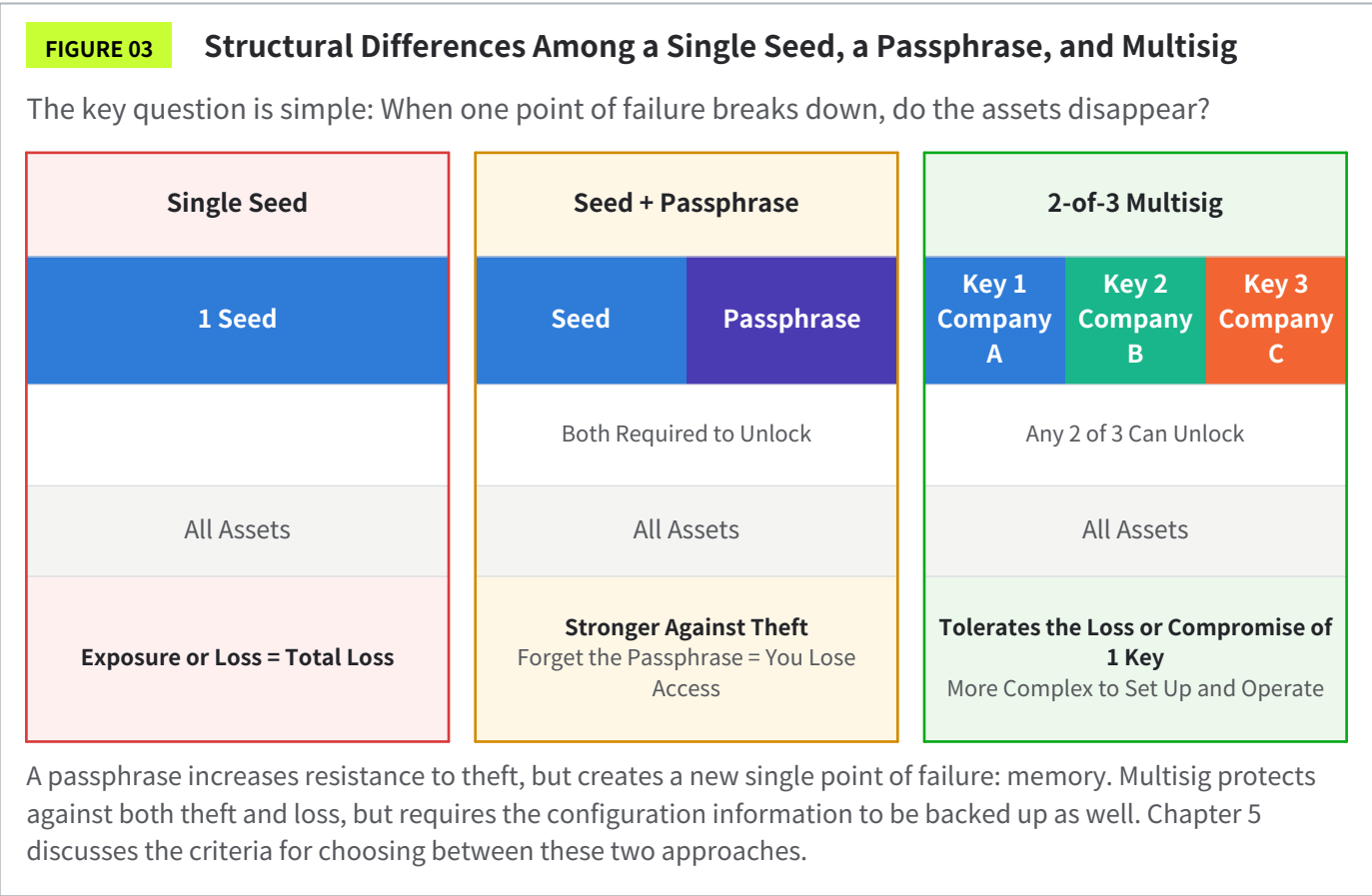
Practical Measures for Individuals

This chapter turns the principles from Chapter 4 into concrete practices. It follows the private key’s lifecycle — generation, backup, storage, signing, recovery — and closes with how to use AI as a verification aid and how to respond to a leak. When choosing a setup, consider your holdings size, your comfort with devices, who will handle recovery or inheritance, and how much time you can put into operations.

Choosing a Setup: Passphrases and Multisig

If you’re new to this, just learning, or only holding a small amount, you can start with a proven single-signature wallet along with backup-and-recovery testing. A single-seed setup is vulnerable if that one seed phrase is leaked or lost, but it’s easier to manage than a more complex setup. Experienced users who need protection against mnemonic leakage should add a sufficiently long, unique BIP39 passphrase, and high-value, more sophisticated setups should consider a thoroughly tested and documented 2-of-3 multisig.

The options break down into three: single-signature, single-signature plus passphrase, and multisig.



Passphrase Setup

A passphrase (BIP39 passphrase) adds one more string — known only to you — on top of a 12- or 24-word seed, producing an entirely different wallet. It's often called the "25th word." Even if someone steals the metal plate with your seed on it, they can't touch the funds without the passphrase.

Its advantage is simplicity: one device is enough, and there's no added cost. If you leave a small amount in the wallet that opens without the passphrase, that wallet alone can also serve as a decoy if you're ever threatened. The drawback is just as clear: since accessing funds requires both elements, a passphrase setup is similar to a 2-of-2 multisig made of two pieces — the mnemonic and the passphrase. If you store the two separately and the passphrase is strong enough, a leak of just one makes accessing the funds difficult. But lose either one, and you can't recover the same wallet at all. It's strong against theft, but it creates one more single point of failure on the loss side. In particular, if you keep the passphrase only in your memory and never record it, your memory itself becomes that point of failure. And as with the Coldcard incident in Chapter 1, if the device generated a vulnerable mnemonic in the first place, adding a short or common passphrase won't be enough to defend against it.

Multisig Setup

In multisig, the common setup is 2-of-3: three independent keys, two of which must sign to move funds. If one key is stolen, that key alone can't move the funds; if one key is lost, the remaining two can still access them. The advantage is that it defends against both theft and loss at once.

It matters that the three keys come from devices made by different manufacturers. If all three come from the same company, the moment a flaw turns up in that company's firmware, all three keys are put at risk together. In other words, a flaw in one manufacturer's device can't weaken keys generated on another manufacturer's device. Applied to the Coldcard incident in Chapter 1: in a 2-of-3 setup where the other two keys are secure, you could exclude the vulnerable key made on the Coldcard, create a new wallet with a new key added in its place, and move your existing funds over. The drawback is that setup and operation are more cumbersome, and beyond the keys themselves, you also need to back up the wallet's configuration information — the output descriptor.

Six Criteria for Choosing a Device

Once you've chosen a setup, the next step is choosing a device. Hardware wallet specs vary so much that manufacturer descriptions alone make comparison difficult. Based on what those earlier incidents actually exploited, this comes down to six criteria.

First, air-gapped operation. The signing device is used with no real-time data connection to a computer, moving Partially Signed Bitcoin Transactions (PSBT) via QR code or microSD card instead. Avoiding a direct connection reduces the attack surface, but it doesn't eliminate risk in the data-transfer path or in the signing device itself.

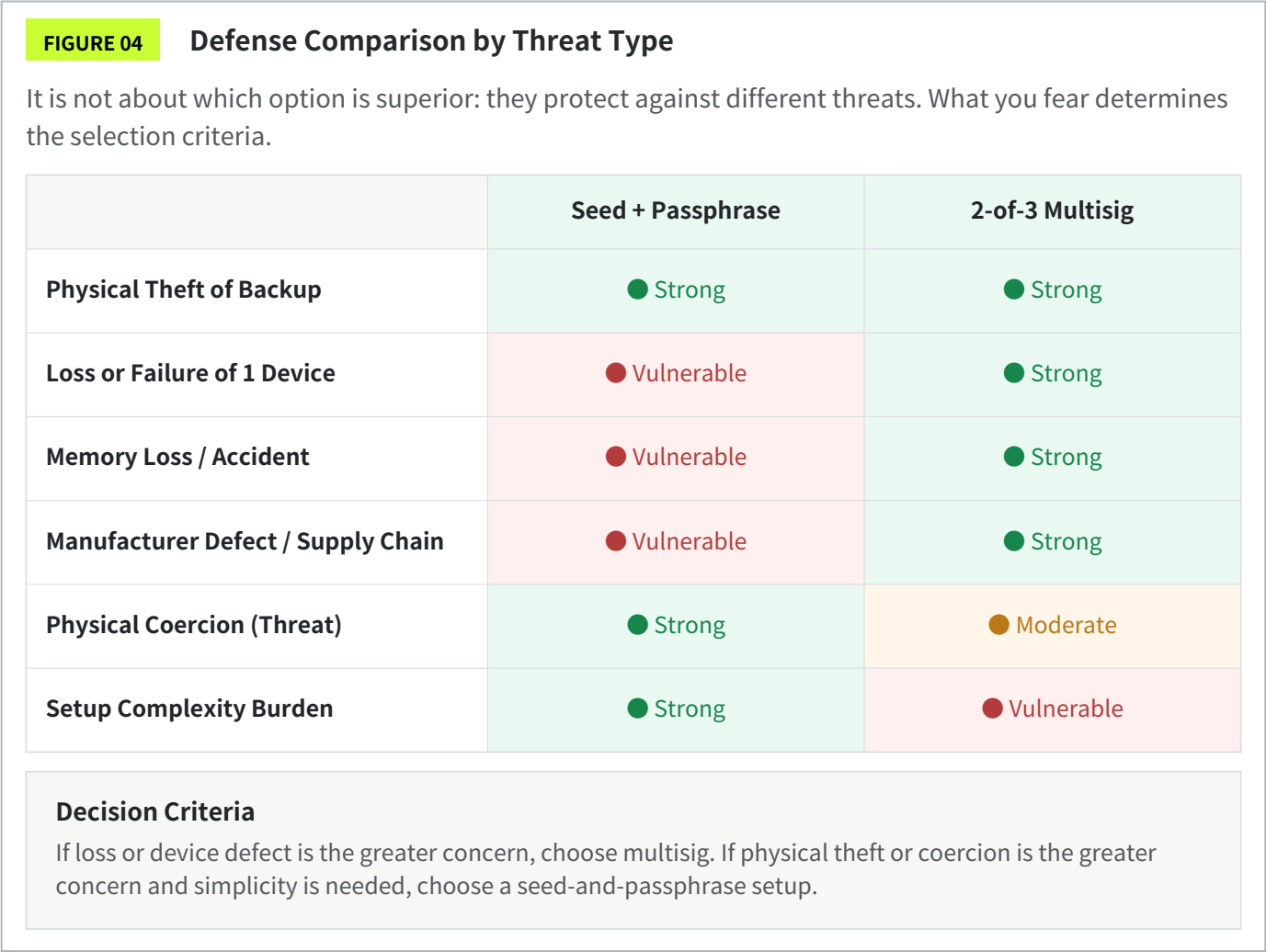
Second, defense against physical key extraction. This covers protecting the key if the device itself is stolen outright. There are two approaches: encrypting the key inside a dedicated security chip (a secure element) designed so the key can never be extracted, or not storing the key on the device at all, wiping it the moment power is cut. Either way, the goal is the same — even if someone gets hold of the device, they can't pull the key out.

Third, defense against supply-chain attacks. This looks at whether the packaging has tamper-evident seals, whether there’s a mechanism to detect that it’s been opened, and whether the device wipes its own key if it detects disassembly. This guards against the device being swapped out in transit.

Fourth, a nonce-generation procedure appropriate to the signature scheme, with verified implementation. ECDSA can use a deterministic method like RFC 6979; Taproot’s Schnorr signatures follow the BIP340 procedure.

Fifth, support for passphrases and multisig. For multiple signing devices to sign in turn while air-gapped, the device needs to support PSBT.

Sixth, whether the user can add their own randomness. If you can feed values generated with dice or coin flips into the seed, your own randomness survives even if the device’s random number generator is broken. The Coldcard incident in Chapter 1 wouldn’t have been caught by the first five criteria — this one closes that gap. In fact, Coldcard’s manufacturer stated that seeds into which entropy had been manually added via 50 or more dice rolls were not affected by this defect.



In the public feature-comparison database this report drew on, six models from four manufacturers met all six criteria: the Coldcard Mk4, Mk5, and Q, which each combine two security chips from different vendors; the Keystone 3 Pro, with three security chips and self-reported PCI-level anti-tamper features; Cardware, which uses an EAL6+-rated security chip; and the SeedSigner, which instead of a security chip, chose not to store the key at all. Since the SeedSigner erases the seed when powered off, there’s no stored key to steal, and

because it's assembled from parts you buy and put together yourself, it's cheaper and carries less risk of being swapped out in transit.²² That said, this is a shortlist based on feature criteria, not a certification of overall product safety. An actual purchase decision should also factor in official documentation, independent security reviews, disclosed vulnerabilities and how they were handled, firmware-verification methods, and supply-chain and recovery risk together.

Coldcard, at the top of this list, is the product from the manufacturer behind the Chapter 1 incident. It stays on the list for two reasons: the problem originated in the firmware's seed-generation code rather than a breach of the device itself, and the manufacturer released patched firmware for all affected models the day after the vulnerability was disclosed. That said, updating the firmware doesn't make an already-vulnerable seed safe — the manufacturer recommended generating a new seed and migrating funds. The public disclosure didn't mention any compensation or exchange program.²³

If you're building a multisig, the key is to mix devices from different manufacturers. If your 2-of-3 mixes devices as different as Coldcard, Keystone, and SeedSigner, a hole in one company's firmware won't stop you from moving funds with the remaining two keys. Applied to the Coldcard incident in Chapter 1: if only one of the three keys had been a Coldcard, a breach of that key wouldn't have breached the security of the other two, and the hacker would never have gained the authority needed to move the funds.

FIGURE 05 Hardware Wallets Meeting All Six Criteria

Out of 74 devices in the public database, six models across four manufacturers met all six criteria.



Coldcard Mk4 • Mk5 • Q

Combines two security chips from different vendors.
Supports dice input.



Keystone 3 Pro

Houses three security chips with payment-terminal-grade (PCI) certification.
Open-source.



Cardware

Uses an EAL6+-rated security chip. Released in 2025, so the source is not public yet.



SeedSigner

Uses no security chip and stores no key. The seed is wiped when power is cut.

FIGURE 05 Hardware Wallets Meeting All Six Criteria (continued)

How to Choose

If you are building a multisig, mix devices from different manufacturers on this list. If all the devices come from one company, a defect in its firmware could put every key at risk at once.

Generation Phase: The Environment Where You Make the Key

Before generating a key, confirm you bought the device from an official retailer, and check the packaging and the device itself for signs of tampering. Reject any device that comes pre-initialized or with a pre-written seed. Download the latest firmware through the official channel, verify its signature or hash, then log the model and firmware version.

Generate the key on a hardware wallet that has never been connected to the internet. If a seed is created on a general-purpose computer or a smartphone app, and that device was already infected, you should treat the seed as compromised from the moment it was created. Section 2.1's Randstorm case shows how especially risky it is to generate a seed inside a browser.

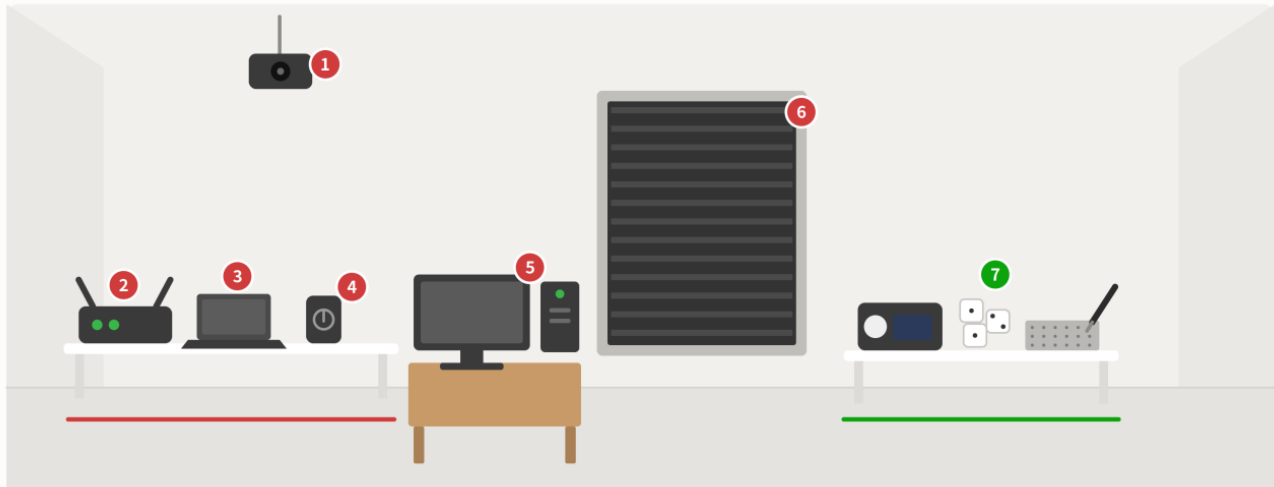
Scaled down from the institutional procedures in Section 3.1 to the individual level, this comes down to: while you're generating the key and writing down the seed, turn off every nearby device that could transmit a signal or record sound or video.

One more thing: if the device supports it, use the feature that lets you enter randomness from dice directly. Roll the dice enough times and feed the results into the seed, and your own randomness survives even if the device's random number generator is broken. In the Coldcard incident, people who generated their seed this way were unaffected by the hacking risk.

Twelve words is enough for the mnemonic. A 12-word seed is 128 bits of random value plus a 4-bit checksum, making 132 bits, cut into 11-bit chunks; a 24-word seed does the same with 256 bits plus 8 bits. Because the actual strength of the cryptography used in Bitcoin signatures is about 128 bits, using 24 words gives essentially the same security level as 12. Fewer words also means less room for error when writing it down, checking it, and recovering it — that's the real, practical advantage of 12 words.

FIGURE 06 The Room Where You Make the Key: What to Turn Off, What to Keep

Scaled down from why institutions use electromagnetic shielding tents, at the individual level this means removing every device that transmits a signal or records something.



No.	Item	Action
1	Indoor CCTV / Home Cameras	Cut power even to doorbell cameras.
2	Router / Wi-Fi	Power it off or disconnect the LAN cable.
3	Laptop / Tablet / Webcam	Move them to another room entirely.
4	Smartphone / Smartwatch	Power off — not airplane mode.
5	Desktop / Monitor / Smart Speaker / TV	Turn off anything with a microphone.
6	Windows and Glass Doors	Lower the blinds all the way. If they're not covered, you can be filmed from outside.
7	What to Keep	Hardware wallet, dice, metal plate, and engraving tool.

Everything in red gets turned off or removed. Row 7, in green, is what you keep.

Generate the seed and write it down by hand only in this room. Don't photograph it, don't type it into a computer, and destroy any temporary notes right after you're done.

Generating a Seed With Dice on a SeedSigner and Registering It in Sparrow

Here's how the principles above actually play out, using a SeedSigner and Sparrow Wallet. SeedSigner was chosen for three reasons: it supports dice input, it doesn't store the key, so you can practice as many times as you like, and it operates using only a camera and QR codes, so you can visually confirm for yourself that it's disconnected from the internet.

First, set up the environment shown in Figure 6 and turn on the device. From the “Tools” menu, choose “New seed,” marked with a dice icon, and the dice-entry screen appears. Roll 50 times for a 12-word seed, or 99 times for a 24-word seed, and enter the results exactly as rolled. Use dice that roll evenly, and don’t photograph or record the screen while entering them.

FIGURE 07 Generating a Seed With Dice on a SeedSigner

Instead of trusting the device's random number generator, the user adds randomness themselves. In the Coldcard incident, those who used this method didn't need to switch wallets.



Step 1: Tools → New seed

Choose the option marked with a dice icon. The camera option is a photo-based method.



Step 2: Enter the Dice Roll Results

Roll 50 times for a 12-word seed, or 99 times for a 24-word seed, and enter the results exactly as rolled.



Step 3: Record the Generated Words

Transfer the words shown on the screen onto a metal plate. Don’t take photos.



Step 4: Select Export Xpub

The fingerprint shown at the top becomes your reference for cross-checking later.

Important

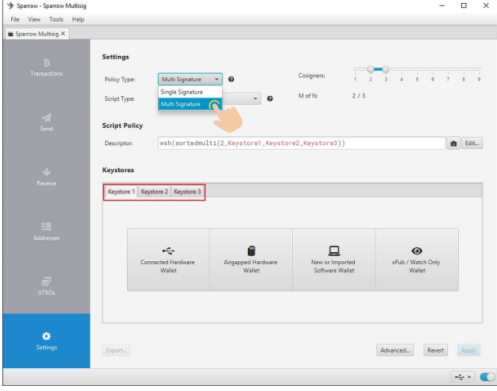
The screen showing the dice-roll entry must never be photographed or recorded. The values in this figure are just an example and have nothing to do with an actual seed.

Once entry is done, the words appear on the screen. Engrave these words onto a metal plate, and check them once more against the device screen. From here on, only the public key and signed transactions are sent to the computer — never the raw private key. Since SeedSigner wipes the seed when powered off, you’ll need to re-enter the backed-up mnemonic the next time you use it.

Next is registering it in your wallet software. On the device, choose “Export Xpub,” and the extended public key (a read-only key that can generate that wallet’s addresses without the private key) appears as a QR code.

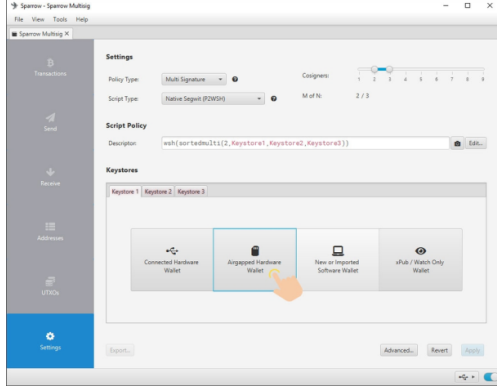
FIGURE 08

The private key never leaves the device. Only the extended public key and derivation path pass through the QR code.



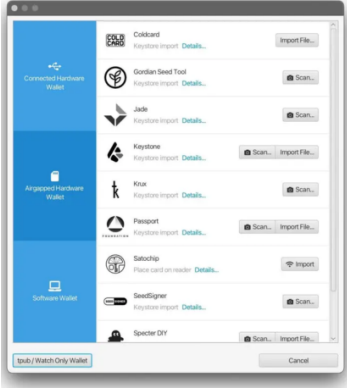
Step 1: Set Policy Type to Multi Signature

Setting Cosigners to 3 makes M of N 2/3, and creates three Keystore tabs.



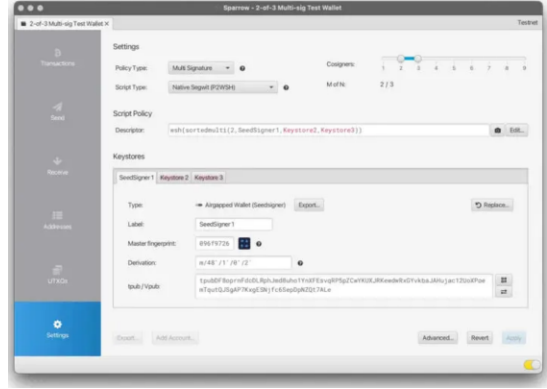
Step 2: Choose the airgapped wallet option under Keystore 1

The script type becomes Native Segwit (P2WSH), and the descriptor becomes `wsh(sortedmulti(2,...))`.



Step 3: Click Scan for the device you have chosen

Choose a different manufacturer for each device and repeat three times. Don't fill all three with the same manufacturer.



Step 4: Cross-check the fingerprint and derivation path

The multisig path begins with m/48'. Confirm that the value on the screen matches the value on the device.

Check

The Step 4 screen is from a testnet wallet, so it shows tpub and coin type 1. A wallet with real funds will show xpub and coin type 0.

At this point, check the derivation path (the specific path followed when generating addresses from a single seed). On the Bitcoin mainnet, a native SegWit single-signature wallet uses `m/84'/0'/0'`, and a native SegWit

multisig wallet uses `m/48'/0'/0'/2'` — choose the path that matches your setup.²⁴ On the computer, open Sparrow, go to “Keystore” → “Airgapped Hardware Wallet” → “SeedSigner” → “Scan,” and read this QR code.

Once registration is done, make sure to check that the master fingerprint (an eight-character ID used to identify the key) and the derivation path shown in Sparrow match what’s on the device screen. If they don’t match, you’ve registered the wrong device or account — stop immediately. This comparison is the starting point for the address verification covered in Section 5.5.

To build a 2-of-3 multisig, change Sparrow’s “Policy Type” to “Multi Signature” and set it to 2 of 3, then repeat the process above on three devices from different manufacturers. Screens for the other manufacturers are omitted here to save space, but the key is to choose devices — from among those meeting the six criteria in the previous section — that come from different companies. Once all three keys are registered, a descriptor is generated containing the wallet’s configuration; this must also be backed up, separately from the keys. Having just the three keys without the descriptor can make recovery difficult. Since the script type, derivation path, and signer order all have to match exactly to reproduce the original addresses, trying every combination by hand amounts to a separate recovery effort in itself.

Backup Phase: How to Record the Mnemonic

Paper is weak against fire and water, so metal is the default choice for recording it. But not all metal is the same. Among publicly available durability tests, the one that has been repeatedly verified is the stress test Jameson Lopp ran on more than 75 products. He applied roughly 1,090°C flame for 10 minutes, then submerged the devices in water, corroded them in hydrochloric acid, and pressed them with a 20-ton press. The test temperature was set at roughly twice the average of a typical house fire, about 590°C.

What determined the results was material and structure. Aluminum products mostly melted, since their melting point falls short of the fire temperature used. Modular designs — where letter tiles slot into rails — lost their record when heat warped the rails and the tiles fell out. A single stainless steel plate, by contrast, with dots punched or letters stamped directly into it, withstood the fire, the acid, and the press. This confirms a simple principle: fewer parts means fewer places to fail.²⁵

304 stainless steel or higher is enough. Titanium is also good, but in realistic disaster-preparedness terms, it showed no meaningful difference from stainless. A single plate engraved directly is recommended. As seen above, the modular tile designs were weak to heat, and plates that reveal letters through a black coating didn’t last 30 seconds in flame. Don’t use engraving-outsourcing services — doing so exposes the seed at three separate points: the company’s staff, the order records, and the delivery route. For reference, BIP-39 words are designed so each one is distinguishable by its first four letters alone, so if space is tight, engraving just the first four letters of each word is fine.

If you’re using a passphrase, the mnemonic isn’t the only thing that needs backing up. If either the mnemonic or the passphrase is lost, you can’t recover the same wallet. The memory in your head is not a backup. The passphrase should be recorded on metal exactly the same way as the mnemonic.

FIGURE 09 Durability Test Results by Metal Backup Method

Tests included exposure to approximately 1,090°C flames, immersion in hydrochloric acid, and 20-ton compression. The results varied depending on the method.

Center Punch (Dot Stamping)	● Excellent	No loss in fire, acid, or compression tests
Electrochemical Etching	● Excellent	Legibility actually improved after acid exposure
Stamping (Letter Engraving)	● Excellent	Discoloration and warping, but still legible
Coating-Dependent Marking	● Caution	Black coating didn't withstand 30 seconds of fire
Tile / Letterboard Assembly	● Unsuitable	Rails warped from heat, causing tiles to fall out
Aluminum, Any Method	● Unsuitable	Low melting point — melts in a fire

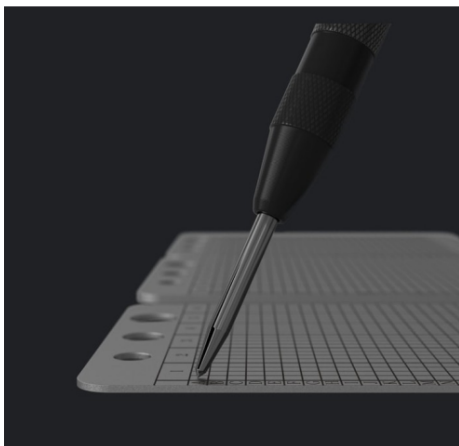
Recommendation: Directly stamping the seed onto a single stainless-steel plate with fewer components was the simplest method and produced consistently good test results.

Engraving services should not be used because they expose the seed to a third party.

Source: Jameson Lopp, Metal Backup Stress Test (more than 75 products tested)

FIGURE 10 Two Methods for Recording a Metal Backup

These are the two methods that passed all three tests: fire, acid, and compression. With either method, you engrave the seed yourself.



Center Punch (Dot Stamping)

Use an automatic center punch to mark dots at the positions corresponding to each word's number on the grid. This method uses the fewest parts.



Stamping (Letter Engraving)

Use letter punches and a hammer to stamp the words into the plate. The words remain legible even after discoloration from heat.

Warning

Don't use engraving-outsourcing services — doing so exposes the seed at three separate points: the company's staff, the order records, and the delivery route.

Shamir Backup and Seed XOR: How Individuals Can Imitate Institutional Key-Splitting

Individuals can also imitate the way institutions split keys into multiple pieces. A well-known example is SLIP-39 Shamir Backup. It breaks the seed into multiple pieces that reconstruct the original key only when a specified threshold is met—such as requiring two out of three pieces. If one piece leaks, no information leaks with it; if one is lost, recovery still works. Some hardware wallets use this as their default backup method.

But it has a limit. When splitting or recombining the seed, the whole seed comes together on a single device at that moment. If that device is infected, the benefit of splitting it into pieces may be lost. Multisig, by contrast, lets each key be generated and stored on a different device, so there's never a need to bring multiple private keys together on one device. Each SLIP-39 share includes an RS1024 checksum that catches input errors, able to detect mistakes in up to three words during recording or entry.²⁶ But hardware and software that support this remain limited, and the generation, recovery, and inheritance procedures are also complex, so you should confirm in advance that your chosen device and software are compatible.

Coldcard's Seed XOR splits the mnemonic into multiple pieces using bit-level operations. What's unusual is that each piece looks, on its own, like a perfectly normal wallet. If you're threatened and hand over one piece, the other party has a hard time telling whether it's a real wallet or just one share of a split one. But since every piece is needed to restore it, losing even one means there's no way to recover it — in other words, it doesn't lower your risk of loss at all.²⁷

Don't Invent Your Own Splitting Method

Devising your own method — like cutting the 24 words in half and storing the two halves separately — is off-limits. A homemade method like this has never been vetted, and if even part of the words leak, the number of combinations an attacker has to try shrinks along with it. If you need to split your seed, use a verified method like SLIP-39 or Seed XOR.

Storage Phase: Where to Keep Things

There are two storage principles. First, never keep the device holding a seed and its backup in the same place. Second, split backups or multisig keys across multiple locations, physically separated from each other. Good candidates include a home safe, a bank safety deposit box in a different area, or a trusted family member's place. If you're using a passphrase, its record follows the same principle as the mnemonic — but keep it in a different location from the mnemonic. Keep both in one place, and if that place is compromised, both go together, erasing the whole point of adding a passphrase.

There are things to know before using a bank safety deposit box. The contents aren't covered by deposit insurance, and the bank doesn't carry separate insurance for them either. If rent falls behind, the bank may open the box and move the contents. For that reason, putting a single-seed setup's entire backup into a safety deposit box isn't recommended. It's safer to store something that can't move money on its own — a single multisig key, or a single Shamir share. Also off-limits: putting enough keys in one bank to meet your multisig threshold.²⁸

If you're entrusting a backup to someone else's space — a safety deposit box or a family member's — here's what's recommended: put it in a tamper-evident bag, sign it and date it, and log the serial number separately. Don't tell staff what you're storing or why. Check the physical item and the seal every 6 to 12

months. If you’re thinking about inheritance, check in advance with your bank whether access is suspended on death, what the box-opening and return process looks like for an heir, and what documents are required — then build that into your emergency access plan.

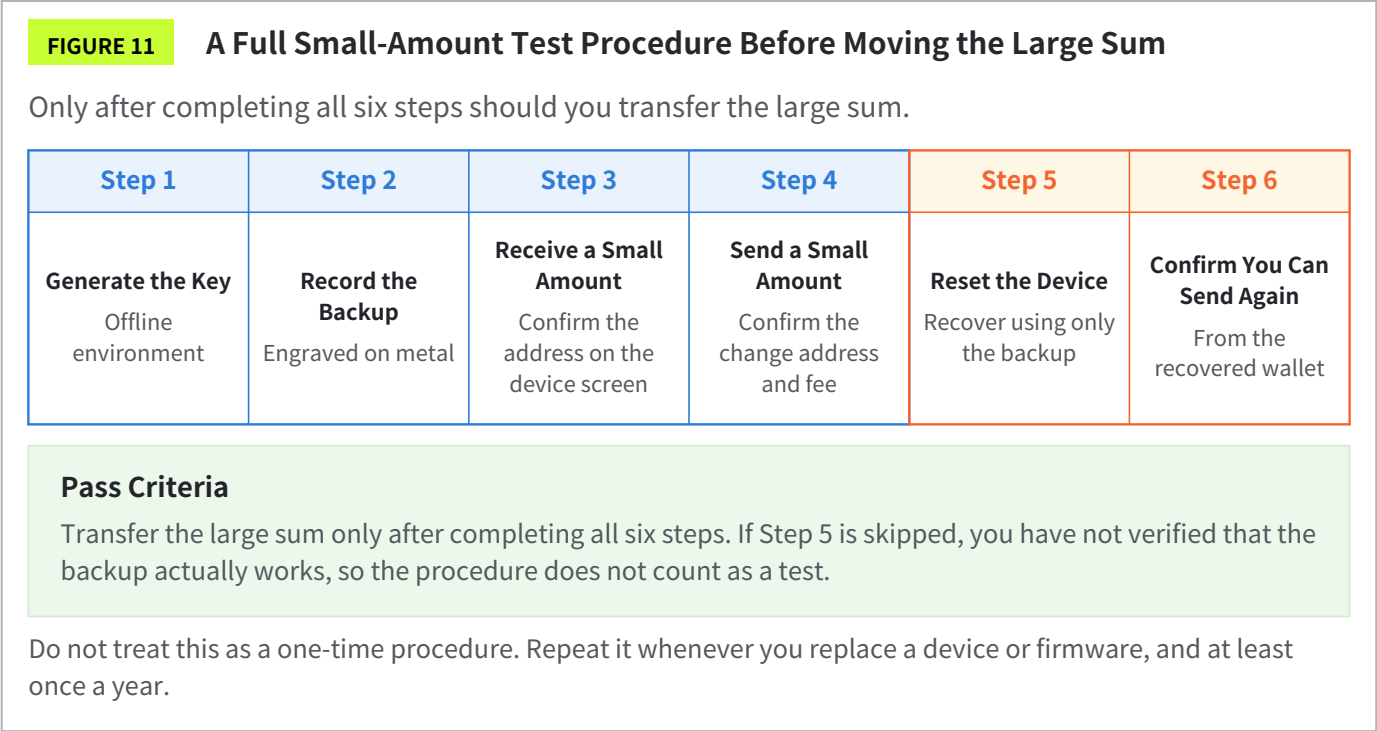
■ Signing Phase: What to Verify

When receiving funds, verify the receiving address directly on the hardware wallet’s screen. An address shown on a computer or phone screen can be swapped out by malware. When sending, verify the destination address, amount, and fee on the hardware wallet’s screen. Since devices differ in how they display and verify change, also check for any unexpected transfer details or change-related warnings.

If you want an extra layer of verification, recalculate the address the device shows using an independent tool. Using Bitcoin Core’s address-derivation feature or a hardware-wallet integration tool, you can regenerate the address from the extended public key. If the two values match, you’ve separately confirmed the address really belongs to your wallet.²⁹ If you built a multisig with devices from different manufacturers, you can cross-check the receiving address across devices, so even if one is infected, you won’t be fooled by a fake address.

■ Recovery Phase: Test the Full Procedure With a Small Amount

This is the most important section in this report. Don’t send a large sum straight to a newly created wallet. First, run the entire wallet process from start to finish with a small amount, confirm everything works, and only then move the large sum over. A backup isn’t a backup until you’ve actually recovered from it, and a wallet isn’t ready to use until you’ve actually sent a transaction from it.



In particular, don’t skip step 5. Checking things while the original device is still around doesn’t tell you whether the backup actually works. It only means something if you reset the device, or use a different one,

and confirm the wallet comes back to life using nothing but the stored backup. Problems like misrecorded seed words, a misremembered passphrase, missing multisig configuration, or a mismatched derivation path almost always show up here. If you find them after putting in the large sum, it's difficult to fix; if you find them at the small-amount stage, you can fix it right away.

Don't treat this test as a one-time thing. Repeat it whenever you change devices or firmware, and at least once a year, to confirm the backup still works.

Using AI as a Verification Aid

FIGURE 12

Where AI Can and Can't Be Used in Self-Custody

There's one rule: AI never sees data related to your wallet. Use it only for public information and conceptual questions.

● Suitable: Tasks With None of Your Wallet Data

- Explaining unfamiliar concepts and terms
- Reviewing published, open-source firmware code
- Walking through manufacturer documentation and vulnerability notices
- Reviewing a procedure or threat-modeling it, with identifiers stripped out
- Writing a verification tool you'll run yourself, offline
- Checking a recovery-test sequence for missing steps

● Forbidden: Tasks Involving Your Wallet Data

- Seed / mnemonic / private key / passphrase
- Extended public key (xpub) and wallet descriptor
- Raw or decoded transaction data (PSBT), before or after signing
- Your address list, holdings, or backup storage location
- Asking AI for a firmware download link
- Delegating crypto operations like address derivation or key calculation

Warning

Never enter your seed, private key, or passphrase — not even partially redacted, not even replaced with an example. The same caution applies to extended public keys and transaction data.

The AI services most people use are external, connected to the internet, and there's no guarantee that what you enter ever disappears.

Before using AI for verification, get the premise straight: the AI services most people use run on infrastructure you don't control, sitting across the internet, and once you send something, it's out of your hands. There's no guarantee your chat history gets deleted. In 2025, shared conversations turned up exposed in search engines, and in at least one case, a company was ordered to preserve user-deleted conversations for litigation.³⁰ Even so, large language models have gotten noticeably better at programming lately, and top-tier models are genuinely useful at several stages of self-custody. As long as you never put in your own wallet data — only public information, conceptual questions, or a plan with the sensitive values stripped out — AI can be a solid verification partner.

Approach 1: Only work with information that's already public.

Questions like what an unfamiliar term means, what a derivation path notation represents, or how reproducible builds work (rebuilding the same firmware from published source code and checking it against the official release) are general knowledge unrelated to your own wallet, so ask freely. You can also read your

wallet's public firmware code together with an AI to look for vulnerabilities, or have it walk you through manufacturer documentation or vulnerability notices. In fact, after the Coldcard incident, AI-assisted analysis reportedly turned up additional problems in several other wallets.³¹

Approach 2: Instead of handing over data, have it build you a tool.

This is the most substantive approach. Rather than sending the data you want to check to an AI, ask it to write a small script that checks that data, then run the script yourself on an offline device. Since the AI only produces code and never sees your data, your wallet information never leaves your device. Here are a few things this approach can do — but never put a mnemonic, private key, passphrase, or extended private key that's actually in use into an AI-generated script; run these only with practice values unrelated to your real wallet.

- A script that converts dice-roll randomness into a mnemonic, for practice
- A script that checks whether a draft transaction's (PSBT) recipient address, amount, and fee match what you intended
- A script that derives addresses from an extended public key and checks them against what's shown on the hardware wallet screen
- A script that calculates the hash of downloaded firmware, compares it to the value the manufacturer published, and reports only the result
- A script that checks, offline, whether a practice mnemonic's checksum is valid
- A script that reads a multisig descriptor containing only public keys and lays out the co-signer setup and derivation path in a human-readable form

There are three things to watch for. First, always read and understand the script before running it — specifically check for anything sending data out, such as network requests, file uploads, or calls to outside services; that's your leak point. If you're not sure what to look for, ask the same AI to point out every part of the code that touches the network. Second, run it on a device with no network connection. Third, before installing any external library the script asks for, confirm it actually exists.

Approach 3: Have it review your design and procedure with identifying details stripped out.

You can also describe your setup without specific identifying information and ask for a review. For example: you built a 2-of-3 using three different manufacturers, stored the keys in three separate places, and backed up the descriptor somewhere — just the skeleton, no addresses or extended public keys. That way you can safely ask what order of failures could break in, or whether any step is missing.

How you phrase the question matters too. Don't ask "is my setup safe" — ask "what attacks are possible against this setup." The first question makes it easy for the AI to give you a reassuring answer. Walking it through your Section 5.6 recovery-test sequence and asking it to flag any missing step is another good use of this approach.

If you absolutely have to analyze real wallet data

If none of the three approaches above solve your problem and you genuinely need to analyze actual wallet data, one option is running an open-weight, open-source model yourself, locally, on a computer disconnected from the internet. Since nothing leaves the device, the nature of the risk is different.

That still isn't a fully safe option, though. The device running the model may keep a record of what you entered, and the moment that device later connects to the internet, that record becomes exposed. If you go this route, don't use that device for anything else, and once verification is done, wipe the logs or reset the device. Keep it separate from any device you use for signing, too. And keep in mind: the weaker the model, the more confidently it tends to produce wrong analysis.

Information you must never enter

There are two categories of information you should never put in. One is information that can cost you money the moment you enter it. The other doesn't drain funds immediately, but once it's out, it can't be taken back.

Information that can cost you your assets immediately

This includes the mnemonic (seed phrase), private keys, passphrases, and extended private keys. A wallet descriptor that includes a private key or extended private key counts as the same kind of secret. Partially redacting this information doesn't make it safe. In particular, don't enter the mnemonic even with just a few words swapped out. For a 12-word mnemonic, if the word positions are known and nine words are exposed, the remaining three words narrow down to a range small enough for a computer to brute-force. Uploading a photo of the seed or a backup file is just as dangerous.

Information that causes irreversible loss of privacy

This includes extended public keys, wallet descriptors that contain only public keys, the raw text and decoded contents of an unsigned transaction, address lists, holdings, and backup storage locations. None of this lets someone move your assets directly — but depending on what's exposed, your balance, transaction history, identity, or storage location can come to light, making you a target for physical threats. Once your transaction history is linked to your identity, that link is hard to undo even by switching addresses.

FIGURE 13 Example Prompts for Using AI as a Verification Aid

One rule: never put data about your own wallet into the AI. Only ask for public information, conceptual questions, or a tool you'll run yourself.

Category	Example Prompt	Why
Suitable • Concept Check	“Explain what each number means in the Bitcoin derivation path <code>m/48'/0'/0'/2'</code> . Also tell me why this path is used for multisig rather than single-signature.”	A general-knowledge question with none of your own wallet's values in it. You're asking about a public standard, so there's nothing to leak.
Suitable • Writing a Verification Tool (most practical)	“Write a Python script that takes a PSBT file and checks: 1) that all outputs' BIP32 derivation paths are consistent with each other; 2) that the output marked as change actually belongs to my account's path; 3) that the difference between input and output totals matches the intended fee. Conditions: don't include any code that connects to a network. Use only the standard library and widely audited open-source packages — I'll run this myself on an offline device.”	You never enter the PSBT itself — only ask for a tool that reads it. The data never leaves your device.
Suitable • Cross-Checking Received Code	“In the script you just gave me, flag everything that sends data out. Point out each relevant line — network requests, file uploads, external API calls, remote logging.”	A mandatory step before running anything. Read the code yourself, but also cross-check with the same tool for anything you might have missed.
Suitable • Reviewing a Setup and Procedure	“Tell me what attacks are possible against this setup: a 2-of-3 multisig using airgapped devices from three different manufacturers, keys split across a home, a bank safety deposit box, and a family member's house, with descriptor copies kept in two of the three locations. I won't tell you the addresses or extended public keys.”	It asks what attacks are possible instead of whether it's safe. The latter question makes it easy to get an agreeable answer.
Forbidden • Entering Wallet Data Directly	“Analyze my PSBT for anything unusual. <code>cHNidP8BAHECAAAAAfuJ2K1cQ9pR...</code> (truncated)” “Give me a final verdict on whether my wallet is safe. My extended public key is <code>xpub6BfhisZn1pCf8WkEo4k9Y5...</code> ”	A PSBT contains the extended public key, master fingerprint, and derivation path together. Once shared, your whole balance and transaction history are exposed, and changing addresses won't undo it. Pasting the output of a decode command is just as bad.

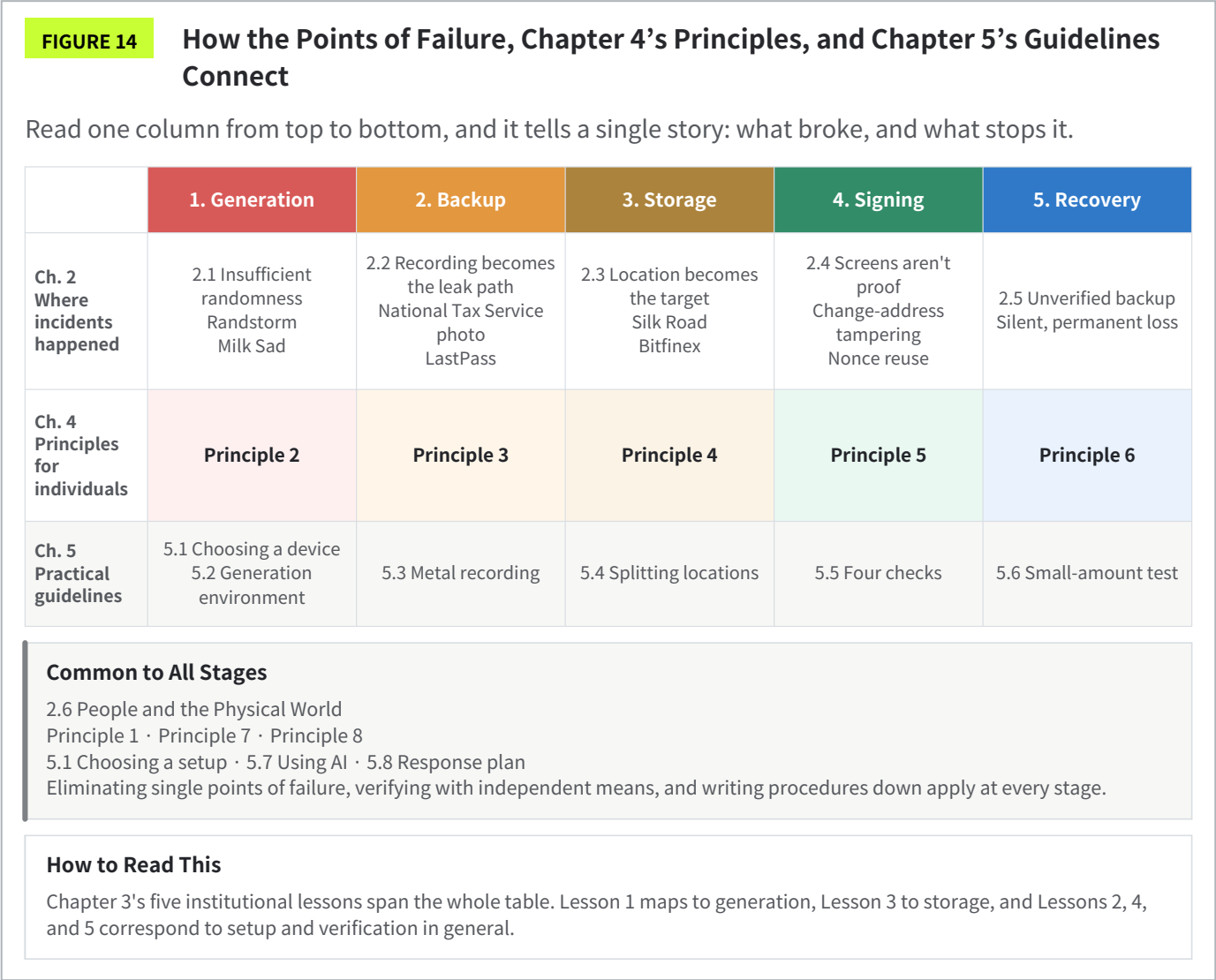
A Response Plan for Leaks and Loss

Decide ahead of time, in writing, how you'll act if something goes wrong. If you lose one hardware wallet, or it's stolen, a passphrase setup means moving all your funds to a new wallet right away; a multisig means moving assets into a new setup that replaces just the compromised key. If there's even a small chance the seed or passphrase has leaked, treat that wallet as already exposed and act on that basis.

Decide in advance how you'll respond if a manufacturer discloses a vulnerability, too. As the Chapter 1 incident showed, updating the firmware doesn't fix a seed that was already generated. If your seed was made during a vulnerable period, there's no answer except generating a new seed and moving your funds. Ahead of time, prepare a safe evacuation wallet and finish the small-amount full-procedure test from Section 5.6 on it. If an attack is underway or you suspect your seed has leaked, confirm the address of that already-verified evacuation wallet and move your assets quickly. In an emergency, don't leave assets sitting in the compromised wallet just to finish a full recovery test on it.

Finally, build inheritance and emergency access into your plan too. If someone you trust has no way to reach your assets after you can no longer access them yourself, that setup is, sooner or later, working toward permanently losing those assets. The responsibility of self-custody doesn't end with your lifetime.

That covers the guidelines in Chapter 5. Below, in one table, is where each incident from Chapter 2 happened, which principle in Chapter 4 addresses it, and where the practical method appears in Chapter 5.



Conclusion

If you can hand this off to a trustworthy exchange with solid internal controls and real regulation behind it, and that's enough for you, you don't need this report. But as the two incidents in Chapter 1 showed, you can lose your assets through no fault of your own — a single choice about how you record something can be enough to lose everything. Self-custody means you carry all of that risk yourself.

Even so, some people have no choice but to take this path. What this report offers isn't a way to eliminate all risk — it's a set of guidelines for closing off known failure points, one by one. As Chapter 2 showed, none of the incidents so far happened because Bitcoin's cryptography was broken. They happened because randomness was insufficient, a record leaked, a key ended up somewhere it shouldn't have, or a check got skipped. Put another way: the territory an individual actually has to manage is wider than most people assume.

One thing worth emphasizing again at the end: the small-amount full-procedure test. Whatever setup you choose, when you're setting up a new wallet, don't move the large sum in right away. First, run the entire process with a small amount — generation, backup, receiving, sending, and recovering using only the backup — and confirm you can send again from the recovered wallet. Only then move the large sum over. It takes less than a day and costs almost nothing, but it surfaces problems that would be hard to do anything about if found later, before you've put real money at stake.

If you're going ahead with self-custody, it's better to treat it as something you keep maintaining, not something you set up once and finish. Devices change, new vulnerabilities turn up, and backups become impossible to verify just by the passage of time. Put a yearly check and recovery drill on your calendar, and keep refining your own manual as you go. That's a more reliable defense than any expensive piece of hardware.

Notice

The figures in this report are based on news coverage, on-chain analysis, and manufacturer disclosures, and may change over time. This report is for informational purposes only and does not constitute investment, legal, or tax advice.

Notes

1. TRM Labs, “The Largest Hardware Wallet Exploit of 2026: Inside the USD 116 Million Coldcard Hack,” accessed August 18, 2026, <https://www.trmlabs.com/resources/blog/the-largest-hardware-wallet-exploit-of-2026-inside-the-usd-116-million-coldcard-hack>. Total damage estimates vary depending on the reporting entity and timeframe. The Hacker News reported approximately 1,083 BTC (~\$70.2 million) on August 1, 2026, which was updated in a later edit to approximately 1,367 BTC (~\$88.6 million) (<https://thehackernews.com/2026/08/coldcard-hardware-wallet-flaw-linked-to.html>). On August 4, 2026, TechCrunch reported that losses exceeded \$130 million, citing figures from Galaxy Research (<https://techcrunch.com/2026/08/04/hackers-steal-over-130-million-by-exploiting-bug-in-offline-hardware-wallets/>). The USD conversion in the main text is based on the spot price on the day of the incident (July 30, 2026: ~\$64,500 per BTC).
2. Block Engineering, “Predictable RNG Fallback and 32-Bit Reseed in COLDCARD Firmware,” accessed August 18, 2026, <https://engineering.block.xyz/blog/predictable-rng-fallback-and-32-bit-reseed-in-coldcard-firmware>; Coinkite, “Coldcard Security Advisory,” accessed August 18, 2026, <https://blog.coinkite.com/coldcard-mk3-seed-generation-warning/>. According to the manufacturer advisory, affected firmware versions include 4.0.1 (March 2021) to 4.1.9 for Mk2 and Mk3, prior to 5.6.0 for Mk4 and Mk5, and prior to 1.5.0Q for Q. The Block report states the starting affected version as 4.0.0. The manufacturer clarified that seeds generated using manual entropy injection via at least 50 fair dice rolls were unaffected by this vulnerability. On July 31, 2026, patched firmware was released for all affected models, alongside a recommendation to generate new seeds and migrate assets for wallets created during the vulnerable window.
3. Ji-hyun Ham, “Is AI Amplifying Security Threats?... Coldcard Incident Shakes ‘Self-Custody’ Myth (Comprehensive),” BlockMedia, August 3, 2026, accessed August 18, 2026, <https://www.blockmedia.co.kr/archives/1123428>.
4. The PR, “Aftermath of NTS Press Release: Seizure News Leaks Crypto,” accessed August 18, 2026, <https://www.the-pr.co.kr/news/articleView.html?idxno=60674>. PRTG (Pre-Retogeum) is an Ethereum-based token. The estimate of approximately 6.9 billion KRW corresponds to roughly 1,725 KRW per token based on a 4-million-token balance.
5. “Stolen Right After Recovery: NTS Seized Crypto Suffers Second Exploit,” Seoul Shinmun, March 2, 2026, accessed August 18, 2026, <https://www.seoul.co.kr/news/society/accident/2026/03/02/20260302500147>; “NTS Returns 4M Stolen Tokens After Recovery... Valuation Standards Remain a Challenge,” E-Daily, May 6, 2026, accessed August 18, 2026, <https://marketin.edaily.co.kr/News/ReadE?newsId=06123766645446296>.
6. Babylon Labs, “Bitcoin Staking,” accessed August 18, 2026, <https://babylonlabs.io/>.
7. Unciphered, “Randstorm: You Can’t Patch a House of Cards,” accessed August 18, 2026, <https://www.unciphered.com/disclosure-of-vulnerable-bitcoin-wallet-library-2/>.
8. Milk Sad Research Team, “Milk Sad: /bx Full Write-up” (CVE-2023-39910), accessed August 18, 2026, <https://milksad.info/disclosure.html>.
9. Brian Krebs, “Feds Link \$150M Cyberheist to 2022 LastPass Hacks,” Krebs on Security, March 2025, accessed August 18, 2026, <https://krebsonsecurity.com/2025/03/feds-link-150m-cyberheist-to-2022-lastpass-hacks/>.
10. Kaspersky, “SparkCat Trojan Stealer Infiltrates App Store and Google Play, Steals Data from Photos,” accessed August 18, 2026, <https://www.kaspersky.com/blog/ios-android-ocr-stealer-sparkcat/52980/>.

11. Samuel Oakford, “Prosecutors in Silk Road Trial Present Damning Evidence From Ross Ulbricht’s Computer,” Vice, January 21, 2015, accessed August 18, 2026, <https://www.vice.com/en/article/prosecutors-in-silk-road-trial-present-damning-evidence-from-ross-ulbrichts-computer/>; “Inside the FBI Takedown of the Mastermind Behind Website Offering Drugs, Guns and Murders for Hire,” CBS News, November 10, 2020, accessed August 18, 2026, <https://www.cbsnews.com/news/ross-ulbricht-dread-pirate-roberts-silk-road-fbi/>.
12. Namcios, “How Law Enforcement Seized 94,000 Bitcoin Stolen From Bitfinex,” Bitcoin Magazine, March 8, 2022, accessed August 18, 2026, <https://bitcoinmagazine.com/technical/how-authorities-found-bitfinex-bitcoin>.
13. Bitcoin Project, “Android Security Vulnerability,” August 11, 2013, accessed August 18, 2026, <https://bitcoin.org/en/alert/2013-08-11-android>; RFC 6979, “Deterministic Usage of the Digital Signature Algorithm,” accessed August 18, 2026, <https://www.rfc-editor.org/rfc/rfc6979>; BIP340, “Schnorr Signatures for secp256k1,” accessed August 18, 2026, <https://bips.dev/340/>.
14. Coinkite, “Troublesome Change Outputs,” accessed August 18, 2026, <http://blog.coinkite.com/troublesome-change/>.
15. Pascal Gauthier, “Message by Ledger’s CEO: Update on the July Data Breach,” Ledger, December 21, 2020, accessed August 18, 2026, <https://www.ledger.com/message-ledgers-ceo-data-leak>.
16. NewsBTC, “Coinbase Uses Electromagnetic Signal Blocking Tents to Secure Crypto,” accessed August 18, 2026, <https://www.newsbtc.com/news/ai-and-tech/coinbase-uses-electromagnetic-signal-blocking-tents-to-secure-cryptocurrencies/>.
17. Reuters, “El Salvador to Transfer Bitcoin Reserves to Multiple Addresses,” August 29, 2025, accessed August 18, 2026, <https://www.reuters.com/world/americas/el-salvador-transfer-bitcoin-reserves-multiple-addresses-2025-08-29/>.
18. Crypto Valley Association Cybersecurity Working Group, “Trusted Key Ceremony Guidelines,” July 2020, accessed August 18, 2026, https://d1c2gz5q23tkk0.cloudfront.net/assets/uploads/2956620/asset/CVA_Trusted_Key_Ceremony_Guidelines.pdf.
19. Republic of Korea, Joint Ministries, “Improvement Plan for Virtual Asset Holding and Management Framework in the Public Sector,” April 10, 2026, accessed August 18, 2026, <https://www.korea.kr/news/policyNewsView.do?newsId=148962458>. For details regarding the NTS public apology and pre-publication review process, see Seoul Shinmun, March 1, 2026 (<https://www.seoul.co.kr/news/economy/2026/03/01/20260301500075>); for the Board of Audit and Inspection’s inspection into property management across four agencies, see Herald Economy, March 4, 2026 (<https://biz.heraldcorp.com/article/10686765>).
20. CryptoCurrency Certification Consortium (C4), “CryptoCurrency Security Standard,” accessed August 18, 2026, <https://cryptoconsortium.org/standards-2/>; NIST CMVP, “FIPS 140-3 Standards,” accessed August 18, 2026, <https://csrc.nist.gov/projects/cryptographic-module-validation-program/fips-140-3-standards>; AICPA, “SOC 1,” accessed August 18, 2026, <https://www.aicpa-cima.com/topic/audit-assurance/audit-and-assurance-greater-than-soc-1>; AICPA, “SOC 2,” accessed August 18, 2026, <https://www.aicpa-cima.com/topic/audit-assurance/audit-and-assurance-greater-than-soc-2/>.
21. The operational mapping between the eight principles and Chapters 2, 3, and 5 is as follows: Principle 1 derives from Section 2.5 (silent failures) and Sections 3.2–3.3 (Lessons 2 and 3), informing Sections 5.1 (selection), 5.3 (Shamir Backup/Seed XOR), and 5.4 (geographic distribution). Principle 2 derives from Section 1.1 (Coldcard RNG flaw), Section 2.1 (generation), and Section 3.1 (Lesson 1), informing Sections 5.1 (device selection) and 5.2 (generation). Principle 3 derives from Section 1.2 (NTS press release) and Section 2.2 (backup), informing Sections 5.3 (backup) and 5.7 (prohibited AI inputs). Principle 4 derives from Section 2.3 (storage), Section 2.6 (Ledger data breach), and Section 3.3 (Lesson 3), informing Sections 5.4 (storage) and 5.8 (incident response). Principle 5 derives from Section 2.4 (signing), informing Section 5.5 (four-step verification) and Section 5.1 (deterministic nonce selection).

Principle 6 derives from Section 2.1 (delayed flaw discovery), Section 2.5 (recovery), and Section 3.5 (Lesson 5), informing Section 5.6 (the small-amount full-procedure test) and the Conclusion. Principle 7 derives from Section 2.4 (trusting the screen) and Sections 3.4–3.5 (Lessons 4 and 5), informing Sections 5.5 (address re-derivation), 5.7 (AI tools), and 5.8 (response workflows). Principle 8 derives from Section 3.4 (Lesson 4), informing Section 5.6 (the small-amount full-procedure test) and Section 5.8 (response workflows). (Figure 14, at the end of Chapter 5, maps these relationships along the key lifecycle axis.)

22. The Bitcoin Hole, “Hardware Wallets” comparison database, accessed August 18, 2026, <https://thebitcoinhole.com/hardware-wallets>. Source dataset available at <https://github.com/thebitcoinhole/database>.
23. Coinkite, “Coldcard Security Advisory,” accessed August 18, 2026, <https://blog.coinkite.com/coldcard-mk3-seed-generation-warning/>; Coinkite, “COLDCARD Security Update: Seed Generation, Transaction Integrity, and Data Isolation,” August 20, 2026, accessed August 30, 2026, <https://blog.coinkite.com/coldcard-security-update-5.6.1-1.5.1q/>. As of August 30, 2026 — this report’s reference date — recommended standard track versions were 5.6.1 for Mk4/Mk5 and 1.5.1Q for Q, with version 4.2.0 for Mk2/Mk3. Edge track releases follow a separate versioning scheme.
24. BIP84, “Derivation Scheme for P2WPKH Based Accounts,” accessed September 5, 2026, <https://bips.dev/84/>; BIP48, “Multi-Script Hierarchy for Multi-Sig Wallets,” accessed September 5, 2026, <https://bips.dev/48/>.
25. Jameson Lopp, “Metal Bitcoin Seed Storage Stress Test (Round IV),” August 2, 2020, accessed August 18, 2026, <https://blog.lope.net/metal-bitcoin-seed-storage-stress-test-iv/>.
26. SatoshiLabs, “SLIP-0039: Shamir’s Secret-Sharing for Mnemonic Codes,” accessed August 18, 2026, <https://github.com/satoshilabs/slips/blob/master/slip-0039.md>.
27. Coldcard Firmware Docs, “Seed XOR,” accessed August 18, 2026, <https://github.com/Coldcard/firmware/blob/master/docs/seed-xor.md>.
28. Unchained, “How to Store a Bitcoin Seed Phrase in a Safe Deposit Box,” accessed August 18, 2026, <https://www.unchained.com/blog/how-to-store-seed-phrase-safe-deposit-box>; FDIC, “Five Things to Know About Safe Deposit Boxes, Home Safes and Your Valuables,” accessed August 18, 2026, <https://www.fdic.gov/consumer-resource-center/five-things-know-about-safe-deposit-boxes-home-safes-and-your-valuables>. Terms and conditions vary by financial institution and should be reviewed prior to use.
29. Bitcoin Core, “deriveaddresses” RPC documentation, accessed August 18, 2026, <https://bitcoincore.org/en/doc/25.0.0/rpc/util/deriveaddresses/>; Bitcoin Core, “HWI,” accessed August 18, 2026, <https://github.com/bitcoin-core/HWI>.
30. Sonya Moisset, “When ‘Private’ Isn’t: The Security Risk of GPT Chats Leaking to Search Engines,” Snyk, August 1, 2025, accessed August 18, 2026, <https://snyk.io/blog/chatgpt-chat-google/>; OpenAI, “How We’re Responding to The New York Times’ Data Demands in Order to Protect User Privacy,” June 5, 2025, accessed August 18, 2026, <https://openai.com/index/response-to-nyt-data-demands/>.
31. Rony Roy, “Coldcard’s RNG Flaw Is Still Draining Wallets, and an AI Audit Just Found 85 More Critical Bugs Across Bitcoin,” crypto.news, August 7, 2026, accessed August 18, 2026, <https://crypto.news/coldcard-rng-flaw-bitcoin-wallet-ai-audit/>

Bitcoin Self-Custody Guide

Self-Custody Guidelines Built From Real Incidents and Best Practices

Researcher

Jeehwan Lee jeehwanlee27@gmail.com

Corresponding Author

Taemin Oh

Editor

Sanghyun Park

Translator

Samuel J. Hahn

The views expressed in this report are those of the author and do not represent the official position of Metanomia.