csiPass User Manual

This guide is for someone who has a csiPass device in hand and wants to use it. It assumes no knowledge of cryptography, and it does not cover building or flashing the device.

It covers every csiPass device rather than one of them. They differ in a few visible ways – the shape of the screen, whether you answer with a button or by touching it, whether there is a coloured light – and those differences are gathered in one place below. Everything after that is the same on all of them.

Before you start: what this device is today

csiPass is a **development prototype**. Registering and signing in against a real website works and has been done. What it does not have is certification, an attestation identity that an organisation could allowlist, or a physical form worth carrying: it is an exposed circuit board, not a sealed key.

Use it with **test accounts only.** Do not make it the only way you can sign in to anything you care about. Every service that accepts a security key also lets you register more than one, and lets you keep a recovery method — use both.

If the device is lost, damaged, or reset, the passkeys on it are gone. Nothing can recover them, by design: the keys never leave the device and exist nowhere else.

What a passkey is

A passkey replaces a password. Instead of you remembering a secret and typing it into a website, the device holds a secret that never leaves it, and proves to the website that it has it.

Two things follow, and both are the point:

- **A fake site learns nothing.** The proof is tied to the real site’s name. A lookalike site gets a proof that is useless to it.
- **A breach of the site leaks nothing you use elsewhere.** The site only ever stores the public half, which is not a secret.

Every sign-in needs the device physically present and a deliberate gesture from a person. Software on your computer cannot make that gesture.

Which device you have

The differences worth knowing before you read on:

	What varies	What it changes
Screen shape	Rectangular or round	Only how the pages are laid out. The same things are on them.
Answering	A dedicated button, or an area of the screen you touch	The gesture. Never who is allowed to make it.
Browsing	A second button, or tapping the screen	Short press or tap moves on; holding opens details.
Light	Some devices have a coloured light, some do not	Nothing. Every state it shows is also on the screen.

Two words are used throughout instead of naming a button each time.

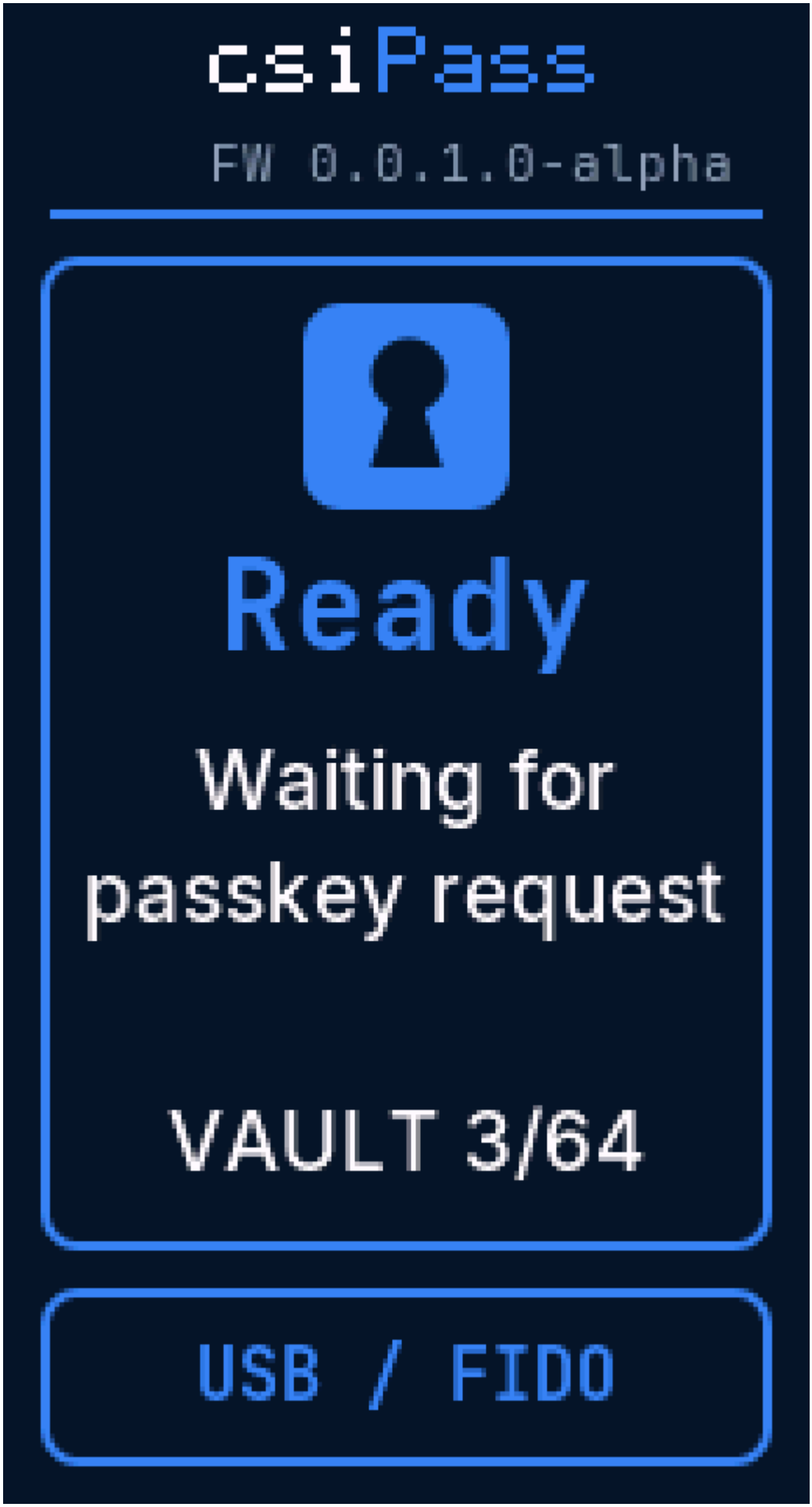
Confirm is how you approve something. On a device with a dedicated button, press that button. On a touch device, touch the area the screen marks for it, which stays in the same place for the life of the device so it can be learned.

Browse is how you look through what is stored: a short press or a tap moves to the next thing, and holding for about three quarters of a second opens the details of what you are looking at.

Confirming and browsing are always separate. Whatever browses can never approve anything, on any device.

Getting to know the device

The screen


csiPass
FW 0.0.1.0-alpha
Ready
Waiting for
passkey request
VAULT 3/64
USB / FIDO

When you first plug the device in, it opens on the **Security** page for a few seconds so you can see how it is hardened, then settles on **Ready**. The Ready screen has three parts. The top strip shows the name and the firmware version. The large middle panel is the message. The bar at the bottom is a hint about what the device wants or what it is doing. On Ready itself that bar reads `USB / FIDO`, and a small status line under the message shows how many stored passkeys it holds – `VAULT 3/64` when they live in encrypted flash, or `RAM 3/64` when they live only in memory and vanish on unplug.

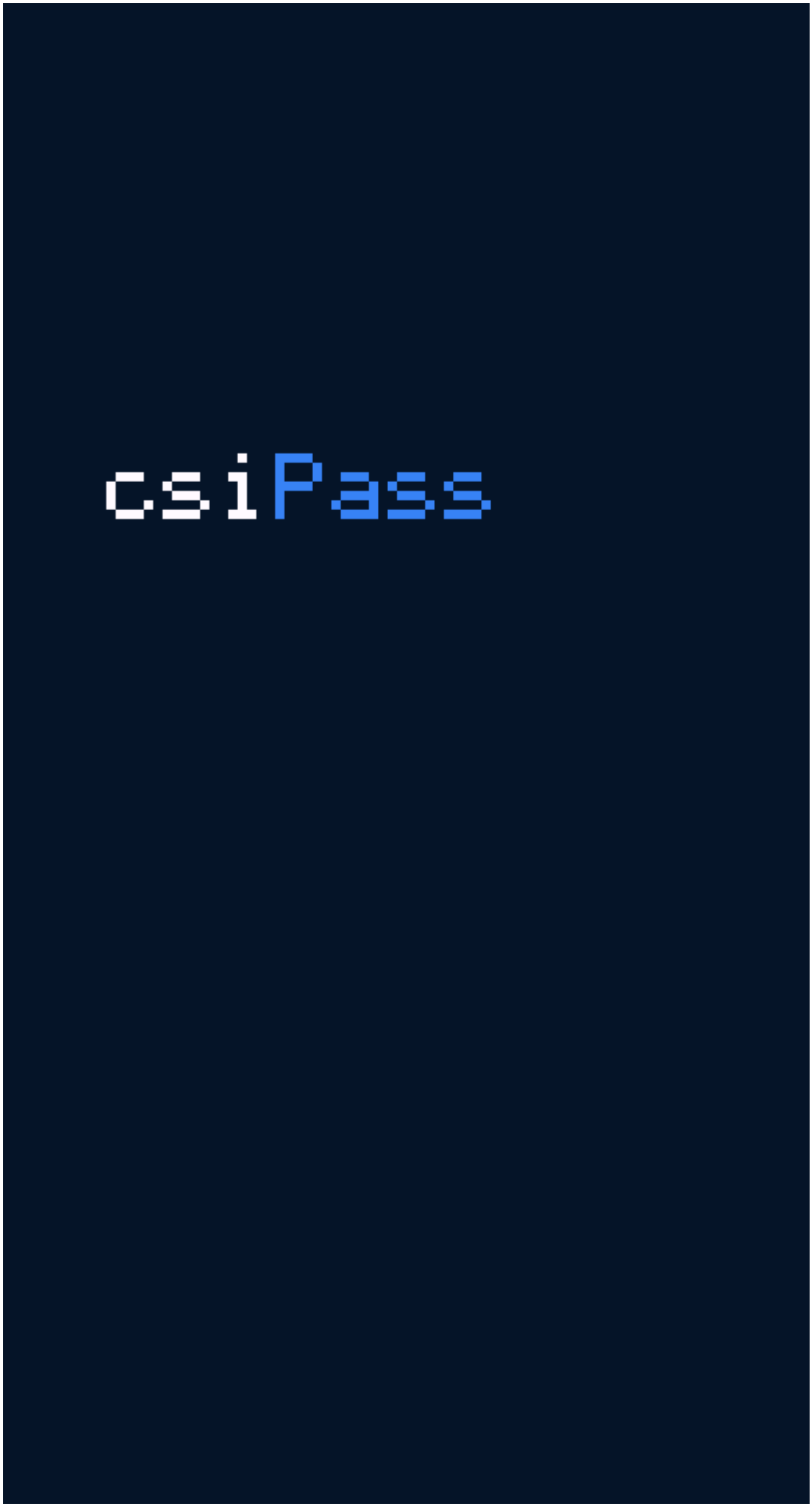
The colour of the border tells you the mood at a glance:

Border	Meaning
Blue	Normal. Nothing needs you.
Amber	**The device is waiting for you to confirm.**
Green	The last thing succeeded.
Red	Something failed, something needs attention, or a destructive confirm is waiting.

When it is waiting for a confirm, the bottom bar usually reads `PRESS BUTTON`, and soft arcs move from the side where the confirm control is – that is how you can tell which gesture it wants without reading the bar.

When it is left alone

After a minute of nothing happening (or the idle delay you set in the companion) the screen dims to about a quarter of its brightness and a screensaver runs. The default is the name drifting slowly around the panel. Other built-in styles (Breathing Dot, Cipher Text, Lissajous, Starfield, Floating Clock) are chosen from the companion Device → Display panel; see the Companion Manual.


csiPass

It is not asleep. Any request from a computer, and anything you press, brings the ordinary screen straight back, and a request waiting for your confirmation stops it appearing at all – so it can never hide something you were meant to see and approve. While idle the processor may drop to a cooler clock (also a Display preference); plugging activity or a press restores full speed before anything you are waiting for.

Its purpose is not decoration. On the models with an OLED panel, a picture that never changes eventually burns into the glass, and something that moves is the answer to that. It also stops the device sitting there telling the room how many passkeys you keep on it.

The light

If your device has one, it breathes blue while idle, turns amber while it is waiting for you, and goes green or red to report the result. It repeats what the screen already says, so a device without one is not missing anything – the screen is always the thing to read.

Everyday use

Registering with a website

1. On the website, choose to add a security key or a passkey.
2. Plug csiPass in if it is not already connected.
3. The screen turns amber and shows **Create passkey** with the site's name and your account under it. **Read them.** This is the moment to notice if the site is not the one you meant.


csiPass
FW 0.0.1.0-alpha
Create passkey
login.microsoft.c
anton@example.
PRESS BUTTON

4. Confirm.
5. The screen turns green and shows **Saved**. On a provisioned device the body says the encrypted vault was updated; on a memory-only board it says the passkey lasts only until restart.


csiPass
FW 0.0.1.0-alpha
Saved
Encrypted vault updated
USB / FIDO

If you set a PIN, the browser asks for it before the device asks you to confirm.

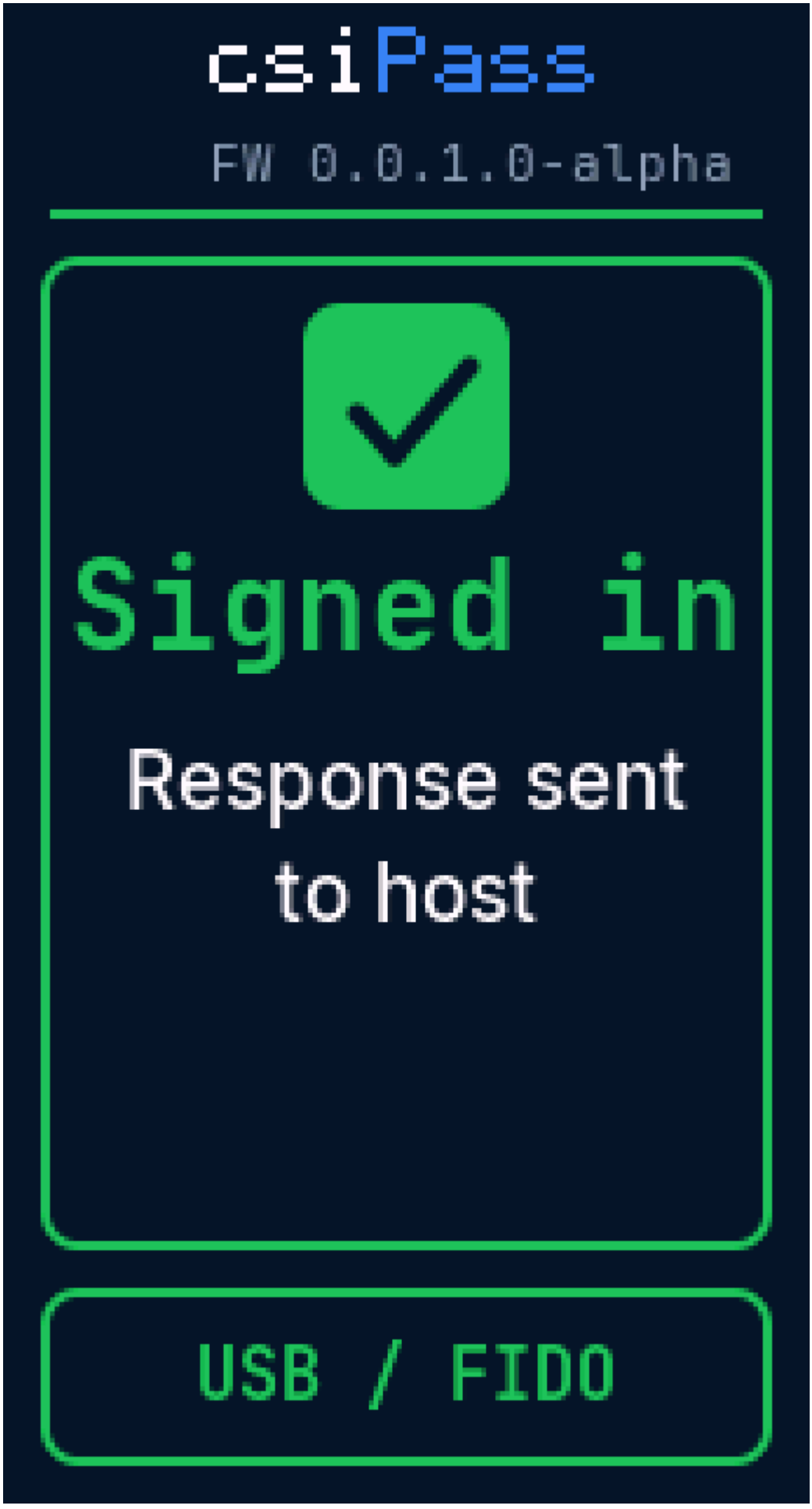
Signing in

1. On the website, choose to sign in with your security key.
2. The screen turns amber and shows **Sign in** with the site’s name.


csiPass
FW 0.0.1.0-alpha
Sign in
login.microsoft.c
anton@example.
PRESS BUTTON

3. Confirm.

4. The screen shows **Signed in**.


csiPass
FW 0.0.1.0-alpha
Signed in
Response sent to host
USB / FIDO

If you have more than one account on that site, the browser will ask which one before the device asks you to confirm.

A sign-in that does not name the site

Occasionally the screen will say **Sign in** and, instead of the site's name, *A site is asking, using the older protocol*.

This is not an error and nothing is wrong with the device. Some services still use an older way of talking to security keys, and that older way sends the device a scrambled form of the site's address rather than the address itself. There is no way back from it to a name, so the device says that plainly instead of showing you something unreadable or, worse, a name it would be guessing at.

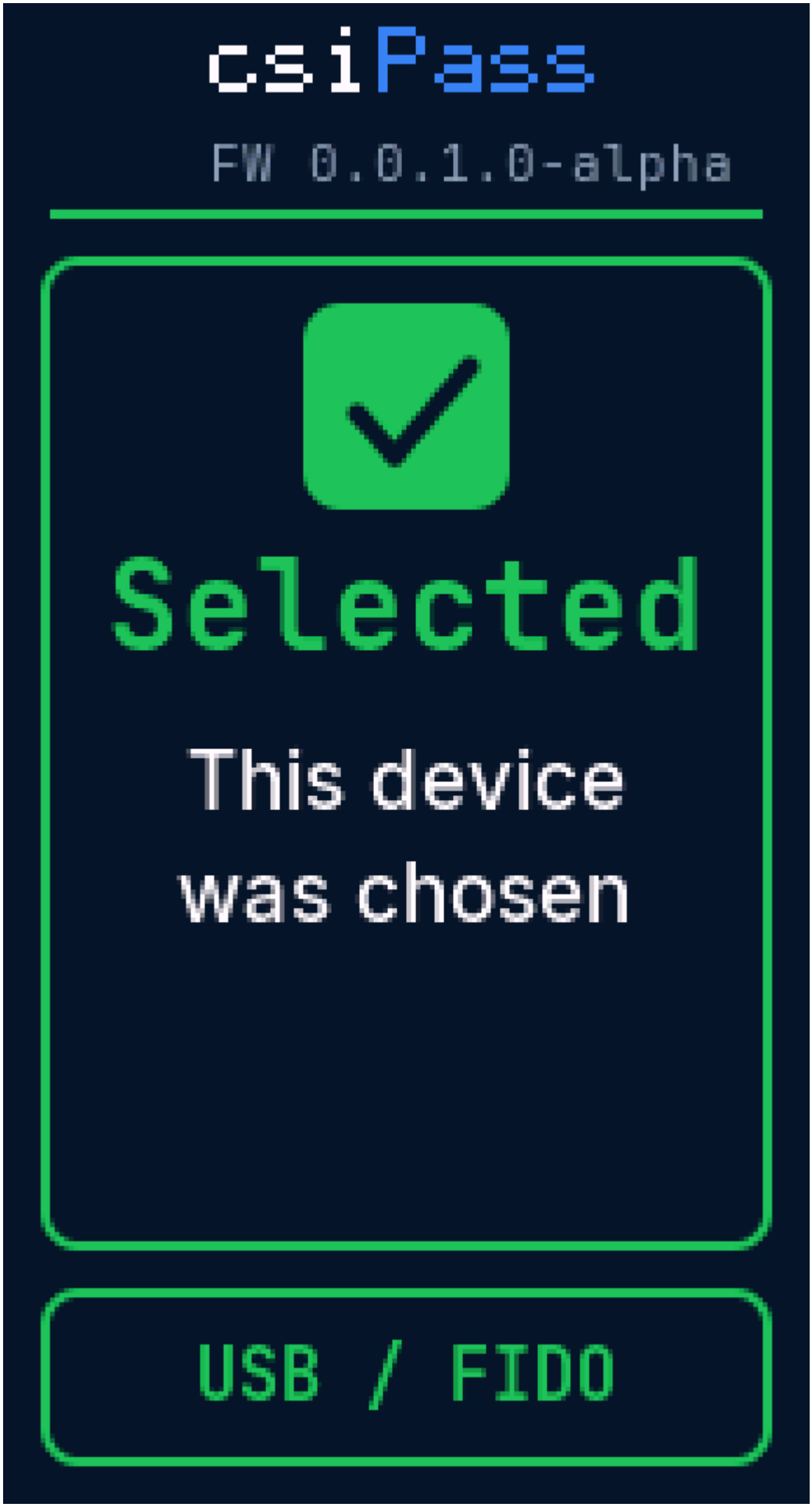
What to do: **treat it with more care than an ordinary sign-in.** Confirm only if you just asked a website to sign you in and are expecting it. If a request like this appears when you were not doing anything, do not confirm — unplug the device instead.

When the computer asks which device

If you have several security keys plugged in, the computer may ask which one you mean. csiPass shows **Select this** with *A host asks which device to use*.


csiPass
FW 0.0.1.0-alpha
Select this
A host asks which device to use
PRESS BUTTON

Confirm on the device you want to use. Nothing is created or signed; you are only pointing at a device. When it works, the screen briefly shows **Selected**.


csiPass
FW 0.0.1.0-alpha
Selected
This device
was chosen
USB / FIDO

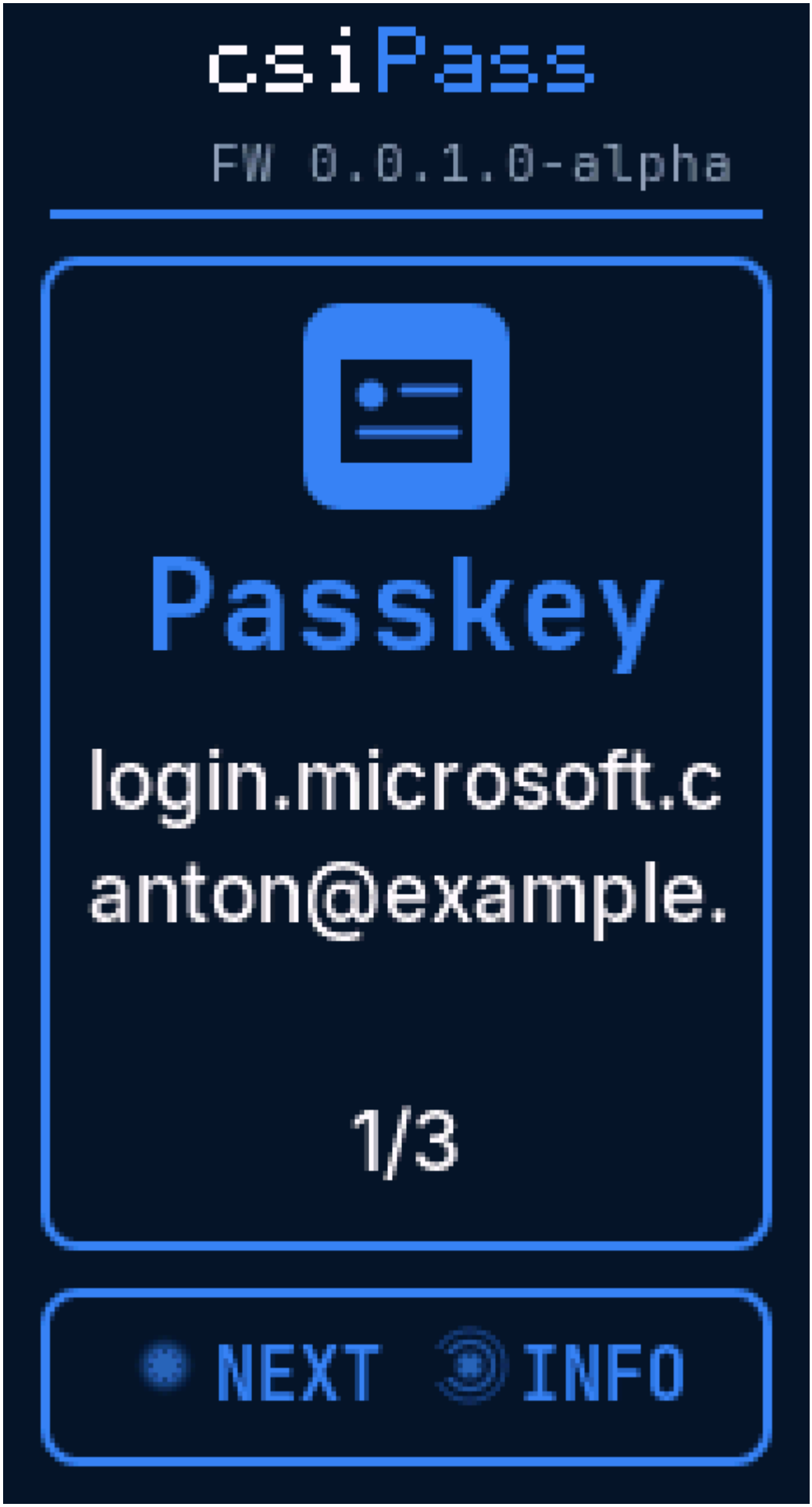
Looking at what is stored

Browse while the screen says **Ready**.

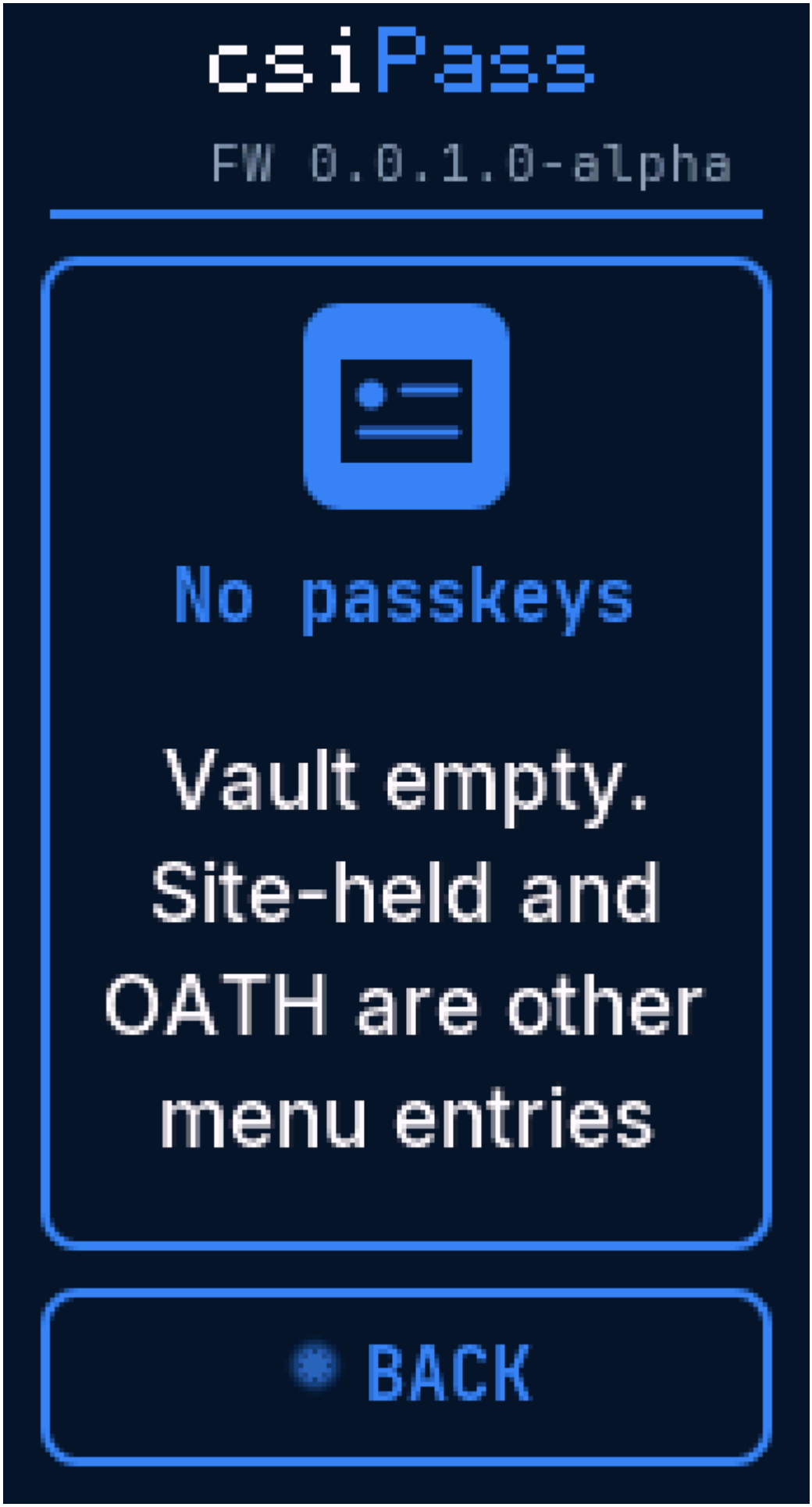
The menu has two levels. Short browse presses cycle the root: **Security** → **Passkeys** → **Site-side** → **OATH**, then back to Ready. Each of those middle pages is an overview. Hold browse on Passkeys, Site-side, or OATH to open that list. Hold on Security opens a list shortcut (**Security long BOOT** in companion Display; default Passkeys).

Browse footers show a soft glowing disc for a short tap and a disc with concentric arcs for a hold, each with a short label (`NEXT`, `OPEN`, `INFO`, `BACK`, `DEL`). Confirm prompts stay as plain centred text (`CONFIRM: YES`, `CONFIRM: CODE`).

In a passkey list, each page is titled **Passkey** and shows the site and account. Short advances; hold opens details. The small number at the foot of the panel is your place in the list.


csiPass
FW 0.0.1.0-alpha
Passkey
login.microsoft.c
anton@example.
1/3
NEXT
INFO

On a device with nothing stored yet, that page says so instead:


csiPass
FW 0.0.1.0-alpha
No passkeys
Vault empty.
Site-held and
OATH are other
menu entries
BACK

Hold on a passkey for details: site and account, then chips for the algorithm, curve, and `credProtect` level (`cP1` .. `cP3`), plus `HMAC` when that extension was stored with the passkey. Short leaves; hold arms deletion.

While browsing, a short press on Confirm returns to **Ready** immediately. When deletion is armed, that same Confirm gesture finishes the deletion instead: the detail stays on screen, the frame turns red, and the same ripple as a sign-in prompt runs behind it.

csiPass

FW 0.0.1.0-alpha

Passkey

login.microsoft.c

anton@example.

ES256 P-256

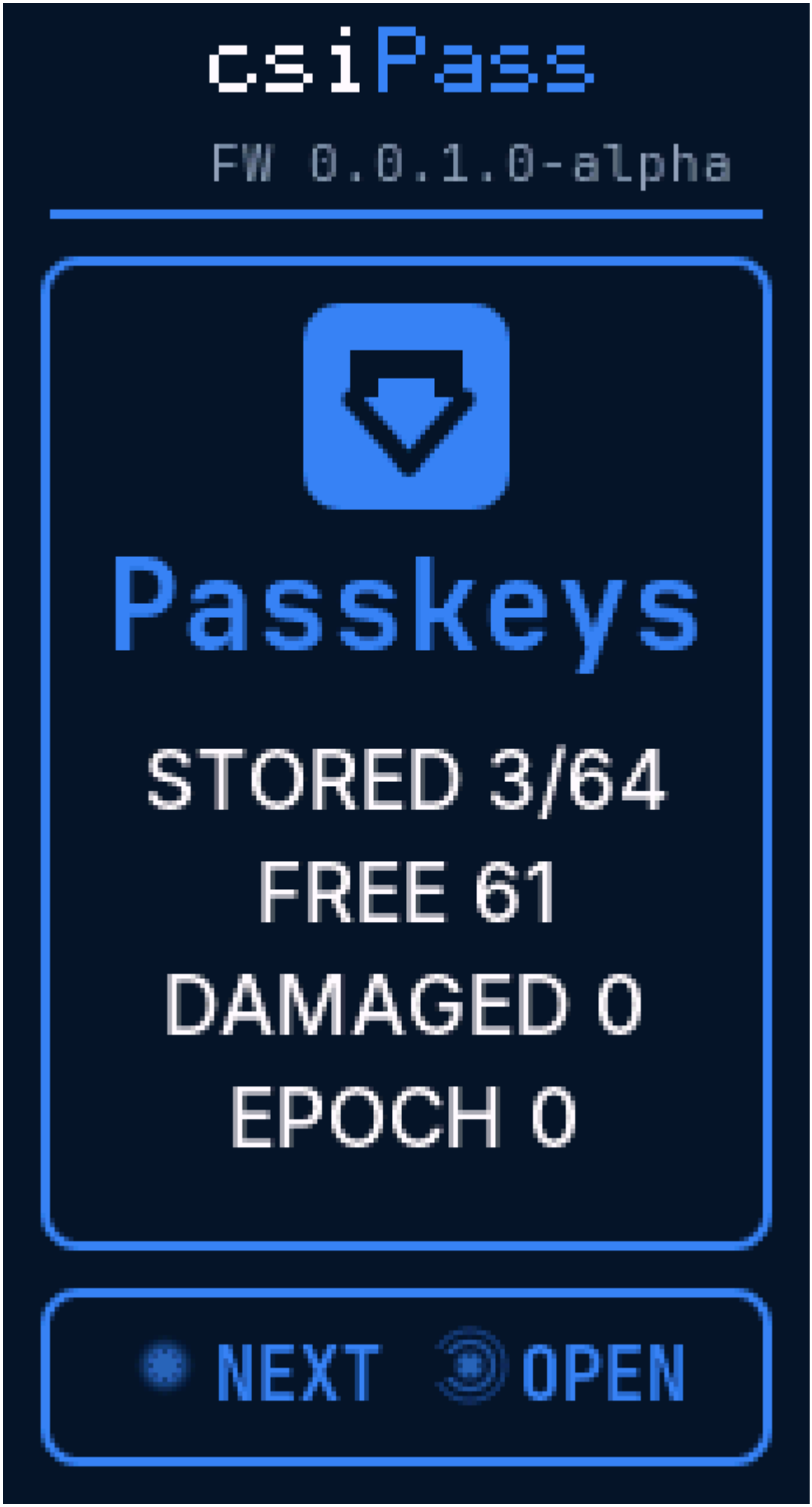
cP2

BACK DEL

A short browse press leaves a detail page. The device returns to **Ready** on its own after about twenty seconds of inactivity while you are browsing.

Site-held registrations

Some sites keep a tag for the passkey instead of asking the device to store it. Those registrations do not appear in the passkey list above. Open the **Site-side** overview from the root menu, then hold to enter the journal – a separate chain from your stored passkeys. Browse only moves inside that chain; Confirm returns to **Ready**.


csiPass
FW 0.0.1.0-alpha
Passkeys
STORED 3/64
FREE 61
DAMAGED 0
EPOCH 0
NEXT
OPEN

If nothing has been recorded yet:

csiPass

FW 0.0.1.0-alpha

Site-held

No site-held registrations recorded yet

BACK

Otherwise each page is titled **Site-held** and shows the site and account. Short advances; hold opens details:

csiPass

FW 0.0.1.0-alpha

Site-held

login.microsoft.c

anton@example.

1/3

NEXT INFO

Hold for details. Chips match a passkey detail, plus `site`. Short `BACK` leaves – there is nothing on the device to delete. Clear these only by erasing the device, or by removing the passkey at the site. The journal holds at most 256 of these registrations; when it is full, the oldest is dropped to make room.

csiPass

FW 0.0.1.0-alpha

Site-held

login.microsoft.c

anton@example.

ES256 P-256

cP1 site

BACK

OATH codes on the glass

One-time codes (HOTP today; TOTP later) share one OATH list, separate from passkeys (up to 256 accounts). Enrol them from the companion Passkeys → OATH tab (paste an `otpauth://` URI or Base32 secret). On the device:

1. Open the **OATH** overview from the root menu, then hold to enter the list.
2. On a list row, **hold Confirm** to show the next code, or hold BOOT for details. Chips on detail name the kind (`HOTP` or later `TOTP`), algorithm, and digits. Confirm on detail also reveals.
3. The code screen shows issuer and account, then digits scaled to fill the space below the account line (with padding). It stays until you press any button (screensaver will not clear it). Codes never go to the computer over USB.

Browse short press leaves a detail; Confirm while not on an OATH detail returns to Ready as usual.

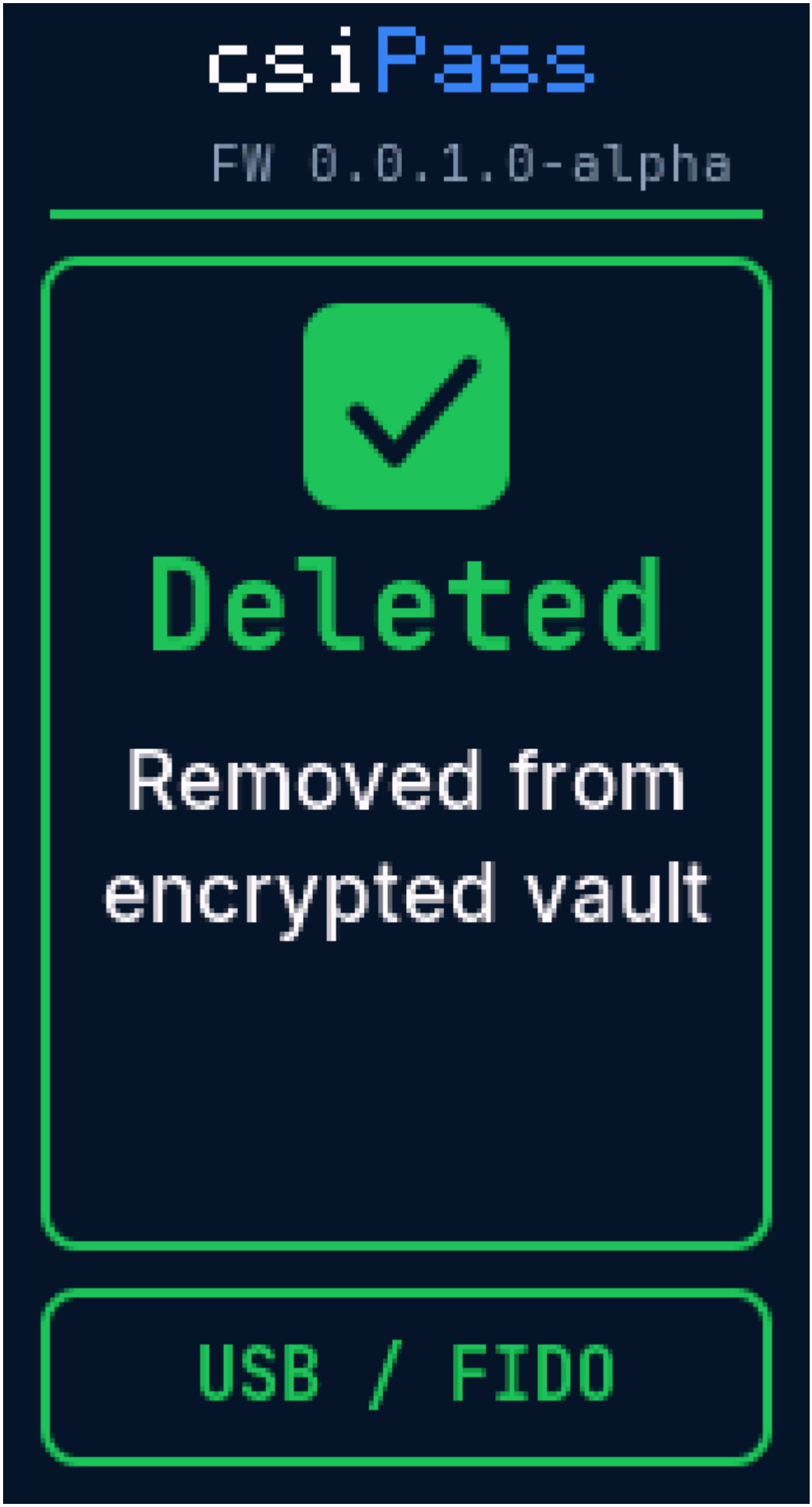
Deleting a passkey from the device

Deleting is deliberate by design, and no single press does it.

1. Browse until you reach the passkey you want.
2. Hold. Its details open.
3. Hold again. The frame turns red; the same site, account, and chips stay on screen; the bottom bar reads `CONFIRM: YES`; and the confirmation ripple runs behind that information, from the same place as a sign-in prompt.


csiPass
FW 0.0.1.0-alpha
Passkey
login.microsoft.c
anton@example.
ES256
P-256
cP2
CONFIRM: YES

4. **Confirm.**


csiPass
FW 0.0.1.0-alpha
Deleted
Removed from encrypted vault
USB / FIDO

Browsing cancels, and so does waiting ten seconds. The way out is the easy gesture and the way through is the deliberate one, on purpose.

Deleting is permanent. If that passkey was your only way into that site, you will need whatever recovery the site offers.

Setting a PIN

A PIN means someone who steals the device cannot use it. Without one, the confirming gesture is the only thing between a thief and your accounts.

Set it through your browser or operating system, not on the device — csiPass has no keypad. In Chrome this is under *Settings → Privacy and security → Security → Manage security keys*; Windows offers it under *Sign-in options → Security Key*.

Rules the device enforces:

- At least 4 characters, and up to 63.
- Wrong PIN eight times in a row and the device stops accepting it entirely. Recovering from that means erasing everything on it.
- Three wrong tries in one session and you must unplug and replug before trying again. This is deliberate: it makes guessing slow.

The device confirms both: it shows **PIN set** the first time (*This device now asks for a PIN*) and **PIN changed** afterwards (*The old PIN no longer works*), in green, so a change you did not make is visible from across the desk.

Changing it. The same place that set it can change it. On Windows: *Settings → Accounts → Sign-in options → Security key → Manage*, press the device's button when asked, then choose **Change** under the PIN. You need the current PIN to do it — a wrong one counts against the tries above, so it is not a way to guess.

There is no way to recover a forgotten PIN. The device does not store your PIN, only enough to check it. If it is gone, the only way back is erasing everything, which is described at the end of this guide.

Reading the Security page


csiPass
FW 0.0.1.0-alpha
Security
ROOT FAKE
BOOT OFF
FLASH OFF...
FAKE ROOT
NEXT

Browse once from **Ready**, or look at the page shown just after you plug in. Four lines in the panel, and the overall verdict in the bottom bar:

Line	What it means
`ROOT`	Whether the device has its own secret storage key. `OK` is good. `NONE` means passkeys live only in memory and disappear when unplugged. `WEAK` is bad — see below. `FAKE` means a bench build is using a stand-in key.
`BOOT`	Whether the device only runs firmware it trusts. `ON` or `OFF`.
`FLASH`	Whether the memory chip is encrypted as a whole. `OFF`, `DEV` (development encryption), or `REL` (release encryption). Prototypes are usually `OFF`.
`EPOCH`	How many times the device has been reset to factory state. Same number as on the Storage page.

The bottom bar is the verdict in plain terms:

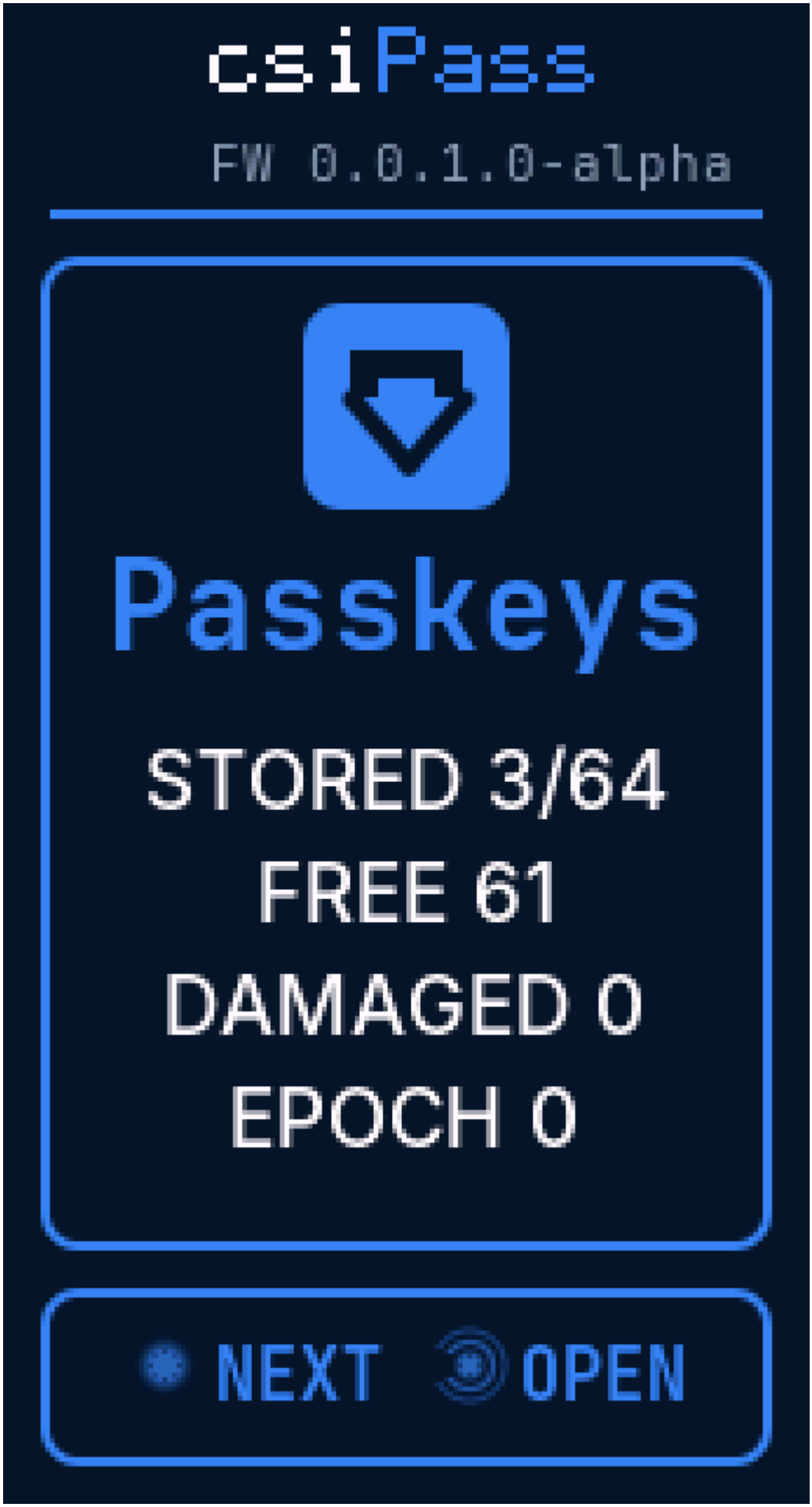
- `DEV MODE` — no storage key. Passkeys vanish when you unplug. Fine for trying it out.
- `VAULT ONLY` — passkeys are stored and encrypted. This is the normal state of a working prototype.
- `PRODUCTION` — everything above plus verified firmware and whole-chip release encryption. No prototype is in this state.
- `INSECURE` — the storage key exists but is readable by software. The encryption is not protecting anything. **Stop using this device for anything real** and ask whoever set it up.
- `FAKE ROOT` — a lab build whose storage key is a stand-in. It behaves like a real one in every other way, which is the point of it, but it protects nothing.

A red border on this page means the verdict is anything other than `PRODUCTION`, which on a prototype is expected.

If the version in the top strip ends in a red `-lab`, you are on a lab image; together with `ROOT FAKE` / `FAKE ROOT` that is the warning not to put a passkey on it that you would mind a stranger having.

Reading the Passkeys overview

From the root menu, open **Passkeys** (or use a Security long-BOOT shortcut to the passkey list). The overview shows how much room is left, and whether any of it has gone bad.

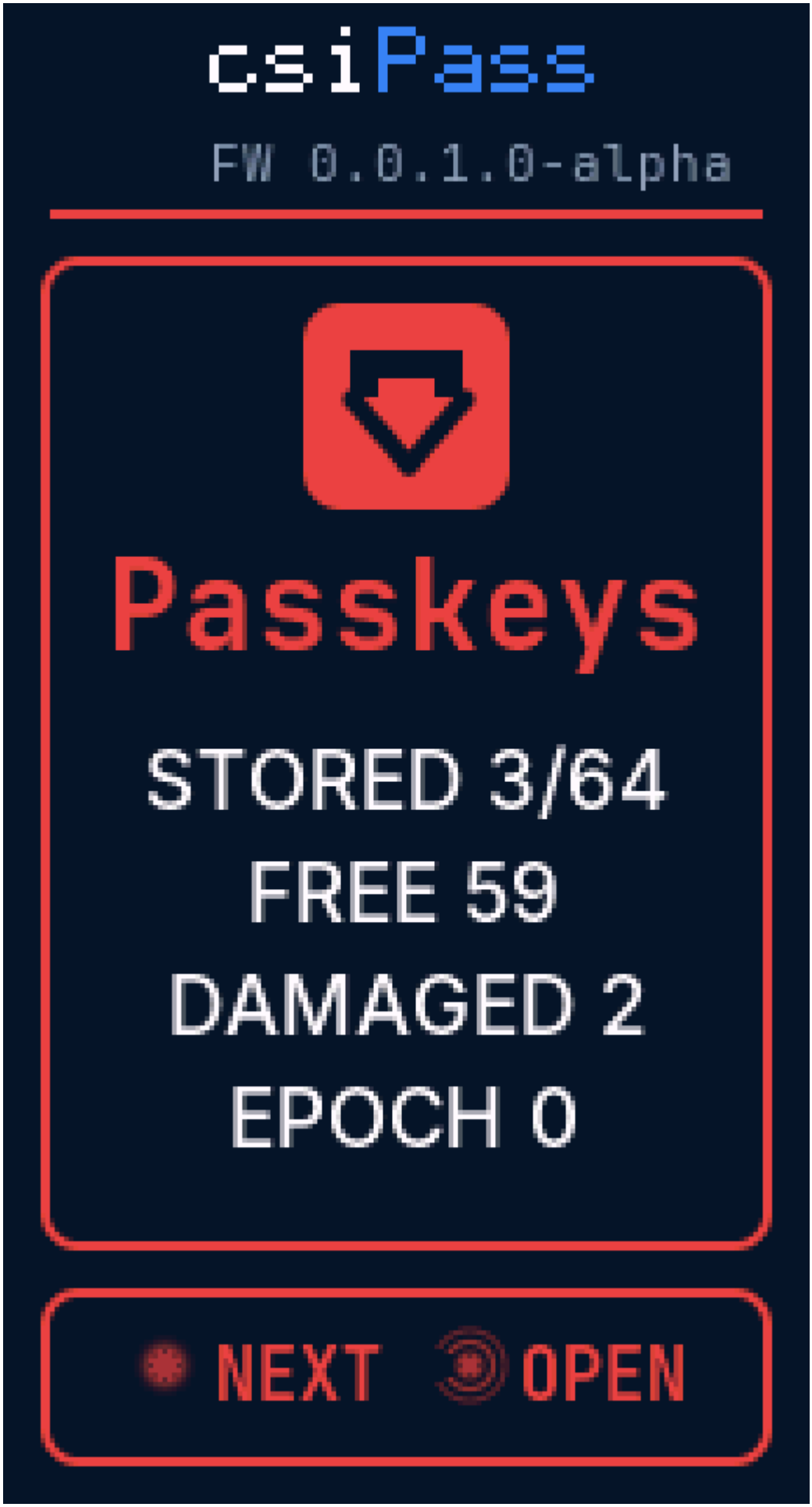

csiPass
FW 0.0.1.0-alpha
Passkeys
STORED 3/64
FREE 61
DAMAGED 0
EPOCH 0
NEXT
OPEN

Line	What it means
`STORED`	Passkeys kept on the device, out of the total it can hold. Not every passkey is one of these – see below.
`FREE`	How many more it will accept.
`DAMAGED`	Slots whose contents no longer decrypt. Normally zero.
`EPOCH`	How many times the device has been reset to factory state.

Footer glyphs: short `NEXT` cycles the root; long `OPEN` enters the passkey list. Site-held and OATH have their own overview pages on the same root cycle. The site-held journal can hold up to 256 registrations; its size does not appear on the Ready line.

On a board with no encrypted vault at all, the overview says `RAM ONLY` instead of the `STORED` / `FREE` / `DAMAGED` / `EPOCH` lines – everything is temporary memory until you unplug.

`FREE` is not simply the total minus `USED`, and the difference matters when something has gone wrong:


csiPass
FW 0.0.1.0-alpha
Passkeys
STORED 3/64
FREE 59
DAMAGED 2
EPOCH 0
NEXT
OPEN

A damaged slot holds something the device can no longer read and will not overwrite, because firmware that destroys data it cannot authenticate is firmware that can be tricked into destroying data. So a damaged slot is neither in use nor available, and `FREE` counts only what a new passkey could actually go into.

Damage is not normal. One damaged slot after an interrupted write is recoverable – the passkey in it is gone, the rest are fine. Several appearing over time means the flash memory is wearing out or the device is being corrupted, and it should stop being trusted with anything that matters.

`EPOCH` changes only when the device is erased, and it is what makes an erase final: every stored passkey is tied to the number that was current when it was written, so raising it makes the old ones unreadable rather than merely forgotten.

Not every passkey is counted here

Some sites ask the device to remember a passkey. Those are the ones on this page and in the list, they take up room, and any of them can be deleted from the device.

Other sites keep a **tag** for the passkey and hand it back each time you sign in. The device reads the tag and works out the secret again from it. Those passkeys use no room at all, so there is no limit on how many you can have. The site decides which kind it wants; you are not asked.

The site does not have your key. The tag it keeps is useless to anyone else: it only means anything to this device, and only for that one site. On another device it opens nothing, and at another site it opens nothing. As with every passkey, the secret itself has never left the device and never will.

What you will notice. For a passkey of this kind you usually have to type your username first, because the site has to know which tag to hand back. For one the device remembers, you can often sign in without typing anything at all – the device finds it. That is the whole practical difference between them.

Two more consequences. The passkey list under Security only shows what the device stored; site-held registrations appear in the separate journal entered from Storage. And a passkey of that kind cannot be deleted from the device, because there is no secret here to delete: remove it at the site, or erase the device, which ends all of them at once and clears the journal.

When something goes wrong

The screen says	What happened	What to do
Timed out	Nobody confirmed in 30 seconds	Try again on the website
Cancelled	The computer withdrew the request	Usually you or the browser cancelled; try again
Request failed	The computer asked for something the device could not do – vault full, passkey already there, or no matching passkey	Try again; if it repeats, the site may want a feature this device lacks, or Storage may be full
No passkeys	Nothing is stored yet	Register with a site first, or open Storage for site-held entries
Not available	The computer asked for a command this device does not implement	Nothing to fix on the device
Insecure root	The storage key is readable	Stop using it for real accounts
Vault error	Stored passkeys could not be read	See below
Delete failed	A passkey could not be removed	Try again; if it persists, erase and re-register
Browse failed	The list changed while you were looking	Browse again from Ready
Hardware error	Something on the board failed at startup	Unplug and replug; if it persists, the device needs attention
USB error	The FIDO USB path failed at startup	Unplug and replug; try another cable and port
Nothing at all	No power, or the screen failed	Try another cable and another port

Two of these are worth seeing before you meet them.


csiPass
FW 0.0.1.0-alpha
Timed out
No confirmation was received
UNAVAILABLE

csiPass

FW 0.0.1.0-alpha

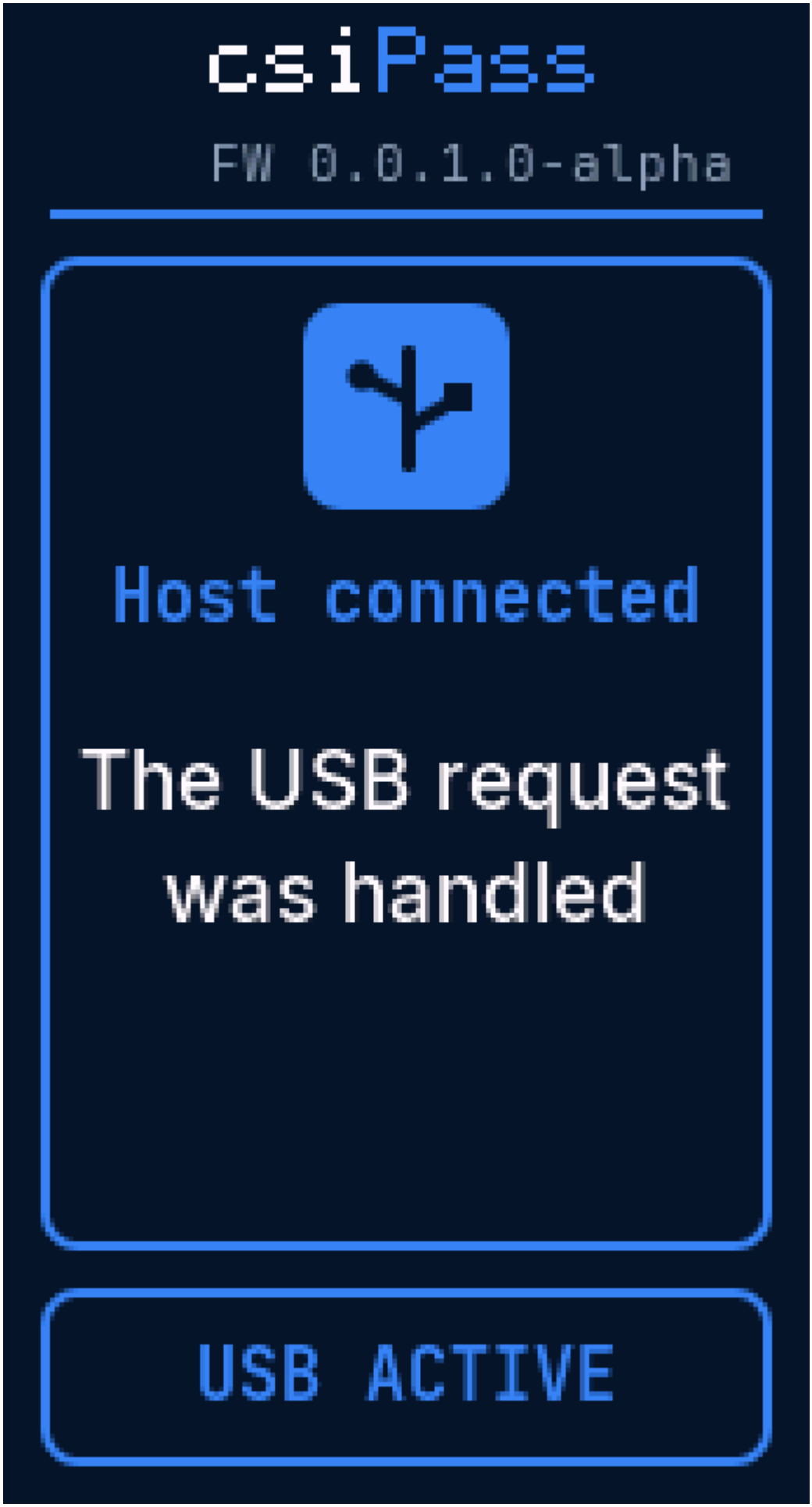
Cancelled

The host cancelled this request

UNAVAILABLE

Neither is a failure of the device, and neither leaves anything half done: a request that was not confirmed simply did not happen.

If the browser does not see the device: try a different USB port and a different cable. Many cables carry power only. The device briefly shows **Host connected** when a computer has actually talked to it over FIDO USB, so that screen is how you tell a dead cable from a dead website.


csiPass
FW 0.0.1.0-alpha
Host connected
The USB request was handled
USB ACTIVE

If you see Vault error: the encrypted storage could not be read. This can happen after an interrupted write. The device refuses to guess at damaged data rather than silently losing some of it. There is no repair; the recovery is to erase and re-register.

Erasing everything

A website or your operating system can reset the device, which deletes every passkey and the PIN together. There is no way to erase only one of the two.

On Windows. Unplug the device. Open *Settings* → *Accounts* → *Sign-in options* → *Security key* and click **Manage**. Plug the device back in, then press its button when Windows asks. In the window that opens, choose **Reset security key**.

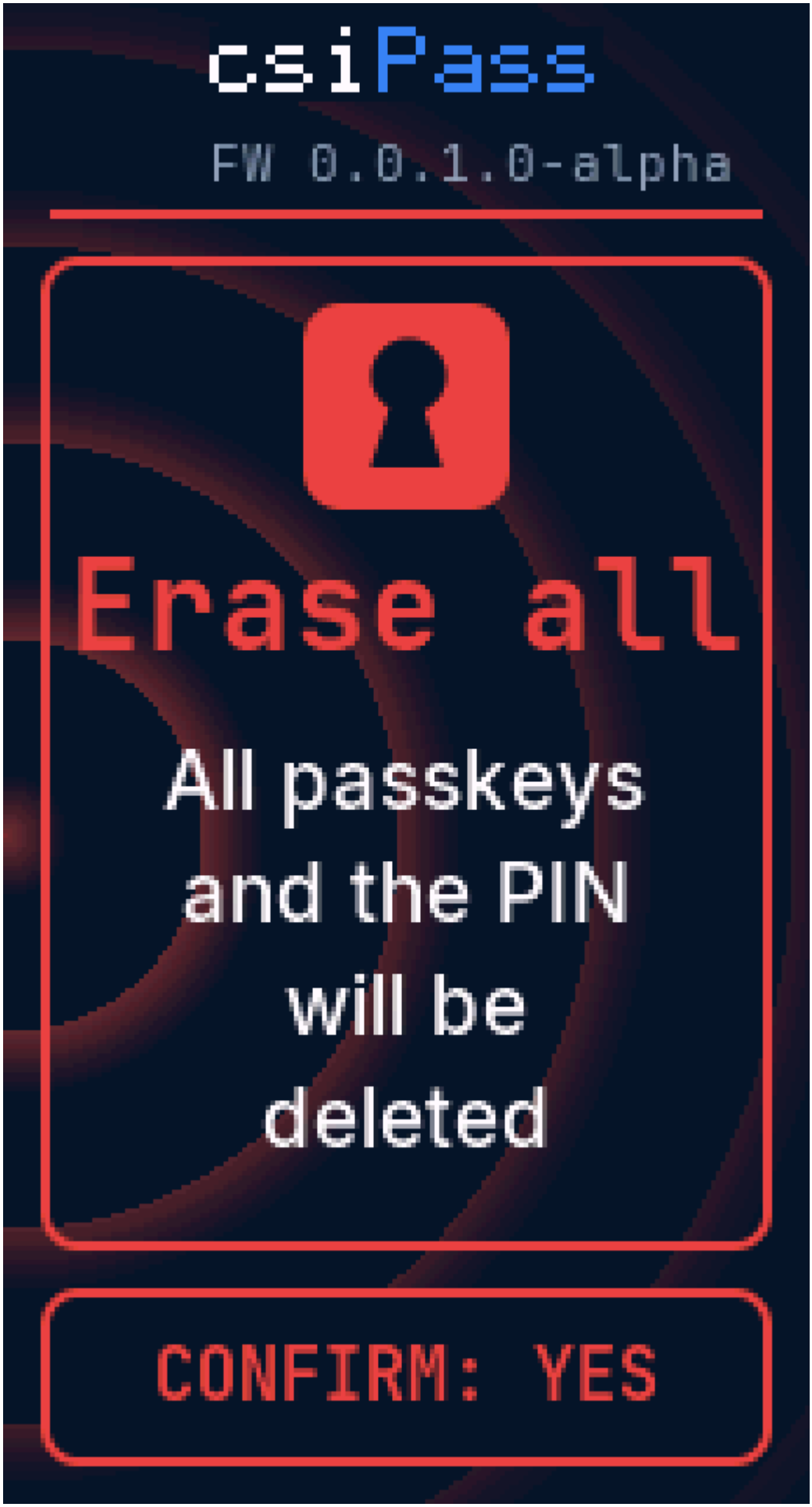
The ten-second rule below means the order matters: get as far as the Manage window first, and plug the device in last.

On any system. Anything that speaks to security keys can ask for the reset — a browser's security-key settings, or your operating system's. What the device requires is the same in every case.

The device only accepts a reset **within ten seconds of being plugged in**. This is deliberate: a computer you left the key plugged into cannot wipe it hours later. If the reset is refused, unplug, replug, and start it again straight away.

The screen turns red and shows **Erase all** with *All passkeys and the PIN will be deleted*. Confirm with the device's own Confirm control. Soft arcs run behind the text, as on any other wait. This cannot be undone.

Afterwards the screen shows **Erased**, the device has no PIN at all, and the next site to ask for one will offer to set a new one.


csiPass
FW 0.0.1.0-alpha
Erase all
All passkeys and the PIN will be deleted
CONFIRM: YES


csiPass
FW 0.0.1.0-alpha
Erased
Encrypted vault
is empty
USB / FIDO

Erasing raises the `EPOCH` on the Security and Storage pages, and clears the site-held journal. That is what makes it final rather than merely thorough: the passkeys that were there are cryptographically unreadable afterwards, not deleted and recoverable.

Things worth knowing

The device shows you the site name before you approve. That line is drawn by the device itself, not by the computer, so malicious software cannot fake it. Reading it is the one habit that makes the device worth carrying.

A gesture cannot be reused. Confirming in advance does nothing, and holding a button down does not queue anything: the device waits for a fresh gesture made after the request arrived, and one gesture approves exactly one thing.

Unplugging is always safe. It cancels whatever was waiting. Nothing is left half-written.

The device does not count your sign-ins. Some security keys report a counter that lets a site notice a cloned key. This one reports zero, which openly says it does not offer that detection rather than pretending to.

Getting help

Report problems at github.com/ABespalov/csipass.

When reporting, say what the screen showed, what website or program you were using, and the firmware version from the top of the screen. Never include a PIN.