Bitcoin Starter Kit

The practical guide to securing bitcoin

Dipun Mistry

Contents

Welcome to the Bitcoin starter kit!

This guide is designed to introduce you to the core concepts of Bitcoin self custody and then build your knowledge as you progress through the various sections. This guide may be useful in a variety of ways, providing educational and interesting information and serving as a practical step-by-step guide.

We will cover topics such as:

- Assessing and applying the appropriate level of security for your needs.
- Exploring safely, avoiding potential security breaches and unexpected compromises.
- Managing and utilising your keys securely.
- Maintaining the safety and security of your bitcoin over time.
- Taking corrective and preventative measures after an incident.
- Future proofing with considerations for disaster recovery and inheritance setups.

When getting started, it makes sense to use the different Bitcoin related apps, services and tools that interest you; play with them, and get a feel for what Bitcoin can do for you. It is NOT necessary to learn every aspect of this guide before exploring Bitcoin; instead, as you broaden your understanding, you can take the steps required to improve the security of your balances as you see fit.

You will get the most from this guide when you decide to start *saving* with bitcoin. There will never be more than 21 million bitcoin in the world, its supply cannot be controlled or manipulated by any government or institution, and it has no limits or restrictions imposed by political agendas. We often find that, for these reasons, learning how to save and protect your bitcoin becomes more important over time, especially as others around you start to understand, appreciate and value bitcoin too.

To complement this guide, we have developed a companion device composed of custom software paired with the Badger2040: a cheap electronic device manufactured by Pimironi in the UK.

- The Badger2040 costs around £16 (at the time of writing).
- It is a simple, secure device that is easy to use.
- It has no wireless transmitters that might leak sensitive data.
- It is pre-built with buttons and a screen, so it requires no electronic wiring or assembly.

Some sections are designed to be thought provoking. We will discuss technical vulnerabilities and how to safely navigate and avoid security breaches and unexpected compromises, and summarise by offerering practical, non-technical solutions that anyone can easily follow.

While some of the threats we discuss may seem unlikely, take the time to consider whether it is really worth the convenience to overlook them; now and into the future. Even if you choose convenience based on your current circumstances, it is always beneficial to understand the risks you are taking and how you might improve your situation in the future if your circumstances change.

Objectives

Educate

Bitcoin provides an alternative to the traditional financial system and can be used for commerce and other financial transactions. Unlike traditional financial systems, which are typically backed by trust in governments and institutions, Bitcoin uses a mathematically secured system called cryptography.

Although this might sound complex, we don't need to understand the all complex mathematics deeply to create a secure wallet. In fact we really only need to worry about **creating large random numbers** and **keeping it secret**. This is the basic principles of "private key" security and will be the main topic of this guide.

This guide will explore the techniques we can use to secure our bitcoin and the tradeoffs we make when using the various types of applications and devices that are available to help us. By gaining the necessary knowledge, you can easily apply the appropriate amount of security for your unique needs.

We will also explore the ways we maintain our security, how we can identify mistakes and warning signs, and the steps we can take to restore security in a timely manner before anyone can take advantage if we ever do compromise it.

The thought of having to learn something novel can be offputting at first, but it is easy to forget how many years we have spent learning about traditional money and yet how little control we have of it. The hope is that in the years to come, the information presented here will become common sense as we come to appreciate the power of controlling our own financial assets.

Empower safe experimentation

Over time, you will discover applications, tools and services that you can use to utilise your bitcoin in different ways. You should be free to experiment with them, but it helps to have a good level of knowledge to prevent you from falling for scams or from losing your bitcoin early on.

Similarly to how banks innovate with debit cards and financial services, the open market is also innovating with Bitcoin. For example, some wallets connect you to a technology similar to the Visa and Mastercard networks in that it enables instant payments and can reduce the cost per transaction, but with a key difference; it is a technology that does not introduce central trusted parties who could be compelled to seize or confiscate your bitcoin.

These innovations do not come without their tradeoffs however; some wallets require the private keys that protect the bitcoin you keep on the wallet to be on an internet connected personal electronic device such as a mobile phone or desktop computer. The information in this guide will not only help you to understand the tradeoffs that come with these innovations, but it will help you to assess them critically and exercise an appropriate level of caution before using them.

Practice

This guide is not purely theoretical. We will walk you through creating your first private key, you will learn practical ways to protect your keys, and the companion device will aid you in performing some of the mathematically complex tasks along the way.

This guide can be used again and again as a reference, or as a tutorial, to help you to create wallets for different purposes and for different situations. If you wish to really internalise the information and be able to work with bitcoin as if it were second nature to you, practice makes perfect.

Not your keys, not your coins

This section will explore investment products and services for comparison purposes. Keep in mind, that this information is for educational purposes only and should not be taken as financial advice. You should do your own research and seek professional financial advice before making any investment decisions.

There are many companies and products that offer bitcoin as a service; that is, they offer to hold and protect bitcoin that you own on your behalf, similar to depositing money in a bank. With such services available to use, often for free too, what is the point of securing your bitcoin by yourself?

These companies and products often provide web services and mobile applications that allow you to access information about your bitcoin deposit, but these companies may not have the actual bitcoin to back the balance on your account, and they can default on your bitcoin deposit at any time.

Some commonly used products are:

- **Exchanges:** where you can trade local currency for Bitcoin and possibly other digital assets.
- **Interest accounts:** where you custody your funds with a company who promises to pay you interest like a bank account.
- **Trading platforms:** where you can manage sometimes automate buys and sells based on market conditions.

Using these types of bitcoin services will reintroduce risks that are associated with traditional finance, such as censorship, political involvement and intervention, and the risk of rehypothecation, bankruptcies, or malpractice by the service provider. These risks may be perceived differently by different individuals depending on their personal situation, but it is important to carefully understand these risks before using these services.

Until you are managing your own keys and interacting directly with the Bitcoin network, you may legally own it, but **you don't *have* the bitcoin**.

An analogy

Imagine you buy a brand new car from a dealership and you decide to have the dealership store it in their garage for safekeeping. One day you go to pick up your car and the dealership tells you that they no longer have it. They claim that they were storing it in a warehouse that caught fire and all of the cars inside were destroyed; also that you are not due any compensation; there was no insurance, no legal protections, and the liability was all yours.

You must now rely on the legal system to fight for a favourable resolution, and of course you have a better chance of this if you can afford good representation or if you know important people in high places. The odds are usually unfairly stacked against the typical individual.

This is the very risk you take when you use custodial services. Upon a default of an asset such as bitcoin, not backed by nor contingent on governments, there is little chance of obtaining recourse or compensation mandated via regulation; there is much less chance of obtaining it in bitcoin or something equivalent in value.

Bitcoin vs trackers

Some custodians do not allow you to withdraw bitcoin, they are typically trading or investment platforms that only allow you to withdraw your national currency. These companies merely provide to you a promise that tracks the price of bitcoin, a promise that may not be backed by it at all.

You may find that they have higher spreads, higher fees, or that they adjust their fees after you have locked in. To take funds out of the platform, you must first convert the asset to your national currency and you will find that as a traditional financial product, it is subject to the rates, fees and terms set by the custodial service provider.

You are not able to participate in the free market, and you are not able to find better rates, because you are not buying bitcoin. These companies offer an investment vehicle, that merely tracks the price of bitcoin.

Some platforms offer ETFs (exchange-traded funds). Although these are also investment vehicles, they may be traded on the stock market, and are usually scrutinized and approved before becoming available to ensure that they meet certain requirements for pricing, valuation, and reporting. Despite this, the same risks as before apply and due diligence must be taken to ensure all the risks are evaluated before investing in general.

It is probably worth repeating here, that this information is for educational purposes only, is not financial advice and does not endorse any particular financial product.

The other hidden risks of trusting a custodian

Custodians are required to hold a large amount of personal and financial information about their clients, including names, addresses, and transaction histories. This is often required due to regulations implemented across the globe, such as AML (anti money laundering) and KYC (know your customer).

These regulations require custodians to collect details and verify the identity of their clients and monitor their transactions for any suspicious activity; of course, different countries implement and enforce these regulations to varying degrees. The process of collecting and storing this information increases the risk of data leaks, where personal and financial information about clients may be accessed by unauthorized parties and results in identity theft or financial fraud.

A custodian may also be subject to pressure from governments or other powerful actors to act in their interests rather than in the best interests of their clients. Many countries carry the risk of government overreach, and custodians may be required to disclose client information or freeze assets at the request of government authorities without requesting your consent, and without you ever being charged of any crime beforehand.

In some cases, this can go beyond traditional financial regulations, such as the case in Canada, early 2022, where the Emergencies Act was invoked and used to freeze the bank accounts, donations, and Bitcoin related exchange accounts of individuals involved in protests relating to covid mandates, without due process. This was done in an attempt to stamp out the protest as a first resort, when other countries had already started relaxing theor mandates and a goal of the protest was to discuss the same for Canada.

Most people think of national currency as physical cash and do not consider the money in our bank accounts; yet most money exists only in digital form in bank accounts. Money should not have inherent features that allow it to be used to manipulate or coerce individuals or groups in order to achieve certain political or other objectives, yet banks have the ability to fully control our money, deterred only by law and regulations, and influenced by politics.

The same is true of custodians holding our bitcoin. Taking custody of your own bitcoin is akin to holding cash in your back pocket or even in a safe, all while remaining in digital form.

Understanding the security of paper, mathematics and electronic devices

As briefly mentioned in the Objectives section, bitcoin's security can usually be boiled down to two things:

1. Create a sufficiently large random number.
2. Keep that number private and hidden.

As an information system, Bitcoin security is not new or unique, but it is unfamiliar to many of us; yet as a monetary system, its security is of utmost importance.

Just like a physical key, a Bitcoin private key is easy to copy if exposed - but in digital form, the chances of others obtaining copies of your key becomes dangerously high due to the existence of malware and hackers, paired with a global internet and local wireless technologies such as wifi and bluetooth.

Although not often spoken about, intelligence agencies such as the CIA are a big threat to information security; not because they are targeting you, but rather because they have been known to insert backdoors into electronic hardware and software in collaboration with hardware manufacturers and developers in secret. This is not a theory, fiction, nor a conspiracy; it is publicly available information. Although we know this much, we have learned that intelligence agencies are good at working in secrecy and we simply cannot know how compromised our devices may or may not be.

The two things that are widely accepted by security experts as potential compromises when using electronic devices are:

1. Data of private or sensitive information can be leaked via communication modules such as wifi, bluetooth or even USB.
2. Hardware can be unknowingly compromised, causing it to produce pre-determined or heavily biased random values in a way that could even be undetectable by anyone other than whoever compromised the device.

This means that we have to be careful about electronic devices. We should **not** trust them to produce a random number, and we should **expect** them to store and attempt to leak all information, including our private keys, without our consent.

Electronic devices provide us with convenience, but when it comes to real security, they make performing proper due diligence and vetting much more difficult.

In this guide, we will explore in more detail how to produce random numbers securely without a computer, and we will explore the ways computers can leak sensitive information and how to prevent it.

TODO… Create installer and document the installer

https://www.raspberrypi.com/documentation/microcontrollers/rp2040.html

Setting up the companion device

In order to follow along in this section, you will need a desktop or laptop computer, and you will need to be computer literate. If you are finding this section difficult, feel free to ask a trusted friend or family member to help; we will not be dealing with any sensitive information in this section.

Setting up the Badger2040 takes 2 steps:

1. Installing the latest firmware.
2. Uploading the custom software.

The firmware is the manufacturer's software designed to work with the peripherals and hardware as it is configured on the device. Once installed, it typically does not need to be updated very often, however it is a good idea, upon receipt, to install the latest firmware to be sure that we have the most efficient, feature rich, bug free firmware available.

The custom software interacts with the firmware in order to communicate with the hardware, and it is responsible for coordinating the user experience, reacting to button presses, and plotting the images for the display to draw; amongst other things. The firmware is pretty dumb but necessary and the software is the brains of the operation.

Installing the firmware

The latest information regarding this part of the setup is detailed as part of the offical getting started guide:

> https://learn.pimoroni.com/article/getting-started-with-pico

We will focus on the relevant parts of the guide here, however some details are subject to change. If you would like to learn more about the device or are having issues following along, consider reading the official guide as well.

In order to install the firmware, we will be:

- Downloading a **flash** uf2 file.
- Downloading a **firmware** uf2 file.
- Connect the device into "bootloader" mode.
- Copy the flash u2f file onto the device.

- Connect the device into "bootloader" mode again.
- Copy the firmware u2f file onto the device.

Preparation

Before we start, we will need to download a couple of files. The first is a factory reset file which can be found on the official RaspberryPi website at:

> https://www.raspberrypi.com/documentation/microcontrollers/raspberry-pi-pico.html#resetting-flash-memory

In the "Resetting flash memory" section on the website referenced above, there should be a link to download a `UF2 file`, and it will dowload a file named similarly to: `flash_nuke.uf2`.

The next file we will need is the latest firmware. This file can be found at:

> https://github.com/pimoroni/pimoroni-pico/releases

Releases will be marked as "Latest" or "Pre-release"; look for the first one marked "Latest" and click on the version number.

Near the bottom of the page, you should see a link to a file labelled similarly to:

```
pimoroni-badger2040-...-micropython-without-badger-os.uf2
```

This is the basic firmware required to operate the device and nothing more. Keep both files in a folder that you can easily find and access later on.

Connecting the device

The next thing we need to do is to connect the device to the computer. The device can be connected using a USB-C data cable and once connected, you will need to take a few extra steps to update the firmware.

On the back of the device, you will see two buttons labelled `boot/usr` (boot), and `rst` (reset), respectively. While *holding down* the boot button, press the reset button to start the device in "bootloader mode". Your computer will detect and notify you of a new external drive connected.

Open the new drive; It may be labelled `RPI-RP2`. You may see the following two files:

- INDEX.HTM
- INFO_UF2.TXT

Copy the `flash_nuke.uf2` file to sit alongside these files. After the file transfer has completed, the device will disconnect itself and restart.

Physically disconnect the device via the USB cable, reconnect, and once again hold the boot button and press the reset button. When you open the drive again, you will see that your `flash_nuke.uf2` file was deleted; this is absolutely normal.

Copy the `pimoroni-badger2040-...-micropython-without-badger-os.uf2` file to the drive. This file will be larger and may take a few seconds to copy, but after completion, the device will disconnect itself and restart again.

At this point, you have installed the latest firmware. Don't forget to clean up and delete the two `uf2` files from your computer; you don't need them anymore.

Installing the software

In order to install the software, we will be:

- Downloading the software.
- Verifying the download.
- Installing **rshell**.
- Uploading the software onto the device.

Creating a key

Let's get straight into it. This chapter will focus on creating a private key. A private key is merely a large random number, however it is important that our number is created fairly using a scientifically sound method.

Handling such a large number with many digits is not easy, in fact it can be very easy to make mistakes when reading, writing, or typing it out. We will transform the number to a series of 12 words which will be much safer to record and easier to protect.

This transformation is not novel, it is a standard known as BIP-39. Storing your words in this standard form is good for a number of reasons:

- The standard has been formally proposed, amended, and published after deep thought and public discourse.
- Standards help to encourage consistent experiences across different products, applications and devices.
- Resources and information can be publicly found online, allowing you to verify the standard yourself.

Feel free to follow along with the instructions as you read this chapter. If you are planning to create a key that you can use securely, consider the following:

- Consider printing the provided instructions and templates before starting.
- Keep devices with microphones or cameras, including mobile phones and laptops, away or turned off.
- Keep your paperwork private and consider how you will destroy them securely after use.

Creating a large random number

As simple as it sounds, coin flipping is sufficient enough to produce our large random number. In order to produce a number large enough, we will need to flip and record our result 128 times in a row.

Imagine recording all heads as the number 1, and all tails as the number 0. After 128 flips, we would have recorded a number in binary that looked like:

```
1001101101110111111011101000011111111110101010001000010101001011
```

. . . but 128 digits long.

This represents a number within a range larger than the number of **milliseconds** that has elapsed since the theorised **big bang** over 13.8 billion years ago: 1,152,921,504,606,846,976 times larger!

Tips for flipping

There is no such thing as a weighted coin

Regardless of how smooth or rough a coin is, how heavy it is on one side, what material it was made or composed of, or how large or mis-shaped it is, a flipping coin will not favour one side over the other while in the air.

Don't let the coin hit the table

It has been theorised that the edge of a coin can cause a bias in the result of a coin flip favouring one side over the other if it hits a hard surface. It is known that a bad edge can affect the result of a coin spin, and it has been observed that some coin flips will spin on a hard surface before landing on a side. The best thing is to flip, catch and reveal.

Don't worry about things like how your hand affects the results or whether you place the coin on the back of your hand before revealing or not. These factors may affect the outcome, but they don't introduce any biases in favour of one side vs the other.

Don't worry about repeating results

Humans are terrible at randomness, we are just wired to look for patterns. When we see the same result multiple times in a row, we feel uneasy because we internally expect "random" to mean "constantly changing", but really it means "unpredictable".

Don't re-flip or change your results just because it feels wrong, you'll only be reducing the randomness of the final answer.

Flip high and fair

I'm sure you know, if you make small, slow controlled flips, you can easily force the side you want to show up. That said, a reasonable height and a reasonable flip speed is more than enough to make the result completely unpredictable.

You don't need to overdo it, just aim to keep the coin in the air for at least 1 second, and ensure that the coin is flipping rapidly in the air.

If you are not familiar with flipping a coin, you can roll it in your hands like a die and throw it into the air, allowing it to rotate in the air before landing onto a soft cusion.

Recording the results

Hopefully by now, you are convinced that creating a massive, truly random number is easy enough to do without a computer.

We have provided a worksheet to help record your flips, you can record them as X's and O's, but something that will become very useful later on, is recording your flips in groups of 11 and then prefixing each group with an O.

The reason why will be apparent when we get to transforming to plain words.

You may record heads as O and tails as X.

Understanding the checksum

You should have noticed by now, that groups of 11 flips make 12 rows, but not completely. There is still space for 4 more results.

This space is reserved for the "checksum". A checksum is a short code that has been calculated from some data, for which we wish to provide integrity. The checksum can be calculated and re-calculated at any time and the result should always be the same.

We'll call it a checksum-code for the rest of this section.

Our new large random number is our data, and if even a single digit of that data was changed after creating our checksum-code, the newly re-calculated checksum-code would no longer match our pre-calculated one.

This becomes useful for detecting typos and other human errors. When we enter our key(s) into a Bitcoin wallet, the software will extract and use the checksum-code to perform validation and provide a level of confidence that there hasn't been any funny business or typing errors before moving on.

One thing to note, is that calculating the checksum-code requires:

1. Using very complex mathematics (often delegated to some sort of computer)
2. Revealing your secret random number to whatever device is doing the calculation

We will deal with the checksum at the end, but for now we will focus on working exclusively on the first 11 rows.

Transforming to plain words

Typically, this would be the last step after calculating the checksum, but doing it now will illustrate a point: the key really is made up by coin flips. Later when we calculate a checksum and complete the key, you will be able to see clearly how little of a change it made to your key

We will transform our flips in two steps:

1. Convert our X's and O's to three character "hexadecimal" codes
2. Use a table to look up and map our codes into words

The three character hexadecimal codes

A quick primer on hexadecimal codes:

- We are most used to handling numbers using the numbers 0-9 for each digit, we call this representation "base 10".
- Computers often use binary to represent numbers, using only 0 or 1 for each digit, we call this "base 2".
- When dealing with large numbers, we often use the "base 16" representation, this is where each digit can be represented by one of the following characters: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.
- An example of a hexadecimal code would be 3AF, which is equivalent to the number 943 in base 10.

To start, we can use our "hex lookup table". In groups of 4, we can map our X and O's to a single character, leaving us with 3 characters per row. As an example, a series such as `OXOX` `XOXO` `XXOX` would translate to `5` `A` `D`.

This is where the prefixed O became useful, this prefix allowed us to split each row equally into 3 groups of 4 results for this exercise.

We can also calculate the code for the twelfth row, but with only 8 out of 12 results, we will not be able to calculate the last character of the code. There is not enough information to find the twelfth word just yet, but calculating the first two characters will still become useful later on.

Mapping codes to words

The final step is to use the "mnemonic words lookup" book to convert each three digit code into a word. For example, `5 A D` would become `remember`.

You will be able to look up 11 of the 12 words that you need. The order of words are important, don't mix them up.

Calculating the checksum and final word

In order to get our final word, we need to complete the last code, and in order to do that, we need to calculate the checksum.

Unfortunately, this step requires some sort of computer, and you may recall that we previously discussed how we cannot trust a computer to create a random number for us, **and** we cannot trust a computer to keep secrets and not leak them.

The random number is very precious, it is unique, newly discovered, never before seen, and central to our security. If anyone were to obtain this number, they could potentially discover and steal our bitcoin.

This is where we must make a calculated tradeoff. Fortunately, by generating our random number by hand, we only need to consider the risk of data leaks. As mentioned in the introduction, the companion device has no wireless transmitters that might leak sensitive data; however it does have a USB connector and persistent storage.

For the sake of brevity, We will focus on using the companion device to calculate the checksum, however to learn more about the risk factors of the device, read XXXX. To learn of some alternative ways to generate a checksum without this device, read XXXX.

Using the companion device

In order to stay secure, we should power the device using a wall charger; this will prevent any data from being leaked through the data pins of the USB cable.

Once the device is powered on, select the XXX option.

TODO: Write code and document this.

Using your first key

Congratulations! You have created your first private key!

By now you have your first set of 12 words, but it's pretty useless by itself. In order to use the key, you're going to need some sort of wallet. Bitcoin wallets come in many shapes and sizes, from specialised hardware (often called signing devices or hardware wallets), to mobile applications running on your personal smart phone.

This is the moment that we decide the fate of our new key:

- Is it going to store a small amount of bitcoin?
- is it going to be used to secure our savings?
- or will it be completely offline and used only for receiving money until some time in the future? – like a one time use piggy bank, but made from several inches thick steel.

Most of us will start by creating spending wallets and then progress to making saving wallets. We will experiment with many different types of spending wallets as we learn more about the ways we can use our bitcoin, but somewhere along the way, we will want to keep some bitcoin safely aside, somewhere that we can top up at any time and where its security is unaffected by our daily activities.

It is important to separate the concept of a **private key** and a **wallet**. The term wallet is pretty overloaded with respect to bitcoin, but in general a wallet is a piece of hardware or software that stores, manages, or uses your private keys (and in some cases, your public keys).

After obtaining a wallet, you may discover that it offers you the ability to create multiple wallets. This is because the term wallet also refers to the

this symbol: →.

```
#!/bin/bash
ARGS=$(echo→${@^^}→|→tr→-d→"→")

n=${#ARGS};→[→$n→-ne→35→]→\
→→&&→echo→"E:H35→$n"→>&2→\
→→&&→exit→1

HEX=$(fold→-w→3→<<<→"${ARGS}0")
```

```
BIN=$(bc→<<<→"obase=2;ibase=16;$HEX"→\
→→|→xargs→printf→"%012d\n"→\
→→|→rev→|→cut→-c-11→|→rev)

OUTPUT=$(echo→"obase=16;ibase=2;"→\
→→$(printf→"%s"→$BIN)→|→bc)

HASH=$(echo→"${OUTPUT:0:32}"→|→xxd→-p→-r→\
→→|→openssl→dgst→-sha256→|→cut→-d→"→"→-f2)

echo→${HASH^^}→|→cut→-c1
```

```
#!/bin/bash
ARGS=$(echo ${@^^} | tr -d " ")
grep -qvEx "[0-9A-F]{35}" <<< $ARGS \
 && echo "BAD INPUT" && exit 1
HEX=$(fold -w3 <<< "$ARGS"0)
BIN=$(bc <<< "obase=2;ibase=16;$(sed "s/$/%800/" <<< $HEX)")
ENT=$(bc <<< "obase=16;ibase=2;$(printf "%011d" $BIN)")
SUM=$(xxd -p -r <<< ${ENT:0:32} | openssl dgst -sha256 | cut -d " " -f2)
echo ${SUM^^} | cut -c1
```

```
#!/bin/bash
# Uppercase and remove spaces
ARGS=$(echo ${@^^} | tr -d " ")

# Validate: 35 characters hexadecimal
grep -qvEx "[0-9A-F]{35}" <<< $ARGS \
  && echo "BAD INPUT" && exit 1

# Append 0 for last character and split into 12 chunks
HEX=$(fold -w3 <<< "$ARGS"0)

# Convert to base2 (binary) and use the mod
# (%) oparator to isolate the first 11 bits.
BIN=$(bc <<< "obase=2;ibase=16;$(sed "s/$/%800/" <<< $HEX)")

# Pad each chunk to 11 digits (prefixed with 0)
# and concatenate to create a single 132 bit number
# then convert to base 16 (hexadecimal)
ENT=$(bc <<< "obase=16;ibase=2;$(printf "%011d" $BIN)")

# Take first 16 bytes (128 bits) and get SHA256 hash of data
```

```
SUM=$(xxd -p -r <<< ${ENT:0:32} | openssl dgst -sha256 | cut -d " " -f2)

# Uppercase and print first character
echo ${SUM^^} | cut -c1
```

280/280 characters:

```
#!/bin/bash
A=$(tr -d " "<<<${@^^})
grep -qvEx "[0-9A-F]{35}"<<<$A &&echo X &&exit 1
H=$(fold -w3<<<$A)0
B=$(bc<<<"obase=2;ibase=16;$(sed "s/$/%800/"<<<$H)")
N=$(bc<<<"obase=16;ibase=2;$(printf "%011d" $B)")
C=$(xxd -p -r<<<$N|openssl dgst -sha256|cut -d " " -f2)
cut -c1<<<${C^^}
```

Security pages

The companion device

TailsOS

Other hardware wallets