Arctica Protocol Sketch v1.1

A secure & private Bitcoin cold storage solution:

Arctica is a Free & Open Source script that installs Bitcoin Core and then walks users through the setup of a secure & private cold storage wallet.

* Private keys are never on any device with a channel to the Internet except through QR codes. This is the best and most secure airgap.

* Arctica uses an encrypted 5 of 7 decaying multisig for bitcoin storage. Our protocol allows up to 4 keys to be stolen without losing any bitcoin, and allows 3 locations to be stolen by an Adversary before privacy is lost. This enables recovery, redundancy, security, and privacy all at once.

* HD Multisig is used so that you can send funds to 1,000s of addresses, but recover all funds using only 5 seeds, which eventually decays down to 1 seed after a long time frame.

*All 7 signing keys in the multisig have their own single signature account of bitcoin stored on them (of ~$1,000), so that any Adversary who finds one part of the multisignature fund is incentivised to steal the smaller 'prize', rather than to continue to hunt for 4 more signing keys. Stealing this will secretly alert the User to move the full fund to a new account.

* Unlike typical multisignature wallets, the capture of a single signing device or backup does not reveal your entire balance and transaction history thanks to the encryption scheme.

* Generic computing hardware is used. Hardware sold specifically for bitcoin storage requires too much trust in the manufacturing of the device, and represents a large supply chain attack risk.

* Minimal software beyond Bitcoin Core. Bitcoin Core is the reference implementation and most trustworthy bitcoin software.

* Open source and easily audited. This makes it less likely to contain a critical security flaw that has not been identified and fixed.

* Practical for non-technical users. By following simple instructions, users with moderate computer literacy can use Arctica. This is important because trusting someone to help you establish your cold storage solution introduces considerable risk.

* Private data stored in small, non-descript packages that are geographically distributed. This way, backup data is easy to hide.

* Your bitcoin are private and counterfeit proof. Arctica uses a Bitcoin Core full node.

This is a 'dumb' node in the sense that there is no public key or private data on it (so there is no way for an Adversary to steal funds, or see how much money you have). It merely keeps up to date with the blockchain, so you don't have to wait to sync when you want to spend

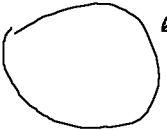
Online Node

(Also Signing Laptop for Attic and Bank Safe SD Cards)

See page 8 for details on the locations and names of each SD Card

2 SD cards ('Attic' and 'Bank Safe') are signed via the Online Node, ie Red. This is because the multisig descriptor (which each SD Card has encrypted - see details on pg 6) has to gain access to the UTXO set somehow in order to create the spending request - it can't be entirely offline. Having a 2nd SD be able to be signed with RED just increases ease of use, and they are both near home anyway, so this does not increase risk any more. There are still 3 more signers required, across 3 Offline Laptops, so this is not a security risk.

Create Spending Request

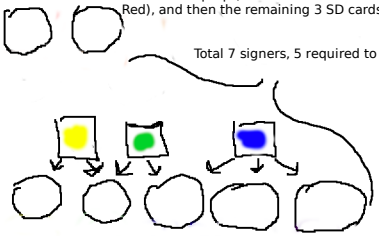


PSBT Coordinator

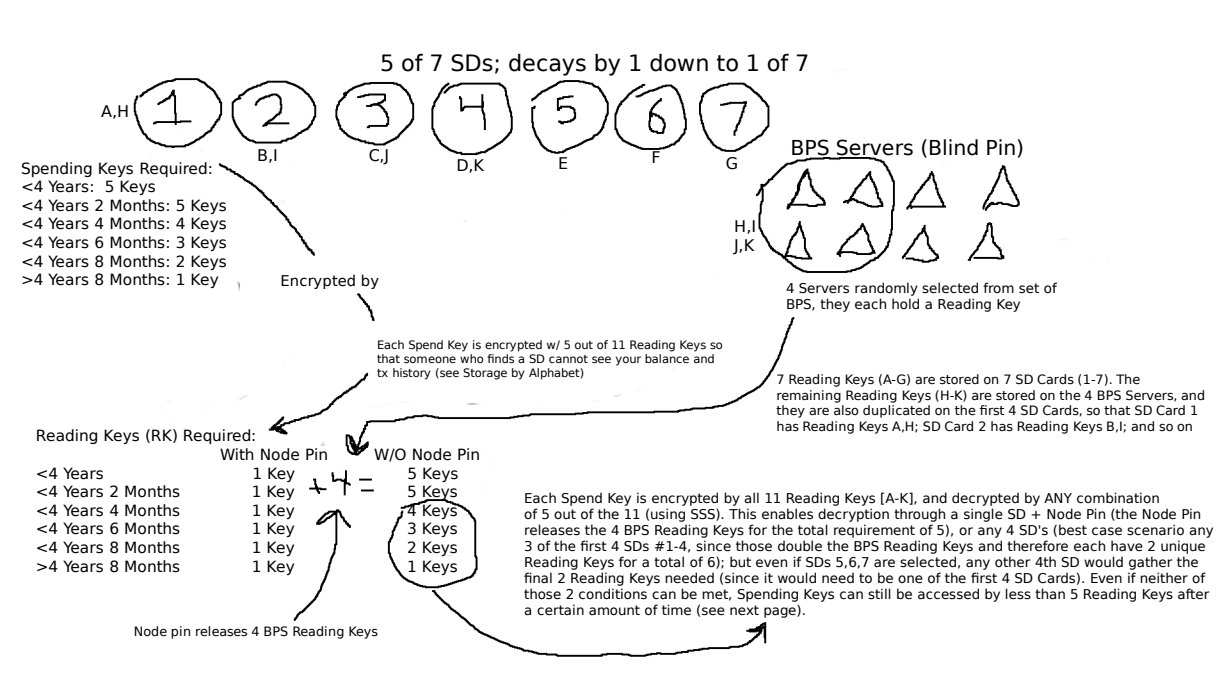
Take Coordinator to each Signer, not keys

QR Code Confirm Laptop

Laptops Yellow, Green, and Blue are Offline Laptops used to sign for the remaining 3 SD cards, for the total requirement of 5 out of 7. The exact assignment of signers isn't actually important, just that when you go to sign 5, only max 2 SD cards are signed on the online laptop (no more than Attic and Bank Safe are signed on the Online Node, Red), and then the remaining 3 SD cards each have assigned a unique Offline Laptop



5 Laptops generate the 7 keys (5 because RED generates 2), which are then stored on SD cards, which also holds the OS and Bitcoin Core as a bootable SD. A backup CD of private data is stored with it. After generation, the laptops are destroyed. Think of the SD card as a mini laptop, since the Laptop can't be stored with the CDs due to size and Evil Maid Risk, and the bootable SD Card makes loading and signing easier later. When you need to spend, repurchase new Laptops. After Signing on RED, the Coordinator is taken from location to location, not the keys. To transfer PSBTs, the SD Card is loaded into 1 of the 3 Offline Signing Laptops (Yellow, Green, or Blue), tx is signed, and QR codes are used to transfer to Coordinator. Repeat for the other 2 Offline Signing Laptops. Confirm QR codes first by scanning on a separate device, confirming the tx data is correct, and ensuring the QR code does not change before loading it on the Coordinator. Then broadcast on Online Node


5 of 7 SDs; decays by 1 down to 1 of 7
A,H
1
2
B,I
3
C,J
4
D,K
5
E
6
F
7
G
BPS Servers (Blind Pin)
H,I
J,K
Spending Keys Required:
<4 Years: 5 Keys
<4 Years 2 Months: 5 Keys
<4 Years 4 Months: 4 Keys
<4 Years 6 Months: 3 Keys
<4 Years 8 Months: 2 Keys
>4 Years 8 Months: 1 Key
Encrypted by
4 Servers randomly selected from set of BPS, they each hold a Reading Key
Each Spend Key is encrypted w/ 5 out of 11 Reading Keys so that someone who finds a SD cannot see your balance and tx history (see Storage by Alphabet)
7 Reading Keys (A-G) are stored on 7 SD Cards (1-7). The remaining Reading Keys (H-K) are stored on the 4 BPS Servers, and they are also duplicated on the first 4 SD Cards, so that SD Card 1 has Reading Keys A,H; SD Card 2 has Reading Keys B,I; and so on
Reading Keys (RK) Required:
With Node Pin
W/O Node Pin
<4 Years 1 Key 5 Keys
<4 Years 2 Months 1 Key +4 = 5 Keys
<4 Years 4 Months 1 Key 4 Keys
<4 Years 6 Months 1 Key 3 Keys
<4 Years 8 Months 1 Key 2 Keys
>4 Years 8 Months 1 Key 1 Keys
Node pin releases 4 BPS Reading Keys
Each Spend Key is encrypted by all 11 Reading Keys [A-K], and decrypted by ANY combination of 5 out of the 11 (using SSS). This enables decryption through a single SD + Node Pin (the Node Pin releases the 4 BPS Reading Keys for the total requirement of 5), or any 4 SD's (best case scenario any 3 of the first 4 SDs #1-4, since those double the BPS Reading Keys and therefore each have 2 unique Reading Keys for a total of 6); but even if SDs 5,6,7 are selected, any other 4th SD would gather the final 2 Reading Keys needed (since it would need to be one of the first 4 SD Cards). Even if neither of those 2 conditions can be met, Spending Keys can still be accessed by less than 5 Reading Keys after a certain amount of time (see next page).

If a user forgets their Node Pin, and has lost 3 SDs (so doesn't have access to 5 Reading Keys) - all is not lost

You'll remember this chart from the previous page:

Reading Keys (RK) Required:

	With Node Pin	W/O Node Pin
<4 Years	1 Key	5 Keys
<4 Years 2 Months	1 Key	5 Keys
<4 Years 4 Months	1 Key	4 Keys
<4 Years 6 Months	1 Key	3 Keys
<4 Years 8 Months	1 Key	2 Keys
>4 Years 8 Months	1 Key	1 Keys

How do we decrypt Spending Keys with <5 Reading Keys and no Node Pin?

Reading Key Publishing Schedule:

Time	Number of Keys
<4 Years	0
<4 Years 2 Months	0
<4 Years 4 Months	1 Key
<4 Years 6 Months	2 Keys
<4 Years 8 Months	3 Keys
>4 Years 8 Months	4 Keys

You can see that the Reading Keys are published concurrently with the decrease in required Reading Keys (above) and required Spend Keys (previous page).

Example: You only have SD's #5,6,7
You have lost all others and Node Pin

According to the Spending Keys schedule, you can spend with 3 keys only after 4 Years and 6 Months. So you wait that long, and go to spend. But each Spend Key is encryped with 5 Reading Keys. As you only have 3 SDs 5,6,7, you only have 3 Reading Keys E,F,G - what do you do? Fortunately, as it has been more than 4 Years and 6 Months, 2 Reading Keys are published by the BPS. They are published to the blockchain, so your online node picks them up automatically. They are useless to everyone else. With 5 Reading Keys now, you can decrypt all 3 Spend Keys, and spend you bitcoin.

Why would BPS Operators help me?

This is because there is a very strong financial incentive for them - on setup, you gave them pre-signed transactions that were only valid after their respective time locks, which upon progragation pays them and publishes your Reading Key stored with them - only both or neither can happen.

All laptops are single purpose

For Laptops:
Online Node, Coordinator, QR Confirm

Run Ubuntu

For SDs:
SD #1-7

There is an encrypted partition and cleartext partition on every SD. Encrypted Partition is encrypted with 11 Reading Keys and decrypted with ANY 5 Reading Keys (using SSS). Remember, each SD has a distinct Reading Key (SDs #1-7 have Reading Keys #A-G), and BPS #1-4 have Reading Keys #H-K. Then those BPS Reading Keys are duplicated onto the SDs #1-4, so that SD #1 has Reading Keys A,H; SD #2 has Reading Keys B,I; SD #3 has Reading Keys C,J; SD #4 has Reading Keys D,K

(This is not a double encryption to what has already been discussed, this is just how it is implemented)

BPS #1-4

Each contain 1 Reading Key (H-K). They also contain the Pre-signed transactions that simultaneously pay the BPS Operators (themselves) and publish their Reading Keys to the blockchain, so that less Reading Keys are required to decrypt the SDs and the Spending Requirement can decay.

Online Node:

* Syncing Blockchain
* Watching all 7 Tripwire Accounts - alerts User when stolen

SDs 1-4:

Encrypted

*Unique Spend Key
*Copy of the Descriptor

Cleartext

*Unique Reading Key
*One copy each of the Unique BPS Reading Keys

Only difference is that SDs 1-4 have a copy of the BPS Reading Keys

SDs 5-7:

Encrypted

*Unique Spend Key
*Copy of the Descriptor

Cleartext

*Unique Reading Key

Tripwire:

Each CD/DVD contains a single-sig key with 0.05 btc, to encourage any adversary who finds one of the 7 keys, to steal this bitcoin immediately, instead of searching to find more CD's & the entire fund. The Node watches these tripwire accounts, and if the bitcoin is moved, the user is notified to immediately move the funds to 7 new CD's and locations.

Protocol Walkthrough:

The Online Node is always syncing to the blockchain. For Key Generation, this Online Node (Red), generates 2 Keys (the Attic Key, and the Bank Safe Key) - more details on Key names and Locations on the next page. You have 5 Offline Laptops that generate the remaining 5 Keys, for a total of 7. These Keys are all saved on their respective SD Cards, with Ubuntu OS and Bitcoin Core on each as well. The Private Key data is also backed up on respective CDs. The 7 CDs and 7 SD Cards are packed into 7 packets, and distributed into 7 geographically distributed locations. The SD Cards are bootable, so in the future it is simple to sign transactions in a secure manner. The Laptops are destroyed (except the online always syncing laptop.)

When you need to spend in 4 years or more, buy 3 Offline Laptops. First, gather your Bank Safe SD Card/CD Packet and your Attic SD Card/-CD Packet. Input your Bank Safe SD Card (Attic could be first, it doesn't matter) into the Online Node and boot into it. To unlock the encrypted partition, enter your Node Pin to retrieve the BPS Server's 4 Reading Keys. Those Keys, plus the Reading Key on the SD Card itself, will unlock the SD Card. You can now create the spending request (ex: send $10,000 to Mom), and make the first signature on it with the Spending Key on the SD Card. Remove the SD Card, boot into the Attic SD Card, and make the 2nd signature on the transaction with it.

Now use a QR code to send it to the Coordinator laptop. But first, check the QR Code on the 'QR Code Confirm Laptop'. Scan it on this Laptop, and 1) Make sure that the QR Code scans as "$10,000 to Mom", and not "$100,000 to M0M". 2) Then, check that the QR Code does not visually change after scanning, before scanning it on the Offline Laptop you are actually trying to get the data to. DO THIS EVERY TIME THE GUIDE SAYS USE A QR CODE.

So, use a QR Code to send the twice-signed transaction to the Coordinator Laptop. Buy a new Laptop, as Offline Laptop Signer 1. You are now going to take the Coordinator Laptop and Offline Laptop Signer 1 to SD Card 3 (Remember, you have 7 locations, and you need to choose 5. I'm not saying this is location 3, but that this is your third pick). Boot SD Card 3 into Offline Laptop Signer 1. Transfer the QR Code from the Coordinator to the Offline Laptop Signer 1. Sign the transaction for the 3rd time (if correct). Transfer the transaction back to the Coordinator via QR Code. Destroy the Offline Laptop Signer 1. Now repeat this paragraph two more times for (Offline Laptop Signer 2 and 3), (SD Card 4 and 5), (Signatures 4 and 5), and Destroying the laptops each time. At the end, you should have a fully signed transaction on the Coordinator.

Now, transfer this fully signed (5 times) transaction to the Online Node via QR Code, and broadcast it to the Bitcoin blockchain. You should be able to do this immediately, since it is always syncing.

Hard Engineering Tradeoffs

*We don't know how to combine this with on-chain privacy tools like coinjoin without increasing risk.
*You can't make a location very secret and easy to access.
*You can't make a location difficult for an Adversary to find and easy for an Heir to discover.

To solve the issues with locations, we have come up with the following suggested trade-offs:

Nickname	Location	Secrecy	Inheritance	Access	Location
Caves	SD 1-4	Very Good	Very Bad	Very Bad	100km away from home and not together
Aunt Jane	SD 5	OK	Very Good	OK	100km away from home and 20km from 1-4
Attic	SD 6	Good	Bad	Good	Near Home
Bank Safe	SD 7	Very Bad	Very Good	Very Good	Near Home

SD 1-4:	Very Secret:	Ex. Buried in secret locations, hidden in foundations of a building, use steganography, etc
SD 5:	Very Inheritable:	Ex. Place in envelope labeled 'Last Will & Testament' and give to Attorney and Friends.
SD 6:	Secret and Accessible:	Ex. Hidden in an attic wall, underneath floorboards, inside upholstery, etc
SD 7:	Very Inheritable & Very Accessible:	Ex. Home safe, Bank safety deposit

Remember, SD 6 and 7 (Attic and Bank Safe), start the spending request by signing on the Online Node; everything else is Booted/Signed Offline!