The handouts, in the order of the work

These are the same sheets as the handouts on the site. If a check fails, the work stops.

- Terms The words used in this guide.
- 03 Project Overview.
- 05 Check the files. Tails, Sparrow 2.5.5, Electrum, the Ian Coleman page, and the dice converter.
- 04 Two boots. Tails is the offline computer. Networking is set to Disabled.
- 06 Make the three keys. Faces 1 through 4 are kept.
- 07 Build the 2-of-3. Sparrow 2.5.5 is the required version.
- 08 Practice spend. A small amount, two signatures, then broadcast.
- 09 Metal and the map. Three titanium plates, three places.
- 11 Hide the wallet map. The passphrase is not a seed.
- 12 If I am gone. Two plates and the map. The words are never typed into a phone.
- 13 If you get stuck.
- 14 If someone challenges it.

Sparrow for Tails is version 2.5.5, file sparrowwallet-2.5.5-x86_64.tar.gz. The Windows installer and the Mac installer do not go on the data USB. A newer download that does not match is a stop, not an upgrade.

These are the words used in this guide, and what they mean.

entropy

Entropy is the randomness that is made with the dice, and it is the material the 24 words are built from.

airgap / Tails, networking off

It is critical that each key is made on a computer that is offline. Tails OS is installed on one of the two USB drives. It can be booted on any computer, and it is designed to forget the session when the machine shuts down. When Tails is booted, networking is set to Disabled.

2-of-3 native SegWit (P2WSH)

This is a wallet that needs any two of three keys before it can spend. One key by itself cannot move the long-term fund.

m/48'/0'/0'/2'

This is the path Sparrow fills in for this kind of wallet. It is confirmed on the screen, and it is not typed from memory.

PSBT

A PSBT is an unsigned payment file that is carried on the USB drive. The seed stays on the offline computer.

descriptor / xpub

The descriptor is the wallet map, and an xpub is a public watch key. Either one can watch the money, and neither one can spend it.

BIP39

BIP39 is the standard list of 2,048 words that a seed is written in. This process uses 24 of those words.

hash

These files are downloaded from their original sources. The fingerprint on that source has to match the fingerprint the computer prints for the saved file. If the lines do not match, the file is deleted.

signature

A signature is the project confirming the file. If the signature fails, the file is deleted and downloaded again.

zero-trust

Zero-trust means a company’s word is not accepted for a download or for a key. Two tools have to agree.

Prepare

- [ ] An established private setting. No camera while the words or the plates are out.
- [ ] One computer, used twice. The first boot watches and never holds a seed. The second boot is Tails, networking off.
- [ ] Two USB drives. One is Tails. The other carries only files that have already been checked.
- [ ] Five casino dice, a pencil, and extra copies of the recording grid.
- [ ] Tails, Sparrow 2.5.5, Electrum, the Ian Coleman page, and the dice converter. Each one checked before it is copied.
- [ ] Three titanium plates. StampSeed is the recommended kit. Only one plate is included, so two more are purchased.
- [ ] A hardware signer is not required to begin. It does not create the key.

Do the setup

- [ ] Run the fingerprint check before the data USB is trusted. A failed check means the file is deleted.
- [ ] Install Tails on its own drive. Networking is set to Disabled before the desktop appears. Persistent Storage is not created.
- [ ] Make the first key with dice, offline. Faces 1 through 4 are kept. Two programs print the same 24 words.
- [ ] Shut down fully. The second key is a new set of rolls. The third key is another new set.
- [ ] Build the 2-of-3 only after no seed is loaded. The path reads m/48'/0'/0'/2'. Export the wallet map.
- [ ] The online computer only watches. A small amount comes in, and the receipt is confirmed in two places.
- [ ] A small amount goes out before any savings. First seed signs. Full shutdown. Second seed signs. Then broadcast.
- [ ] Stamp each plate, read it back, and restore it from the metal. Destroy the paper only after that restore passes.
- [ ] Keep the three plates in three places. The map is not stored beside all three seeds.

House rules

- [ ] No photos of seeds, plates, or screens that show the words.
- [ ] A seed is never loaded on a computer that had the network up in that session.
- [ ] One seed per Tails boot. Full shutdown before the next seed.
- [ ] Persistent Storage is not created, and it is not unlocked.
- [ ] If a fingerprint or a signature fails, the file is deleted. The work stops.

- [ ] A signer can be added later, and only from more than one maker.

The files have to be verified before the computer is taken offline. A file that fails this check is deleted. These files must be downloaded from their original sources. Tails, Sparrow 2.5.5, Electrum, and the Ian Coleman page are not supplied by this site. The dice converter is the one file this site provides. A compromised file compromises the funds.

The files this kit checks

- Tails 7.13. The file is tails-amd64-7.13.img or tails-amd64-7.13.iso. From tails.net only. Wait until that page says the check passed.
- Sparrow 2.5.5. For Tails, sparrowwallet-2.5.5-x86_64.tar.gz. A newer download will not match. That is a stop, not an upgrade. The Windows or Mac installer is only for the computer that watches.
- Electrum 4.8.2. The file is electrum-4.8.2-x86_64.AppImage. From electrum.org. Not a look-alike page.
- Ian Coleman. Save bip39-standalone.html from the official GitHub repo. Open it only while offline.
- The dice converter is already in this kit. Its fingerprint is in VERIFY.md.

Compare the fingerprint

One window is opened, one line is pasted, and the result is compared to the fingerprint on the project’s own site. Every character. One difference means the file is deleted.

Windows. Start menu, PowerShell.

```
$ cd
$ Get-FileHash .\filename -Algorithm SHA256
```

cd is typed, then a space, then the folder is dragged in. Enter is pressed. The second line is pasted with the real file name.

Mac. Spotlight, Terminal.

```
$ cd
$ shasum -a 256 filename
```

Then the kit check

The downloads are placed next to verify_kit.sh. It checks the pinned fingerprints. It does not check signatures. One failure means the work stops. That file is not copied.

```
$ chmod +x verify_kit.sh
$ ./verify_kit.sh
```

Later, two tools must agree

- The same dice bits have to print the same 24 words in the converter and on the offline Ian Coleman page.
- The same seed has to show the same first addresses in Sparrow and Electrum. If they differ, the work stops.
- The wallet map, imported with no seed, has to show those same addresses.
- Before a payment is signed, both tools have to show the same destination, amount, change, and fee.

Next. Two boots. Tails is written to a drive only after this page is done.

It is critical that each key is made on a computer that is offline. Tails OS is installed on one of the two USB drives. It can be booted on any computer, and it is designed to forget the session when the machine shuts down. When Tails is booted, networking is set to Disabled. This is the offline computer. The other USB drive carries only files that have already been checked. The household computer is used first, on the network, and it never holds a seed. It is shut down fully before Tails is booted.

Airgap does mean

- Networking is set to Disabled on the Tails welcome screen, before the desktop appears.
- The seed exists in memory only. Shutdown wipes it.
- Files cross on a clean USB. Those files are the payment file, the wallet map, and the checked programs.
- The household computer stays watch-only. It holds the map, and it never holds a seed.
- A stronger setup removes the Wi-Fi card. Ethernet is never plugged. Tails is still used.

Airgap does not mean

- Networking turned off after Sparrow has already opened.
- Unplugging the cable on a disk that was online yesterday.
- Photographing a seed.
- A password manager, a printer, or a cloud note used as the backup.

Prepare Tails on a trusted online computer

01. The Tails file is downloaded only from the official Tails site. The signing key is fetched from more than one published source.

02. The signature and the checksum are checked before the drive is written. If the check fails, the work stops.

03. Tails is written to its own USB with the official installer. The drive is labeled TAILS. It is not mixed with the data USB.

04. Persistent Storage is not created. The passphrase is skipped. The drive stays amnesic.

05. The data USB receives only files that have already passed. The Sparrow file has to be named sparrowwallet-2.5.5-x86_64.tar.gz. That is the Linux file. The Windows installer and the Mac installer do not go on this drive. The drive is labeled DATA.

Every offline session at home

01. The Tails drive is inserted and the computer is booted. Networking is set to Disabled. Persistent Storage is not unlocked. Tails is started.

02. The data USB is plugged in. Only the file needed for that session is copied onto the live session.

03. The session converts the dice, restores one seed, signs one payment file, or exports the wallet map.

04. The result is copied back to the data USB. Tails is shut down fully. The screen goes dark before the drive is unplugged.

05. Two seeds are never loaded in one session. Networking is never turned on while a seed is in memory.

Do not use Persistent Storage

- Tails will offer an encrypted area on the same USB. It is not created, and an existing one is not unlocked.
- It is not hidden. Anyone holding the USB can see that it exists, and the passphrase can be forced or tricked out.
- The Electrum feature writes the wallet onto the stick. That file contains the seed, protected only by the wallet password. That is a spending convenience, not this setup.
- Extra software and saved settings can undo choices Tails already tested.
- It stays off. The seed stays in memory. Shutdown wipes the session. That is why this USB exists.

Stronger chassis, optional

- An old laptop can be the dedicated Tails machine. The Wi-Fi card is pulled. Bluetooth usually leaves with it. Leaving the

card in the slot and marking it disabled is not the same thing.

- Ethernet is often still on the board. It is never plugged. It is disabled in the BIOS when that option exists.
- The previous owner’s disk is ignored or removed. Tails does not use it. That operating system is not booted.
- The Tails file is still checked. Networking is still set to Disabled. A clean data USB is still used.

One computer is used twice. The first boot watches the wallet and broadcasts a finished payment. It never holds a seed, and it is shut down fully. The second boot is the Tails USB on that same computer, with networking off. It loads exactly one seed, signs, copies the payment file out, and shuts down. The only secret that crosses is a signature file, not a seed. A networked machine with a seed loaded is a single point of failure. That session is treated as burned.

Next. The three keys are made next. A word is not written until Tails is running with networking turned off.

This work starts after Tails has booted with networking set to Disabled, and before any plate is stamped. Five casino dice are required, and every face has to be readable. Faces 1 through 4 are kept. Faces 5 and 6 are rolled again. The work is done three times, for three keys, and the computer is shut down fully between them. An established private setting is required, and there cannot be a camera in the room.

The faces

- 1 is 00. 2 is 01. 3 is 10. 4 is 11.
- 5 and 6 are not bits. Roll those dice again.
- Read the dice left to right, the same way every shake.
- Example: 4, 6, 1, 3, 2 becomes 11, skip the 6, then 00, 10, 01.
- Write only 0 and 1. At least 256 bits are required. More is fine. The extra bits are folded down.

Turn the bits into 24 words

01. Boot Tails. Networking Disabled. Copy the checked converter and the Ian Coleman page from the data USB to the Desktop.

02. Run the converter. Paste the bits. It prints 24 words. Word 24 is the check. Do not invent it. Do not fix it.

03. Open the Ian Coleman page in the offline browser. Entropy type: Binary. 24 words. Paste the same bits. The 24 words must match. If they differ, stop. Do not pick the prettier list.

04. Write the 24 words on the grid. Read them back out loud once.

05. Shut Tails down. Wait until the screen is dark. Then start the next key. Seed 2 is a new set of rolls. Seed 3 is a new set of rolls. Never continue a list from the previous boot.

```
$ cd ~/Desktop
$ python3 bip39_dice_converter.py
```

The last word is a check

The 24 words are not 24 free choices. The last word carries a short check so a copying mistake is caught before metal is stamped. At least two of these checks are used. If they disagree, the work stops.

- Two tools, same bits, same 24 words. Converter, then the offline Ian Coleman page. Do not pick the prettier list.
- Let the wallet refuse a bad list. Sparrow or Electrum, networking off, will open a valid list and reject a broken one. If it will not open, do not change a word to force it.
- The math, if it needs to be seen. Twenty-four words are 264 bits. The first 256 are the dice. The last 8 are the check. Only word 24 can change to make that true. It is not chosen. The converter appends it after the file has been checked.

Next. The 2-of-3 is built next. One seed is loaded per boot. Two lists are never loaded in one session. The recording grid is the next page. Print extra copies.

Dice recording sheet

Read left to right. Write only 00 / 01 / 10 / 11. Leave a cell blank if that die was 5 or 6. Print extra copies — each seed needs ~50–60 shakes.

Seed # 1 2 3 **Date** _____ **Reader** _____ **Checker** _____

#	Die 1	Die 2	Die 3	Die 4	Die 5	Bits this shake
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						

Running bit count (filled pairs × 2): _____ Target ≥ 256, better 320–400. Do not photograph this sheet. Destroy it after the 24 words are stamped and verified.

Sparrow 2.5.5 is the required version, and the signing is done on the Tails USB. Networking stays turned off for the entire session. The file on the data USB has to be named sparrowwallet-2.5.5-x86_64.tar.gz. That is the Linux file. The Windows installer and the Mac installer do not belong on that drive. The fingerprint and the signature are checked on the online computer first. Sparrow is not downloaded from inside Tails.

The terminal is not a test

A few lines that are printed here have to be typed. In Tails the Terminal is opened from the Applications menu at the top left. The screen shows something like amnesia@amnesia:~$. The computer printed that. The line is typed after it. Enter is pressed. The prompt has to come back before the next line.

- The line is typed exactly. Spaces count.
- The ./ in front of Sparrow is not skipped.
- Zero and the letter O are not the same. Lowercase L and the number 1 are not the same.
- If the next line is red or says No such file, the work stops. A fix is not invented.
- Backspace is allowed. These lines cannot break Tails.

Launch Sparrow offline

01. Tails is booted with networking set to Disabled. Tails is started.

02. The archive is copied from the data USB to the Desktop.

03. Terminal is opened. These four lines are typed, one at a time, with Enter after each.

```
$ cd ~/Desktop
$ tar -xvf sparrowwallet-2.5.5-x86_64.tar.gz
$ cd Sparrow
$ ./bin/Sparrow
```

Wait 15 to 30 seconds. A Sparrow window should open. If it does, the terminal is finished for this session.

One seed per boot, then the public key

The vault is not built in this step. One public key is copied per boot. A public key is not the 24 words. It still shows the money later, so it goes on the data USB, not in a photo.

01. Boot with networking off. Launch Sparrow. File, New Wallet.

02. Multi Signature. Native Segwit. 2 of 3. This sets the path. It must read m/48'/0'/0'/2'. If it shows anything else, stop. Do not type a different path.

03. Keystore 1, New or Imported Software Wallet. Enter the 24 words for this seed only. Confirm the path again.

04. On that keystore, export the xpub. Save it on the data USB as s1-xpub.txt, then s2, then s3. Do not apply a finished wallet. Close Sparrow. Shut down. Wait for a dark screen.

05. Repeat for Seed 2. Then for Seed 3. Three boots. One seed each. Never continue from the previous boot.

Build the 2-of-3 with no seed loaded

01. Fresh Tails boot. Networking off. Launch Sparrow. Do not type a seed.

02. File, New Wallet. Multi-signature. Native Segwit. 2 of 3.

03. Import the three public keys from the data USB. Confirm the path still reads m/48'/0'/0'/2'.

04. Export the output descriptor to the data USB. This is the wallet map. It watches. It cannot spend. Shut down.

05. On the everyday computer, import that map into Sparrow as watch-only. This computer never holds a seed.

06. Still offline, on a later boot, open the same map in Electrum with one seed. The first addresses must match Sparrow. If they differ, stop.

Sign in the airgap session

01. Import the 2-of-3 wallet from the output descriptor, or restore one software keystore from one seed.

02. File, Open Transaction, From File, select the .psbt.

03. Read destination, amount, change, and fee against the written notes. Then sign.

04. Export with a new filename such as tx-partial-s1.psbt. Close Sparrow. Shut Tails down fully.

Safety at home

- Keep networking disabled. If it is on, shut down and treat the session as burned.
- Load only one seed at a time. Seed 3 stays boxed unless Seed 1 or 2 is gone.
- The computer is shut down fully before the data USB is carried back to the online computer.
- Do not unlock Persistent Storage on a spend. Restore the one seed into memory, sign, and shut down. Do not let Electrum keep the wallet on the USB.
- Test the whole path with a small amount before savings touch this wallet.

Next. Practice spend. A small amount only. Savings stay where they are until that amount has moved both ways.

A small amount is used, enough to pay a fee, and not savings. The destination, the amount, and the fee are written on paper before the file is exported. Those notes are read again before each signature. The first seed signs on Tails. The computer is shut down fully. The second seed signs on a new boot. Then the payment is broadcast.

Step 1. Build the spend, online, watch-only

01. Watch-only Sparrow is opened. The receive history is checked again so the wallet is the right one.

02. Send tab: destination, amount, fee. Read every field against a written note.

03. Finalize for signing. Export tx-unsigned.psbt to the data USB.

Step 2. First signature

01. Boot Tails, networking disabled. Do not unlock Persistent Storage. Launch verified Sparrow. Or unlock hardware wallet A that already holds Seed 1.

02. Restore Seed 1 only, or confirm Seed 1 is the only key on that device. Open tx-unsigned.psbt. Confirm the fields. Sign.

03. Export tx-partial-s1.psbt. Full shutdown of Tails, or unplug the hardware signer.

Step 3. Second signature

01. Fresh Tails boot, networking disabled. Do not unlock Persistent Storage. Restore Seed 2 only. Or unlock hardware wallet B from a different maker.

02. Open tx-partial-s1.psbt. Confirm the first signature and the unchanged spend. Sign.

03. Export tx-signed.psbt. Full shutdown.

Step 4. Broadcast

01. Watch-only Sparrow: File, Open Transaction, fully signed file.

02. Confirm it is fully signed and the fields are unchanged. Broadcast. Record the transaction id in the recovery workbook, not on this site.

If a key is later put on hardware

- The seed is made on air-gapped Tails first. The finished 24 words are imported. The device does not roll its own randomness for this setup.
- Examples, not a ranking: Foundation Passport, Blockstream Jade, BitBox02, SeedSigner or Krux. Official shops only.
- Two keys on two makers. Never three of the same box. A failed maker is one lost key. The plates and the three places still hold the wallet.
- If Sparrow and Electrum disagree about the payment file, the work stops. A small amount is tested first.

Next. The plates are stamped next. A plate is not stamped until a small payment has gone out and come back.

A small payment has gone out and come back before the plates are stamped. Paper is not the backup. Three titanium plates are required, with one seed stamped on each plate. StampSeed is the recommended kit. Only one plate is included in the kit. Two more plates will have to be purchased additionally. Once all three keys are created and can be reproduced in an offline, air-gapped environment, they can be stamped into titanium. Those plates then have to be kept in three separate places. If two keys are stored in the same house, the separation is gone, and the setup has a single point of failure again.

The plates

- Three titanium plates are required, with one seed stamped on each plate.
- StampSeed is the recommended kit. Only one plate is included in the kit. Two more plates will have to be purchased additionally.
- Cryptotag is the other option. It punches the word numbers, and the kit includes two plates, so one more plate has to be purchased. One method is used on a plate, and the two methods are not mixed.
- Regardless of the product used, three places are required, far enough apart that losing one place does not lose all three keys.
- A photo, a note, a password manager, or an email is not a backup. That copy is a leak.

How to stamp at home

01. An established private setting is required, and there is no camera in the room. The plate sits on a solid block, and each letter is stamped deep.

02. The words are numbered 1 through 24. The plate is labeled quietly, such as S1. A name is not used, and the word Bitcoin is not used.

03. Read the plate back against the paper seed. A mis-stamped word is a bricked backup.

04. Restore from the plate on offline Tails and confirm the first addresses.

05. Destroy the paper only after the plate-restore check passes. Destroy the bit notes in the same pass.

The plate that stays home

The 24 words on the home plate, alone, open one small wallet. Leave a little bitcoin there. A thief who loads only those words sees that balance and may stop.

The same words plus a passphrase open a different wallet for modest spending. Write the passphrase down. Do not stamp it on the plate. Do not keep it in the same drawer.

Those words are still one key of the long-term fund. The fund needs any two plates. Do not put a passphrase on the 2-of-3.

What goes on each plate

Plate A. Seed 1 only. Home safe or fire box. Never Seed 2 or 3 here.

Plate B. Seed 2 only. A second place. Not the same place as A or C.

Plate C. Seed 3 only. A third place. Not the same place as A or B.

Print the wallet map and keep the recovery workbook. Store that packet with Plate A and a copy at one other place, never beside all three seeds. Hardware wallets do not replace these plates. Once a year at home: restore from the plates, receive a small amount, spend a small amount, and update the places. Write a letter that points to this packet and which two places are needed. The letter does not contain the words.

Next. Hide the wallet map. The plates are not the map, and the map is not a seed.

The wallet map exists before this work, and it is not left in a drawer as plain text. The map can watch the money. It cannot spend. Anyone who can read it can see the balance, so it is encrypted. The passphrase that opens the file is not a seed, and it is not a plate.

What is required

- The wallet map from the 2-of-3, on the data USB. It is called map.txt only while the work is happening.
- A passphrase that can be remembered and also written down. It is not one of the 24-word lists. It is not the decoy passphrase.
- Tails, with networking set to Disabled. This is not done on the everyday computer.

The steps

01. Tails is booted. Networking is set to Disabled. map.txt is copied to the Desktop. It is not opened on a phone.

02. Terminal is opened. The line below is run. It asks for a passphrase twice. If the two entries differ, it will not make the file.

03. The result is map.txt.gpg. That file is copied to two USB drives. map.txt is deleted from the Desktop and from the USB. The trash is emptied.

04. The passphrase is written on paper. That paper is kept away from the plates and away from the USB drives.

05. On a later offline boot, the file is decrypted once. The map has to open, and the first address has to match what was written down. If a plain copy was made, it is encrypted again. The plain copy is deleted.

```
$ gpg --symmetric --cipher-algo AES256 map.txt
```

Stop if

- The xpubs are not written on paper just in case.
- A letter code or a homemade cipher is not invented. The command above is the one that is used.
- The passphrase is not one of the seeds, and it is not stamped on a plate. A different passphrase is chosen, and the work is started over.

If this passphrase is lost, the coins are not gone. Two plates can rebuild the map. The path is m/48'/0'/0'/2'. Write that path on the inheritance sheet. It is not a secret.

Next. If I am gone. That page is for a spouse. It does not contain words.

This page is for a spouse who has never used Bitcoin. It is read before a plate is touched. One plate cannot spend the long-term fund. The fund needs any two plates and the wallet map. The words are never typed into a phone.

Do not

- Do not type the words into a phone.
- Do not type the words into a website.
- Do not photograph the plates.
- Do not put both plates into one device.
- Do not buy the devices from a marketplace listing or a link in a message.

Do this

01. Find two plates in two places. Leave the third where it is.

02. The map is found. If it asks for a passphrase, that paper is stored away from the plates. The map is decrypted on a computer that is trusted, or with someone who can follow the map sheet. The map is not the words.

03. Buy a Jade from Blockstream’s own shop. Buy a BitBox from BitBox’s own shop. Two makers. Not two of the same box.

04. Put plate A into one device. Put plate B into the other. Never both plates in one device.

05. Load the map so the devices are looking at the same fund. The path is m/48'/0'/0'/2'. This line is not a secret.

06. Send a small test first. If the test arrives, the fund can move. If the devices disagree, stop and do not force it.

The plate at home is a decoy

The words on the home plate, alone, open a small wallet. That is not the long-term fund. The long-term fund still needs a second plate. Do not treat the home balance as the whole savings.

Next. The person doing the setup, if stuck, uses the next sheet. The spouse does not need that sheet.

What am I actually making?

Three keys are made at home, with dice, on a computer that is not on the internet. Any two of them can spend. One cannot. The keys are stamped into titanium and kept in three places. A hardware wallet is optional, and it does not create the keys.

I already own a hardware wallet. Do I start there?

No. The device stays in the drawer until the plates exist and a small payment has moved. A hardware signer does not create the key. If a key is later loaded onto a device, a second key goes onto a device from a different maker. All three keys are not put on hardware. The plates are the backup.

The download is newer than the version on the sheet.

Stop. This kit checks Tails 7.13, Sparrow 2.5.5 (sparrowwallet-2.5.5-x86_64.tar.gz), and Electrum 4.8.2. A newer file will not match. That is a stop, not an upgrade. Delete it.

The two word lists do not match.

Stop. Do not pick the list that looks nicer. Do not change word 24 to force a wallet to open. Roll again, or find the copying mistake. A fixed word is a destroyed key.

Can I keep the words in a password manager?

No. A photo, a note, a password manager, or an email is one hacked account away from the whole seed. The words go on metal. The map is encrypted. The passphrase for the map is on paper, away from the plates.

What if the map is lost, or its passphrase is lost?

The coins are not gone. Two plates can rebuild the map. The path is 2-of-3, native SegWit, m/48'/0'/0'/2'. Write that path on the inheritance sheet. It is not a secret.

What does my family do?

Give them the If I am gone sheet. They need two plates and the map. They buy a Jade and a BitBox from the makers' own shops. One plate goes in each device. They do not boot Tails. They do not type the words into a phone.

Next. The last sheet is only if someone challenges the method. It is not required to finish the setup.

These are the three questions people use to poke a hole in the method. The answer is only what is true. A step that is not on the sheets is not added.

The words were typed on a laptop.

Yes. On Tails, with networking off, one seed, then a full shutdown. Tails runs from memory and forgets the session. That does not stop a hardware keylogger. One captured seed still cannot spend. The fund needs two keys. Two seeds are never typed on the same boot.

Why not start with a SeedSigner?

SeedSigner is a signer that can be added later. It is not how these keys are made. The randomness is dice. The computer is ordinary, and it is offline. The inheritance sheet is not a SeedSigner guide. It is the letter for heirs: one plate in a Jade, one plate in a BitBox.

If the map passphrase is lost, are the coins gone?

No. That passphrase hides the map. It does not spend. Two plates can rebuild the map if the path is known: 2-of-3, native SegWit, m/48'/0'/0'/2'. Keep that path outside the encrypted file. A homemade cipher is not used.