SHIELDED ORDINALS · How it works, illustrated

SHORD

Private Holdings, Public Proofs

A Bitcoin-native privacy protocol



00 Start here

What SHORD is, and what you actually hold

> SHORD is a privacy protocol on Bitcoin: the art stays public, who holds which piece does not. What you hold is a note recording which piece and whose it is; the chain keeps only its fingerprint, the readable note and the keys stay in your own vault, and only your key can spend it.

IN THE PLATE

1. Your vault, on your device: keys, readable notes, proofs.
2. The envelope: every action rides in a Bitcoin transaction; for the art, in its OP_RETURN (the sign).
3. The blocks: Bitcoin's record, in order, open to all.
4. The indexer: checks envelopes by SHORD rules and keeps the ledger; 9F3A on its screen is the ledger's digest (chapter 08).

An ordinary Ordinal belongs to whoever holds the address it sits in, for anyone to see. SHORD parks the art at a NUMS address nobody can spend from and keeps who holds it in notes; in the project's words, "The art of every piece is public. Who holds it is not." Bitcoin checks only the BTC transaction and indexers work out the SHORD ledger by its rules, so a transaction Bitcoin has confirmed can still be rejected by SHORD. The examples in this guide are Shielded Nakas (NAKA) and SHNAKAS, the rune mined by holding one.

Public on chain	Not visible on chain
The art and where it sits	Which piece is whose
A SHORD transaction, its nullifier and ciphertext	Which note was spent, and to whom
A listing's number, price, payout	The keys and notes in a vault

Warning The signature your wallet makes to open the web vault yields every key of the vault. Never give it to anyone (chapter 01).



Card 00 One private send: where each thing comes from and lives

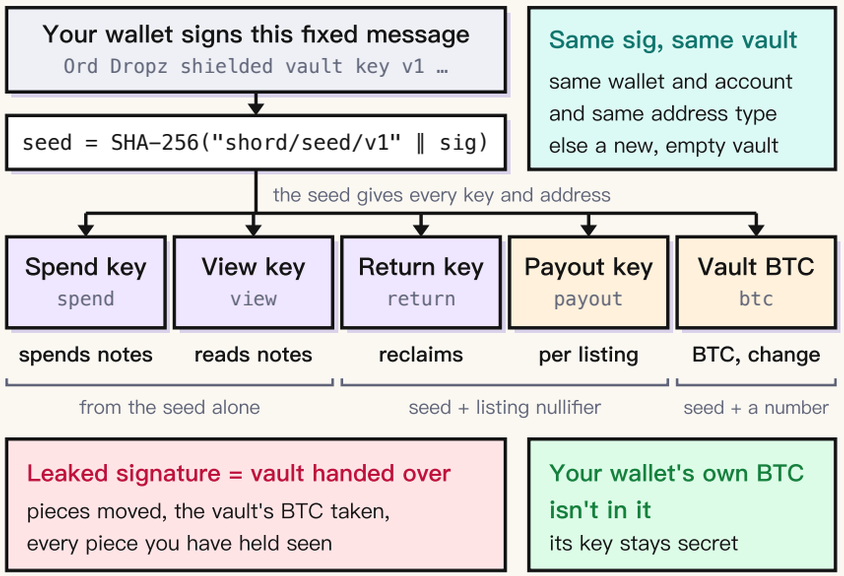
Thing · use	Vaults	Bitcoin blocks	Indexer
Keys spend, read	made from the seed, stay here	never on chain	never touches them
Readable note which piece, whose	sender makes it, receiver opens it	never on chain	can't read it
Commitment cm public fingerprint	the sender computes it	in the envelope	adds it to the tree
Ciphertext ct receiver reads it	to the receiver's view key	in the envelope	keeps it for scans
Nullifier nf no double spend	computed to spend the old note	in the envelope	lists it as spent
Proof π the send is valid	the sender makes it on device	in the envelope	checked, not kept in the ledger

Indexer column: all from the blocks; same chain, same rules, same ledger

TECHNICAL NOTES The cards are for checking; skip them and the story still reads. This one follows a private send; in a mint, purchase or claim the receiver makes the new note, the indexer completes its commitment, and no proof is carried. Where the proof parameters come from is in appendix D.



Card 01 One signature, every key



TECHNICAL NOTES On the card, ‖ joins two pieces end to end. Every key is SHA-256(a fixed label ‖ seed …), with labels such as shord/spend and shord/view. The phone's 24 words are this 32-byte seed plus a checksum byte. The shielded address shord1… is the owner key Poseidon(sk) followed by the view key; neither reveals a private key, so it can be given out.

01 Vault

What holding rests on, where the keys come from

Holding means having the keys: whichever note your vault's keys can spend, that piece is yours. A web vault's keys all come from one signature by your wallet, so that signature amounts to the whole vault.

IN THE PLATE

1. The official site: one signed message opens your vault.
2. The signed sheet: the signature becomes the vault's seed.
3. Its shadow is a bunch of keys: every key comes from the seed.
4. The copy, off by the red 0: sign there and it has the keys.
5. The phone: the seed as 24 words (not a wallet's seed phrase) opens the same vault in the Shielded Vault.

The keys need no backup. Sign again on any device with the same wallet, account and address type and the same vault opens; anything else opens a new, empty one. The notes are in envelopes on chain and the vault finds them by scanning with its keys; for now only the site's indexer lists them all, and the project says the new version will be open source, so your own indexer could too (appendix H).

The shielded address shord1… also comes from the seed and is fine to give out: the site says it is “only needed when someone sends you a piece directly”, and buying and selling don't use it.

The signing box says “Anyone with this signature can move your shielded pieces.” The signature computes the whole vault, which holds not only the pieces but the vault's BTC and the notes you have received.

Myth Signing a message without sending money can't cost me anything.

Fact **This signature is every key to your vault. Sign it only on the official site, and give it to no one.**

02 Issuing

Rules in the parent, and who gets the mint money

> **One parent inscription holds a collection's supply, prices and payout addresses. Mints pay by these rules, the indexer checks them, and what doesn't match doesn't count.**

IN THE PLATE

1. The tablet: the parent, carved with NAKA's supply and prices.
2. The mint: A pays, note in hand, and spends a permit coin.
3. The split: the artist 15,000; the relay its fee and 3 credits.
4. After: a later tx inscribes the piece as a child, parked at NUMS.

The indexer checks three things in a mint. The number is within supply and not yet minted. Each payout script gets at least what the rules ask; paying more is fine. The tx spends a permit coin, and NAKA's permit coins only the platform can sign. With all three in place the indexer accepts the mint. The minter's vault builds the new note and the indexer completes its commitment.

With the site's default a mint comes with 3 relay credits and costs 25,500 sats, plus the miner fee; the card shows who gets each part. Each credit pays the official relay for one private send, listing or cancel (chapter 04).

NAKA's art isn't on chain up front: the parent holds no pictures, only a hash tree root over every file, and each picture is inscribed after its mint. The parent goes to NUMS after the last child, and then nobody can add one; on 28 October it wasn't there yet (appendix G). The SHNAKAS rune has its own etching and isn't in this parent (How mining works).

Myth The rules are a website setting the project can change at will.

Fact **They're carved in the parent inscription for good, and no website setting can change them; the indexer reads only that parent on chain.**



Card 02 **Rules in the parent, payments checked against them**

Parent inscription · the rules on chain (NAKA, in sats)			
Pieces	Supply 3,333	File tree root artRoot	
Mint	Mint price 15,000	Inscribe fee 3,000	Platform fee 0
Relay, sale	Relay credit 2,500 each	Sale fee 2%, at least 1,000	
Keys	Mint permit key gateKey	Listing lock key lockKey	
Payouts	Artist, platform and relay scripts; NAKA: artist = platform		

↓ The indexer reads the rules from the parent's transaction, not a website

Minting one piece (MINT)

15,000 → artist
3,000 inscription fee → relay
Default 3 × 2,500 → relay
Platform mint fee 0
Total 25,500, miner fee extra
The tx spends one permit coin

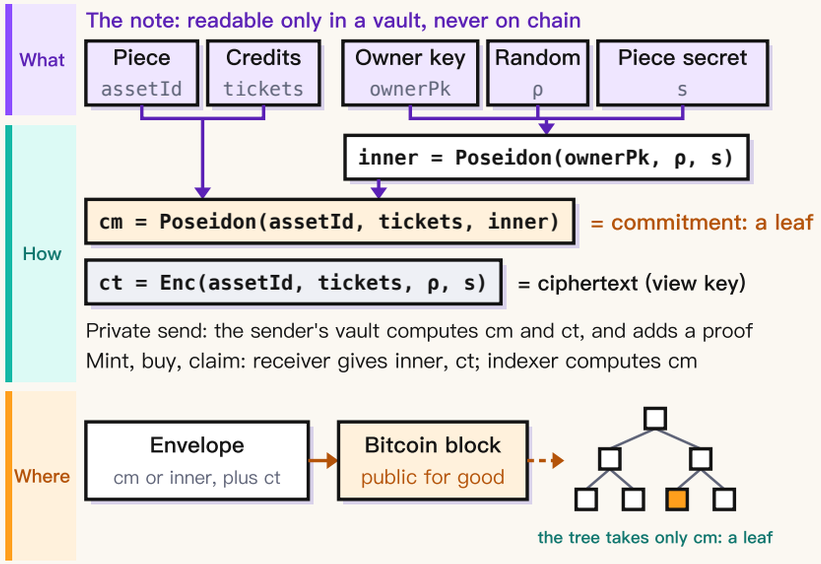
Selling one piece (BUY)

Price → seller
2%, at least 1,000 → platform
Paid by the buyer on top
e.g. 100,000: buyer pays 102,000
No royalty on NAKA
The tx spends the listing lock

TECHNICAL NOTES The collection id in each envelope header is the first 8 bytes of the SHA-256 of the parent's inscription id, and the indexer takes only envelopes carrying it. A permit coin is a Bitcoin output locked to the platform's permit key; each mint transaction spends one, and a batch of up to 20 pieces also spends just one.



Card 03 One note; on chain, only its commitment and ciphertext



TECHNICAL NOTES The record tree is 20 levels deep, holds up to 1,048,576 notes and only grows: each new note adds a leaf, and a spent note's leaf stays. A spend's proof rests on a tree root from the last 100 blocks and shows only that the note is in the tree, not which leaf; whether it was already spent, the indexer checks against the nullifier list (chapter 04).

03 Storage

Where the art is, and what you bought

What you buy isn't the picture but a note. The picture, once inscribed, sits at the NUMS address where nobody can move it; a change of hands changes the note, never the picture.

IN THE PLATE

1. The art in the NUMS gallery: anyone sees it, nobody takes it.
2. The keyless lock: this address holds all 3,327 NAKA pieces; no one knows its key, and it has no script (appendix G).
3. A's note says #4: it points at that piece, which stays in the gallery.
4. CM on top of the blocks: the note's commitment.
5. SHNAKAS: the 21 million runes sit at this address too.

The note records which piece, how many relayed actions are left (chapter 04) and whose it is (the owner's public key), plus two random values, ρ and the piece secret s, unrelated to the image.

The note itself never goes on chain. Its fingerprint does, the commitment, which the random values keep from showing the piece or the owner; indexers collect every commitment into a record tree, a public list of every note there has been. A fingerprint can't be turned back into a note, so each comes with a ciphertext locked to the holder's view key; the readable note stays in the vaults that make and receive it.

A mint, purchase or claim shows the number anyway; a private send doesn't, and afterwards outsiders can't tell which piece the new commitment is or whose.

SHNAKAS works the same way: a pouch is a note numbered by pouch, its amount public against that number, so anyone sees what it holds but not whose (How mining works).

Myth When I buy a NAKA, the image moves into my wallet.

Fact **The art stays at the NUMS address. An ordinary wallet won't show it; your holdings are in the SHORD vault.**

04 Transfer

A private send, and what the chain shows

> **Anyone can check that a private send is valid and its note not spent twice, but not which note was spent or who got it.**

IN THE PLATE

1. The old note, crossed out: void once the send is accepted.
2. The envelope: in the newest block, the car with the letter.
3. The spent list: first digits, red is new; a repeat is refused.
4. B's key: the view key opens the ciphertext; a new note appears.

A sends #4 to B's shielded address (shord1…) or name (bob.naka, appendix J). A's vault computes the old note's nullifier, writes B a new note encrypted to B's view key, and proves the old note is in the tree and A's and the new one is the same piece, never saying which. A pays the miner fee to send it, or the official relay sends it for one relay credit or the site's prepaid relays (chapter 08, appendix I04).

Only once the proof checks out and the nullifier is new does the indexer record it and add the new commitment to its tree, so a second spend is refused; and since no one can match a nullifier to a leaf, no one knows which note was spent. B's vault opens the ciphertext; once its commitment matches the tree's, the note is B's. The ciphertext is written to B's view key: A's vault writes it by the rules, and B's vault opens it at the next scan (appendix H).

A whole SHNAKAS pouch is sent just like a piece. Giving part takes two steps: split it into 2 to 16 pouches, then send one. A split shows the pouch number and new sizes, not whose they are (appendix I07).

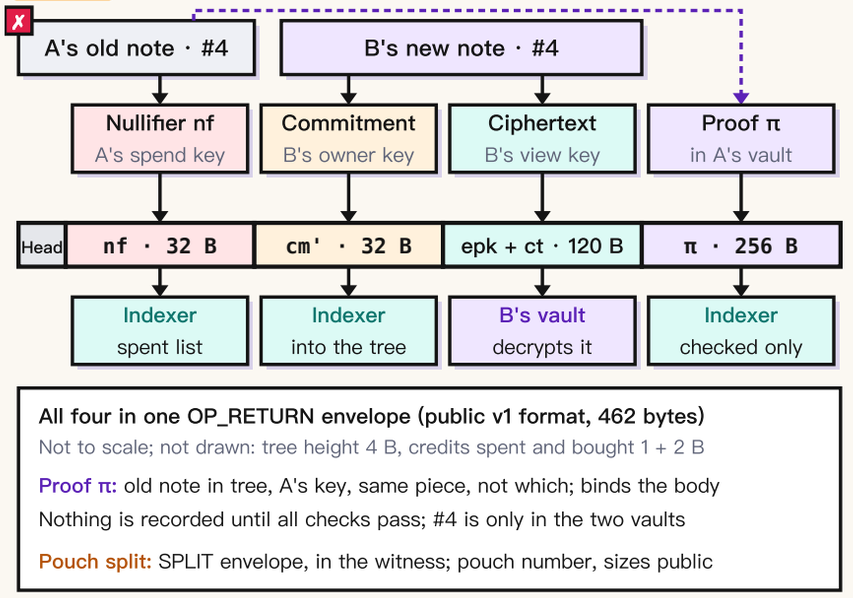
A private send collects no money, so in a trade whoever goes first trusts the other; to swap both at once, use a listing (chapter 05).

Myth A private send hides everything, even that I sent it.

Fact **The chain shows a note was spent and, if you paid the fee yourself, which wallet sent it.**



Card 04 **A private send: one envelope, each part to its place**



All four in one OP_RETURN envelope (public v1 format, 462 bytes)

Not to scale; not drawn: tree height 4 B, credits spent and bought 1 + 2 B

Proof π: old note in tree, A's key, same piece, not which; binds the body

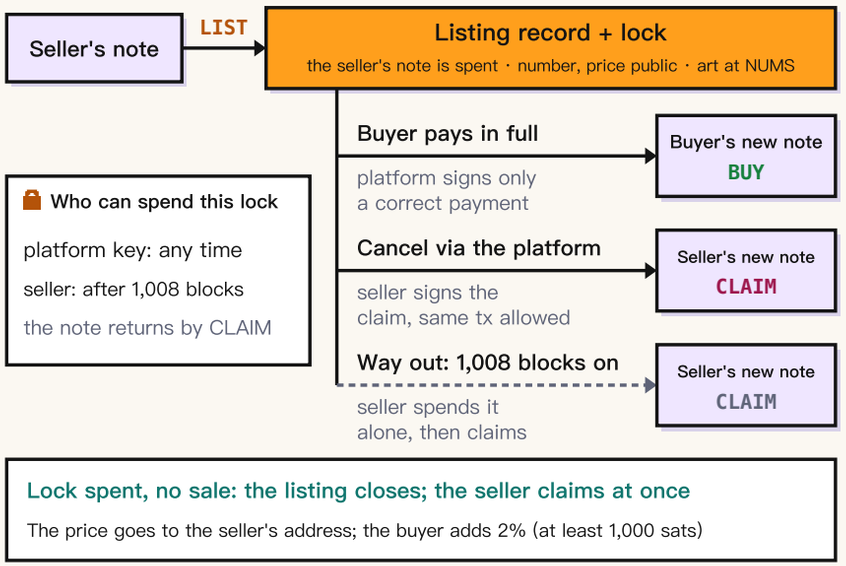
Nothing is recorded until all checks pass; #4 is only in the two vaults

Pouch split: SPLIT envelope, in the witness; pouch number, sizes public

TECHNICAL NOTES The indexer never marks the old leaf spent, because that would tie this spend to the transaction that made the leaf. It records a nullifier instead, worked out from ρ and the owner's spend key. Leave out the key and private sends give themselves away: the new note's ρ comes from the old nullifier, which is public, so anyone could compute the new note's nullifier in advance and, note after note, chain every move of the piece.



Card 05 One lock, three ways out



TECHNICAL NOTES The lock is a P2TR output with two ways to spend it: the platform's key by the key path, at any time, and the seller's return key by the script path, once the lock is 1,008 blocks old. Spending it gets the platform only the sats inside (330 in every lock the official relay builds; the protocol sets no amount). The piece comes back only with a claim signed by the return key, which comes from the seller's seed and this listing's nullifier, so the platform can't derive it.

05 Trading

What a listing reveals, and the platform's role

A NAKA listing shows number, price and payout; a lock decides where the piece goes. The platform holds the key: it can close the listing, not take the piece or the money.

IN THE PLATE

1. The price tag: number and price in a block, for all to see.
2. The lock: a 330-sat output, its key on the platform's board.
3. The buyer: price to the seller, plus 2% (at least 1,000 sats).
4. The hourglass: the seller's way out after 1,008 blocks.
5. CLOSED: lock spent, no sale; the seller must claim it.

A listing (LIST) gives up the seller's nullifier and makes the lock in the same transaction. Once it's accepted the piece has no spendable note; how the lock is spent decides the next one (next page).

The payout address can be a vault address, the wallet or one typed in (bc1p, bc1q and 3... have all sold); the price must clear its dust limit.

A purchase (BUY) spends the lock and pays in one transaction, and the buyer makes the new note. The platform signs the lock only for a correct payment, the protocol document says: at 1,000,000 sats the seller must get at least that and the platform at least 20,000, or the indexer rejects the purchase.

A cancel (CLAIM): the platform spends the lock and the seller signs the reclaim with the return key, in one transaction if they like. Without the platform, the seller can spend the lock alone at 1,008 blocks. A lock spent with no sale closes the listing, and the seller can claim it straight away, with no 1,008-block wait.

Myth A confirmed payment means a completed purchase.

Fact **The payment goes into a block, then the indexer checks it against the listing; it's a sale once accepted and the new note is in your vault.**

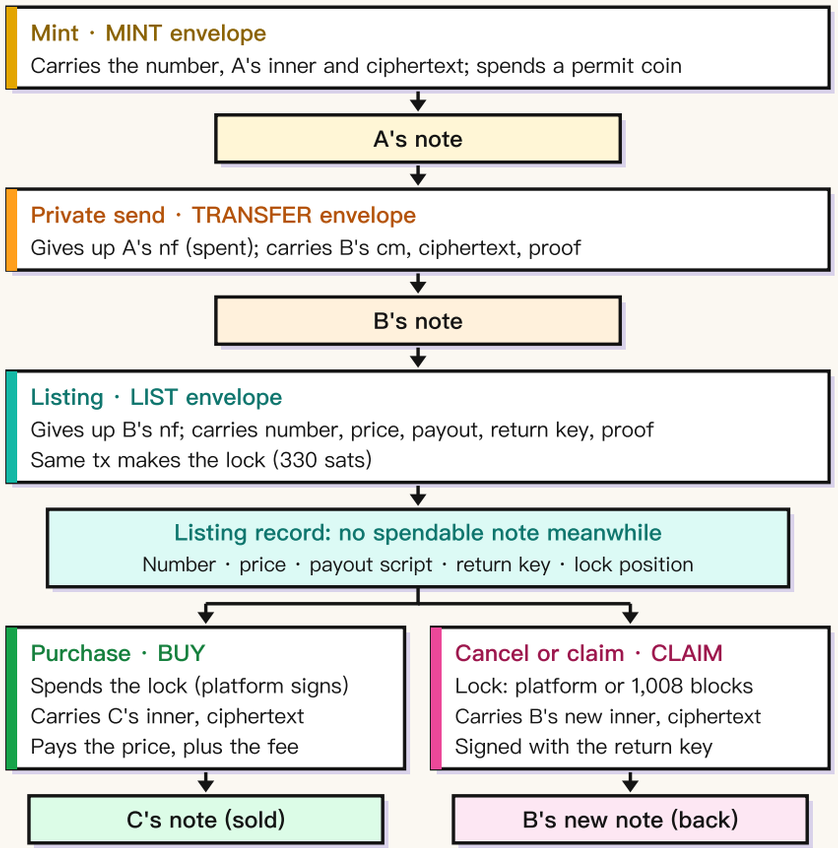
Ownership flow

Transactions and envelopes

> **The last page showed a listing's three ways out; this one walks #4 from mint to sale, then shows why a mempool envelope can't be changed and who can stop a purchase before it's mined.**

This is NAKA's on-chain listing. Others now list off chain until the sale; on 27 October the project said Nakas won't change.

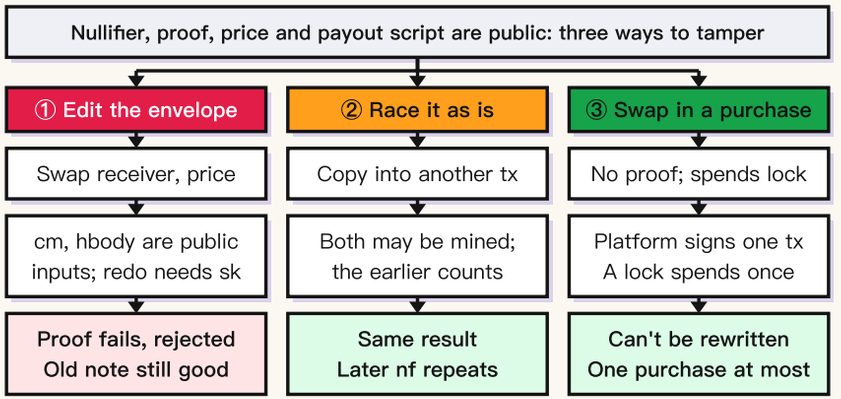
Flow 01 #4 from mint to sale: each transaction and envelope



Lock spent without a valid purchase: the listing closes, the seller claims later

The listing id is the nf given up at listing; the lock records no ownership

Flow 02 An envelope in the mempool: readable, not changeable



Redo a proof: spend key. Sign a claim: return key. Both stay with the holder

Sends, listings, claims can move as is; buy and mint signatures fit one tx

Before a purchase is mined, who can stop it

Who	How	Result
Buyer	Before it confirms, spends the same input again in a higher-fee transaction (RBF)	If that confirms first, the purchase can't be mined and the listing stays; the buyer pays only the new fee
Platform	Spends the listing lock first with its key	If that confirms first, the purchase can't be mined and costs the buyer nothing; the listing closes and the seller claims
Seller	Can't before the lock is 1,008 blocks old (a cancel needs the platform); after that, can spend the lock and race the purchase	Whichever confirms first: the seller's closes the listing and reclaims; the purchase's sells
Anyone else	Can't change your envelope (chart above); racing for the same listing takes another purchase the platform signed	A lock spends once: at most one purchase succeeds
A miner	Leaves the purchase out	Other miners may still mine it

06 Proving

Holding proofs, and what each one gives away

The vault makes the proof; note and keys stay home. Uses differ in where it goes and whether the note moves: votes go to the vote service, Prove to the site, the note untouched; a base reward claim goes on chain and renews the note.

IN THE PLATE

1. The proof card: all #46 hands over; the note stays in the safe.
2. The checklist: SNAP (in the snapshot), VALID (proof holds).
3. The ballot box: choice and proof go to the vote service, off chain.
4. The tag board: the vote service's public log, a row per vote.
5. The listing: #4 listed later; its nullifier matches the red row.

Votes follow a snapshot: each NAKA held at its block gets one, even if sold later. The proof uses the spend circuit but spends nothing; each published vote carries a tag made from the nullifier, so anyone can check the tally. The tag matches nothing until the note is spent; a listing then shows the number too.

Snickets can vote too, no NAKA needed: no proof, the vault just signs for the platform, and Snickets voted don't come back (appendix I06).

A base reward claim goes on chain: the envelope turns the old note into a new one plus a SHNAKAS pouch, nullifier and commitment public, piece hidden. The bonus is a separate signed envelope showing the number; late and batch claims in appendix I06.

The Prove button runs another circuit, the holder proof: it goes to the site's verifier, your wallet as claimant, showing only a root, an event id and a de-duplication mark, not even the nullifier. It shows the note is in the tree, not that it is unspent: a sold note's leaf stays, so the newest root still proves it.

Myth Voting or pressing Prove spends my NAKA.

Fact **Neither touches the note or the chain; a base claim renews it, and mining carries on (chapter 07).**



Card 06 **Three uses, and what each hands over**

Site vote spend circuit	Proves	An eligible note held at the snapshot
	Hands over	Choice and proof, nullifier inside
	The note	Not spent, not on chain
	Edge	**Once spent, its public tag shows the vote**
Base reward spend circuit	Proves	A week–long note is in the eligibility tree
	Hands over	On chain: nullifier, new commitment, proof
	The note	Normal claim: spent, replaced by a new one
	Edge	**Number hidden, nullifier public**
Holder proof holder circuit	Proves	I know a tree note's data and its key
	Hands over	Proof and claimant, via the Prove button
	The note	Not spent, not on chain
	Edge	**Doesn't check it unspent: old leaves stay**

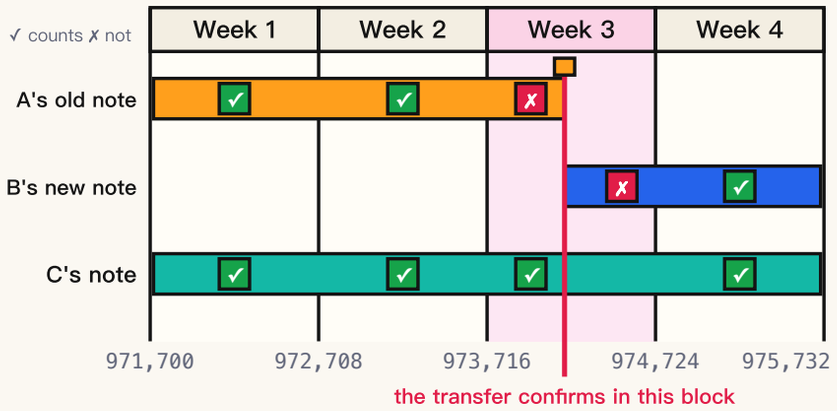
The bonus is a separate BONUS envelope: a signature, no proof, number public

Tally and tags match the public log; proofs aren't public, eligibility only sampled

TECHNICAL NOTES The holder proof's de-duplication mark is Poseidon(s, context, 13). The context comes from the collection, purpose and claimant, so a piece proves once per event, and with the claimant inside, a copied proof is no use to anyone else. A vote's tag is a different thing, made from the nullifier, topic and round: it is where the match above comes from.



Card 07 **A week is 1,008 blocks**



A week is yours if you held from its start until after its end
Except: a new note from a reward claim keeps the week

1,008 blocks, 266,640 a week, 78 weeks; week 3 is blocks 973,717—974,724

TECHNICAL NOTES Eligibility is just two block heights: the block that created the note is no later than the week's start, and the block that spent it is after its end (or it's unspent). Weights follow the listings every block, so a trait floor's change moves the bonus. The weeks' block ranges are in appendix E.

07 Mining

Holding a NAKA mines SHNAKAS; why a transfer breaks the week

SHNAKAS pays by whole weeks of 1,008 blocks: a note held from a week's start to its end earns that week's share. Shares wait on the books until a claim makes them pouches.

IN THE PLATE

1. A bell post every 1,008 blocks; between two of them is week 3.
2. The transfer: A's letter lands mid-week, where B's note starts.
3. The crossed-out coins: week 3's share, for neither A nor B.
4. C and the chest: held all week; a claim collects the share.

The plate shows chapter 04's transfer, confirmed mid-week 3: A's old note is spent in that block, short of the week's end; B's new note starts there, not at its start; so week 3 counts for neither. Earlier weeks stand: A keeps weeks 1 and 2, and B counts from week 4.

Listing, cancelling and sending to yourself (adding relay credits is one) spend the old note and break the week too. A base claim spends it too, but the same transaction returns the piece to your vault and the indexer adds the new note to that week's tree, so mining goes on. Bonuses, repricing, votes and holder proofs don't touch it.

Eligibility says whether a week counts, the weight how much: 100% to start, extra for a high trait floor or a low number, up to 400%. The rules are on chain, so anyone gets the same figures (appendix I07); the Naka Sigil extension shows each piece's week. Listed and unminted pieces hold shares nobody can claim.

Shares can pile up unclaimed; the base needs no number, the bonus needs it shown. Rewards run to block 1,050,324 and claims stay open 52 weeks more, to 1,102,740; the next page follows a claim.

Myth List or sell, and your full weeks are gone too.

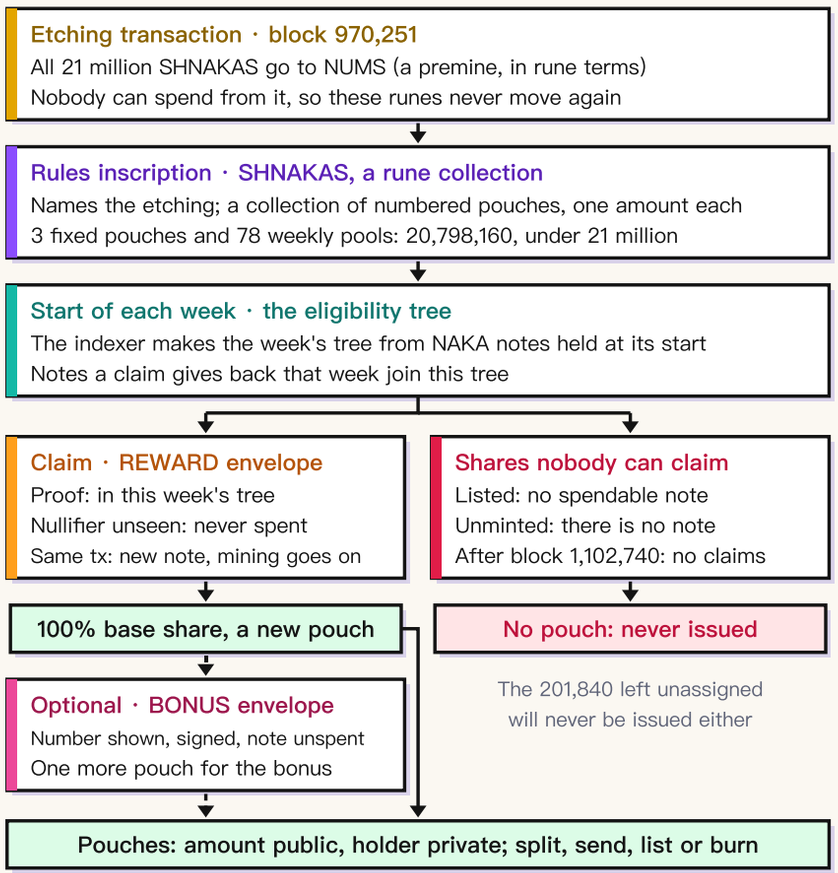
Fact **Full weeks stay claimable, even after a sale; listed weeks earn nothing, and earning resumes after the reclaim.**

How mining works

From etching to pouches

The shares saved up on the last page become pouches only when claimed. In SHORD, SHNAKAS is a collection of numbered pouches, each one exact amount, not pieces; the 21 million runes went to the NUMS address at the etching and never move, and pouches don't turn back into them. Flow chart 03 follows one claim.

Flow chart 03 — One claim: how a share becomes a pouch



Pouches don't turn back into the runes at NUMS, which stay put either way

A normal claim; late claims and several weeks at once: appendix I07

How the totals add up

The flow chart follows one claim; all claims together, set beside the ceiling and the rules, make this table. The ceiling was fixed at the etching, each week's pool is in the on-chain rules inscription, and the issued total adds up the pouches in the site's ledger.

Figure	Where it comes from	SHNAKAS
Ceiling	All of it went to the NUMS address at the etching; the site calls it the ceiling, “the most that can ever exist”	21,000,000
Most the rules issue	3 fixed pouches of 80, plus 78 weeks of 266,640; the rules assign the other 201,840 nowhere	20,798,160
Issued	By block 974,323: 987 pouches, 506 base (74,989.91) and 481 bonus (92,265.79); the 3 fixed pouches unminted	167,255.70
Never issued	Listed weeks' and unminted numbers' shares nobody can claim, and whatever is unclaimed at the deadline	not counted here
Burned	Pouches burned for Snickets go to an address nobody can spend from; the ledger can't tell which	the site's token page
Circulating	Issued minus burned, as the site's token page counts it	the site's token page

Why it's called the ceiling

At the etching all 21 million went to NUMS, which rune terms call a premine. But nobody can spend from NUMS, so this premine belongs to no one and only caps the total of pouches; this guide follows the site and calls it the ceiling.

Myth The shares nobody claimed were burned.

Fact **A burn is an issued pouch that can never be spent again, as when burned for Snickets. An unclaimed share never becomes a pouch, so the ledger never had it: “whatever is never claimed is never issued”, the site says.**

08 Verification

Who keeps the ledger, and can you check it?

> **The ledger is worked out from the Bitcoin blocks by an indexer under fixed rules, and the site's is no exception. To check it, replay the blocks with your own indexer and compare digests with the site's at the same block. For now the public code lags the site's rules, so the check is partial.**

IN THE PLATE

1. Blocks arrive in order, their heights on the cars.
2. Two indexers, the site's and one you run yourself.
3. Same rules, same block, same digest; 9F3A is its start.
4. The printed tape: each machine's own ledger and checkpoints.

An indexer reads block by block and records each SHORD envelope that passes the rules. No clock or server takes part, so the same rules and parameters give anyone the same result.

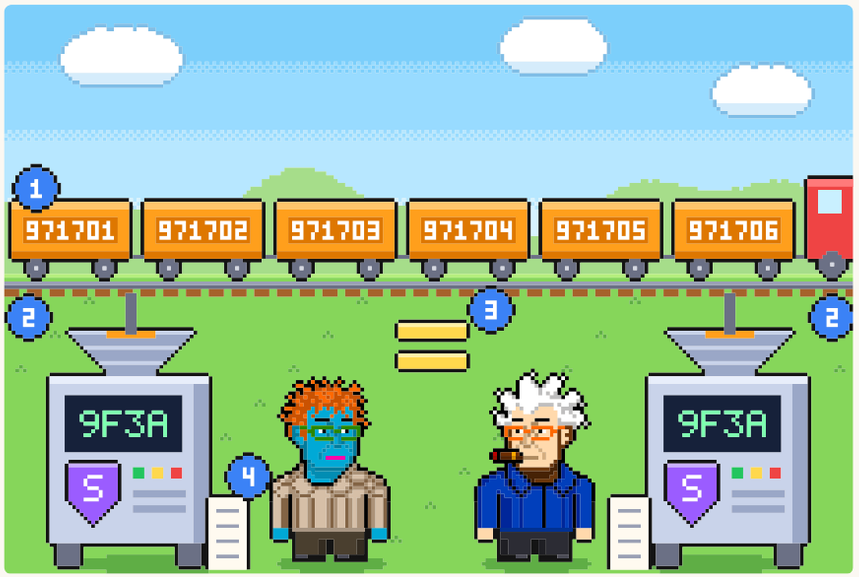
Each indexer keeps its own ledger, checkpointed every few blocks; a restart resumes from the newest good one, or from the start block.

Checking means comparing digests at the same block. On 11 October the public code's replay matched the site's at block 970,879. Today the site's rules are newer: the public code recognises only six message types, the later rewards, off-chain listings and splits aren't public yet, and its tree now differs from the site's (appendix H).

The relay's envelopes can be checked without an indexer: since 29 October it has sent from the wallet the site's status interface names (bc1pr5vsx…5hcp73), and a block explorer shows every one. Appendix I08 has the payout address in the rules.

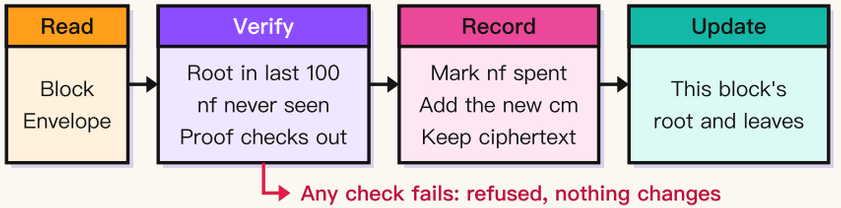
Myth The indexer holds the whole ledger, so it knows who holds each piece.

Fact **It holds only commitments, nullifiers and ciphertexts, so after a private send it can't tell a note's piece or owner. Only the recipient's vault can open the ciphertext.**



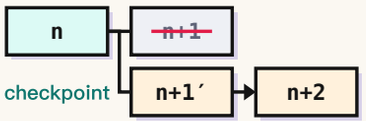
Card 08 What the indexer checks, what the digest covers

A private send: its proof is checked with public data, no private keys



On its machine: tree · nfs · ciphertexts · mints · listings · checkpoints

Reorg: back to a checkpoint



n+1′: a new block, same height

Digest v1 covers

root, leaves · nfs · minted

listings: state, piece, credits, price

Leaves out

ciphertexts · txids · locks

payout scripts

Compare full digests at the same block: same height, same hash

Given the same Bitcoin chain, rules, verification key and start height

TECHNICAL NOTES A checkpoint spares a restart the early blocks. It is used only if its rule hashes, the roots rebuilt from it and its last 12 blocks still check out; otherwise an older one is tried. After a reorg the indexer rereads from one still on chain, serving the old ledger meanwhile.

Quick answers

> **Ten things people often ask that the chapters don't spell out, from sending to the money, the pouches and the safety of your keys. The full list, by topic, is in appendix K.**

Q 01 How long does a private send take, and how does the other side know it arrived?

A Through the official relay, envelopes wait to go out in a batch, then the transaction confirms and the indexer accepts it. Paying the fee yourself skips the batch. Nothing notifies the receiver: the note has arrived when their vault syncs and shows it.

Q 02 I sent a private transfer to the wrong address. Can I undo it?

A No. Once the envelope is on chain, the old note is spent. A shielded address has no checksum, so a typo usually still makes a "valid" address, and the piece goes where nobody can open it. Paste the address, and check its start and end before sending.

Q 03 Can I send BTC or another inscription straight to the NUMS address to make it private?

A No. An ordinary transfer creates no SHORD note, and nobody has the key to what you send there, so it stays there.

Q 04 If I copy my note data, or open the vault on two devices, can I spend it twice?

A No. One note has one nullifier, and the indexer accepts only the first spend. A later one paid with other BTC can still confirm, and its miner fee is wasted.

Q 05 Where is the BTC after a sale?

A At the payout address you chose when listing: on the vault's Bitcoin page for "private address", to withdraw anywhere, or straight to the connected wallet or the address you typed.

Q 06 Can the SHNAKAS I claim go into an ordinary rune wallet or market?

A No. What you claim are pouches in the SHORD ledger; the 21 million runes went to the NUMS address at the etching, which nobody can spend from, and pouches don't turn back into runes. Appendix I07 has what pouches can do.

Q 07 The art is locked at NUMS, so what do I need to keep safe?

A What's locked is the Bitcoin output the art sits on; ownership lives in the SHORD ledger and follows your vault's keys, so those keys are what to keep safe (chapters 01, 03).

Q 08 My unlock signature leaked. Does signing again, or a new PIN, help?

A No. The same wallet and address give the same signature and keys. To change vaults, open a new one with another wallet address and send your notes there privately.

Q 09 What if my view key leaks?

A Whoever has it can open the ciphertexts written to you, past and future, and see which notes you received, but without the spend key can't move them.

Q 10 What is the trusted setup, and does it affect my piece?

A One-time random numbers used to make the proving keys. NAKA's were made by the team alone, and they can't take the piece from your vault; a multi-party ceremony is planned for the next collection (appendix D).

Trading and safekeeping

> **In SHORD your keys decide, and every step is final once it's on chain. Check these before you act and your assets stay safely yours.**

01 Use your bookmarked official site and check the address before signing the unlock message, which is for the official site only: never give it to anyone.

02 After you pay, the sale is done only when the BTC confirms, the indexer accepts it and a new note appears in your vault.

03 Private deals move BTC and note separately, with no lock holding either side: trade only with people you trust and have them confirm it in their vault (escrow: appendix K).

04 Fees vary: NAKA charges the buyer 2% (min. 1,000 sats), no royalty; Trollbox and others, 1% plus 1% royalty.

05 If you want a week's reward, don't list, cancel or transfer during that week, and don't send to yourself either.

06 The web vault opens from your wallet, and the same wallet and payment address always reopen it: keep the wallet's phrase safe. The Shielded Vault app has its own 24 words, not UniSat's or any wallet's; write them on paper, no screenshots.

07 Pieces and pouches go to the private address, BTC to the Bitcoin page's; inscriptions or runes sent there are spent as BTC. The QR code and link that move a vault to a phone are the vault: don't share them, and never open a vault link someone sends you, or what you put in is theirs.

08 Prepaid relays live only in the browser that bought them: use them up before switching devices or clearing data. Relay credits in a note go with the piece to the buyer.



The cast: A is #2, B #0, buyer C #21, and #4 the piece bought and sold; visitors #12, #14, #42; in the museum #15, #1, #3; minting #7; showing a proof #46; running indexers #37, #49.

Scan to read online

`ordinalser.github.io/naka-sigil/learn/shord`

Read online or download either PDF.

Scan to join Naka's Chinese Telegram

`t.me/+fRT4yVcxZ6VlNTBl`

Questions, talk and trading.

This edition: 3 November 2026. Live data from 28 October; the relay rechecked 2 November, the project's posts on X to 2 November, the site's vault, market and lounge on 3 November. Code: public commit 3cdbdfdf (12 October) and the site client (SHA-256 57457629..., read 28 October, unchanged 3 November). Every claim's source and check date, and later changes, are under Sources; the appendix is on the web only. The booths' Ord Dropz logo is redrawn in pixels from the official icon; the eyes in the window are this guide's own.