

Janmat: Technical & User Documentation

Oct 1, 2026 · @Anas

1. Overview

Janmat is a voting portal where no single party, including the government, has to be trusted: every vote is secret from everyone, every voter can check their vote was counted, and nobody can change the result without being caught. It covers every election in India from Lok Sabha to Gram Panchayat on one shared public ledger.

This document serves two readers. Sections 2 to 6 are for people who use the portal (voters, officials, auditors). Sections 7 to 14 are for engineers and reviewers who need to know how it works, how it is deployed, and what happens when it is attacked.

The four guarantees

Guarantee	What it means for a voter	How it is enforced
Secret ballot	Nobody can see who you voted for, not officials, not the government, not the operators	Blind-signed anonymous credentials cut identity off the ballot; ballots are encrypted on the voter's phone; only constituency totals are ever decrypted
Individual verifiability	You can check your own vote was recorded, unchanged, and counted	Receipt code, Merkle inclusion proof, and a full recount any phone can run
Tamper evidence	Any change to a past vote or result is detected	Hash-chained blocks signed by at least 4 of 5 independent guardians
Zero trust	No official can act alone or outside their role	Every official action is a signed transaction that all guardians re-check; decryption needs 3 of 5 trustees

Prototype versus production

The code in this repository runs the full cryptographic protocol for real: blind RSA signatures, ElGamal encryption with zero-knowledge proofs, distributed key generation, threshold decryption and a signed hash chain. What is simulated is the surroundings.

Area	In this prototype	In production
Aadhaar	A synthetic population, OTP always 123456	Licensed AUA/KUA calling UIDAI authentication
Guardians	Five nodes inside one Node.js process	Five organisations, each running its own servers
Official keys	Stored in JSON files on the server	Hardware security modules and smart cards held by each official
Network anonymity	Ballots sent without cookies or login	An anonymising relay so even the IP address is hidden
Scale	Hundreds of ballots	About 97 crore electors, sharded by state

2. Roles and logins

Every role holds its own signing key, and the guardians only accept a transaction signed by the key registered for that role in the genesis block. Holding a login is not enough: an official who tries something outside their role is refused by all five guardians.

Role	Who in real life	Can do	Cannot do
Voter	Any eligible citizen	Log in with Aadhaar, get one anonymous voting pass per election, vote, re-vote, check their vote	Vote twice (the second ballot replaces the first), see anyone else's vote
Election Commission (ECI)	Chief Electoral Officer	Create elections, open polls, close polls, declare results in an emergency	Add candidates, decrypt anything, open polls before keys, key ceremony and candidate lists are done
Credential Issuer	ECI IT wing	Publish one blind-signing key per constituency; issue anonymous voting passes	See which anonymous key it signed, issue a second pass to the same person

Role	Who in real life	Can do	Cannot do
Returning Officer (RO)	District election officer	Accept nominations, finalise the list, declare the result for their own district	Act for another district, change a count
Trustee (key-holder)	ECI, Supreme Court nominee, Leader of Opposition nominee, CAG, academic panel	Join the key ceremony; after polls close, publish a proven decryption share of the totals	Decrypt alone, decrypt a single ballot, decrypt before polls close
Guardian (ledger node)	ECI, Supreme Court registry, CAG, opposition-nominated node, civil-society node	Check and sign every block, audit its own copy, go offline and catch up	Forge a block alone (needs 4 of 5 signatures)
Identity gateway	Authentication User Agency	Check Aadhaar and electoral roll, issue a signed eligibility statement with per-election pseudonyms	See ballots
Public auditor	Anyone	Download and re-verify the entire ledger, every proof, every result	Nothing is hidden from them except who voted how

Demo credentials

Officials sign in at **Officials' portal** (top bar). Every official passcode is `demo123`.

Username	Role
<code>eci</code>	Chief Electoral Officer
<code>issuer</code>	Credential Issuer
<code>ro.pune</code>	Returning Officer, Pune district
<code>ro.mumbai</code>	Returning Officer, Mumbai City district
<code>ro.delhi</code>	Returning Officer, New Delhi district

Username	Role
trustee.eci , trustee.sc , trustee.lop , trustee.cag , trustee.acad	The five trustees
guardian.eci , guardian.sc , guardian.cag , guardian.opp , guardian.civ	The five guardian nodes

Voters log in with Aadhaar on **Log in**. The OTP is always 123456 , and the login page has a "Use demo code" button.

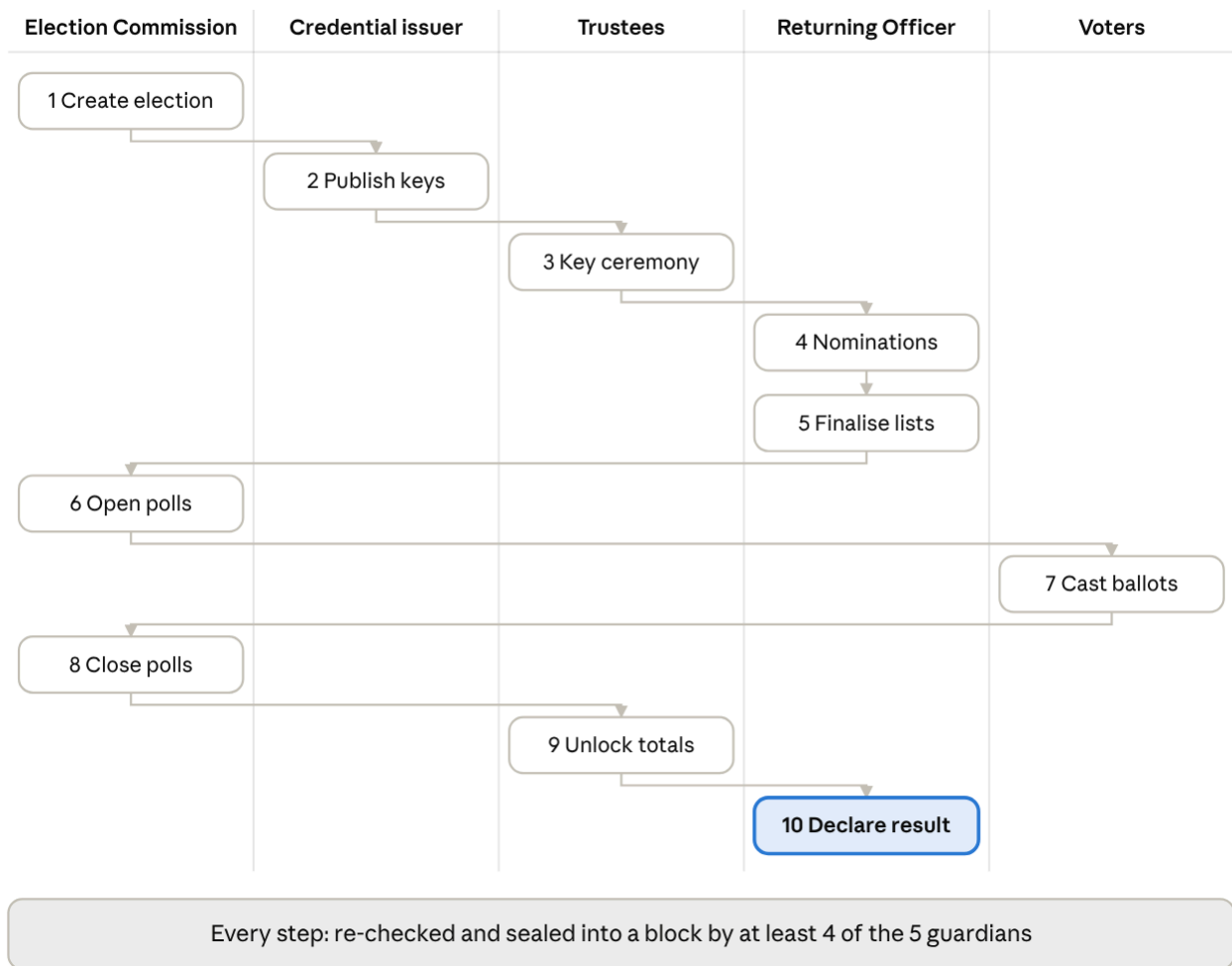
Aadhaar	Name	Area	Votes in
3136 6175 9250	Ananya Kulkarni	Kothrud, Pune	Lok Sabha (Pune), Vidhan Sabha (Kothrud), Pune Municipal (Ward 12)
8270 2971 4762	Harpreet Singh	Shivajinagar, Pune	Lok Sabha (Pune), Vidhan Sabha (Shivajinagar), Pune Municipal (Ward 14)
5807 3011 8825	Imran Shaikh	Colaba, Mumbai	Lok Sabha (Mumbai South), Vidhan Sabha (Colaba)
5679 1084 5302	Omkar Bhosale	Paud, Pune	Lok Sabha (Baramati), Vidhan Sabha (Bhor), Gram Panchayat Paud
4263 2294 0729	Lakshmi Iyer	Connaught Place, New Delhi	Lok Sabha (New Delhi)
8448 8859 2031	Rohan Pawar	Kothrud, Pune	None: age 15, shows the ineligible path

A voter only sees elections that exist on the ledger and cover their area, and only once the Election Commission has created them.

3. The end-to-end chain

An election is nothing but a sequence of signed transactions on the ledger, and each one is accepted only after all five guardians independently check it. The chain starts at block 0, the constitution, which holds every party's public key, the recognised parties and symbols, and the constituency map.

Every step is signed by its own official and sealed by the guardians



election lifecycle · 10 signed steps across 5 roles

The work passes between five hands in a fixed order, and the guardians refuse any step taken out of turn, such as opening polls before the key ceremony or declaring before three trustees have decrypted.

#	Step	Signed by	Transaction	What every guardian checks before accepting
0	Constitution	All 5 guardians	manifest (genesis only)	Its hash matches the one published in the Gazette; quorum is at least $2/3 + 1$

#	Step	Signed by	Transaction	What every guardian checks before accepting
1	Create the election	Chief Electoral Officer	<code>election-create</code>	Signer is the registered ECI key; ID unique; every constituency exists and matches the election type (Lok Sabha to PC, Vidhan Sabha to AC, local to ward)
2	Publish issuer keys	Credential Issuer	<code>issuer-keys</code>	Signer is the issuer; exactly one RSA key of 2048 bits or more per constituency; only during setup
3	Key ceremony	Each of the 5 trustees	<code>dkg-deal</code> ×5	Signer is that trustee; one deal each; 3 public commitments that are valid curve points; one sealed share for every trustee
4	Nominations	Returning Officer of that district	<code>candidate</code> ×n	Signer is the RO for that constituency's district; list not yet final; symbol matches the party or is a free symbol for independents; one candidate per party; no duplicate symbol
5	Finalise the list	Returning Officer	<code>finalize</code>	Same RO; at least one candidate. NOTA is appended automatically

#	Step	Signed by	Transaction	What every guardian checks before accepting
6	Open the polls	Chief Electoral Officer	open	Issuer keys published, all 5 trustees dealt, every list final. The election key is fixed here as the sum of the trustees' first commitments
7	Voting	Voters, anonymously	ballot $\times$ many	Polls open; the credential is the issuer's blind signature for this constituency; ballot signed by its anonymous key; sequence higher than any earlier ballot from that key; zero-knowledge proofs show exactly one vote
8	Close the polls	Chief Electoral Officer	close	Signer is ECI; polls are open
9	Unlock the totals	At least 3 trustees	partial $\times$ (trustees $\times$ constituencies)	Polls closed; signer is that trustee; a Chaum-Pedersen proof shows the share was computed with that trustee's real key share, against the encrypted total the guardian recomputes itself

#	Step	Signed by	Transaction	What every guardian checks before accepting
10	Declare the result	Returning Officer (or ECI)	tally	Signer may act for that district; every decryption share used was published on the ledger; the encrypted totals match the ballots; decrypted counts match; counts add up to the number of ballots

What the chain looks like after a real run

This is the timeline the portal printed for a Pune Municipal election run end to end with 20 synthetic voters:

1. Block 1: the Chief Electoral Officer created the election with 2 wards.
2. Block 2: the Credential Issuer published 2 blind-signing keys.
3. Blocks 3 to 7: each of the 5 trustees joined the key ceremony.
4. Blocks 8 to 19: the Pune Returning Officer accepted 10 nominations and finalised both lists.
5. Block 20: polls opened.
6. Blocks 21 to 23: 20 encrypted ballots.
7. Block 24: polls closed.
8. Blocks 25 to 27: three trustees each re-verified the whole ledger and published decryption shares.
9. Blocks 28 and 29: the Returning Officer declared both results.

Every one of these is visible on the public **Timeline** page with its block number and transaction hash, and anyone can re-verify the whole chain on the **Is it fair?** page.

4. Running the demo

The demo starts with an empty ledger that holds only the constitution; you create every election yourself, either in one click from the Demo centre or by hand as each official.

Install and start

Requires Node.js 23.6 or newer (it runs TypeScript directly) and npm.

```
cd election
npm install
npm run dev      # API on :8787 and website on http://localhost:5173
```

Command	What it does
<code>npm run dev</code>	Starts the API (restarts on code changes) and the website
<code>npm test</code>	Runs the 24 end-to-end protocol checks, including every attack in section 11
<code>npm run typecheck</code>	Type-checks the whole codebase
<code>npm run reset</code>	Deletes <code>data/</code> : all keys, the ledger and every election. The next start runs a fresh key ceremony
<code>npm start</code>	Builds the website and serves everything from :8787

After a reset, a browser that voted before shows a warning that the public record changed. That warning is correct behaviour; in the demo, press "This is a fresh demo: trust the new record".

Fastest path: one click

1. Open **Demo centre** (top bar).
2. Pick an election template and a number of synthetic voters, then press **Run it end to end**.
3. Watch the live log: each line is one official signing one step.
4. Press **Open the public timeline** to see every transaction with its block number, then **See results** and **Re-count it myself**.

By hand: see every login do its part

1. Sign in as `eci` . Pick a template (for example Gram Panchayat, Paud) and press **Create election**.
2. Sign in as `issuer` and press **Publish blind-signing keys**.
3. Sign in as each trustee (`trustee.eci` , `trustee.sc` , `trustee.lop` , `trustee.cag` , `trustee.acad`) and press **Deal my sealed key shares**.

4. Sign in as `ro.pune` . Add candidates (name, party or independent with a free symbol), then **Finalise list**.
5. Sign in as `eci` and press **Open the polls**. The button stays disabled until steps 2 to 4 are complete.
6. Vote as a citizen: **Log in**, tap Omkar Bhosale (Aadhaar 5679 1084 5302), **Use demo code**, **Continue**, then vote. Add more with **Add 25 synthetic voters** in the Demo centre.
7. Sign in as `eci` and press **Close the polls**.
8. Sign in as any three trustees and press **Verify the ledger and unlock the totals**.
9. Sign in as `ro.pune` and press **Declare result**.
10. As the voter, open **Check my vote**: it now ends with "Your vote was counted".

Any step can also be done for you with **Do the next step for me** on the Demo centre card, which acts as the correct official for that step.

Try to break it

- **Is it fair? → Try to cheat**: add a fake vote, count a vote twice, change a past vote, or change a vote and rewrite the hashes. Each is refused or detected, with the reason.
- Sign in as `guardian.opp` and **Switch node off**. The other four keep going. Switch it back on and it re-verifies every block it missed.
- Sign in as `ro.mumbai` and try to act on a Pune constituency: the guardians refuse it.

5. Voter guide

Voting takes four screens and about two minutes: prove it is you, press Vote next to your candidate, confirm, keep your receipt. The design follows public-service sites such as the Aadhaar portal and is built to be usable by someone aged 60 to 70 on a phone.

Voting, step by step

1. **Log in**. Enter your 12-digit Aadhaar number (a picture shows where it is printed on the card), tick consent, and enter the 6-digit code sent to your mobile.
2. **Your elections**. You see only the elections for your area. Each card says your seat and whether voting is open.
3. **Is this you?** Your name, age, the last four digits of your Aadhaar, the election and your area. Press "Yes, this is me", or "No" to go back.
4. **Choose**. Each candidate has a number, a photo, the party symbol and a large blue **Vote** button, like the EVM. "Read all names aloud" and a "Listen" link under each name speak them.

5. **Check your choice.** The candidate is shown large, with their symbol. Press "Yes, cast my vote" or "No, choose again". Nothing is sent before this.
6. **Done.** A green tick and a receipt number such as 3B38-F0F1-A605. Write it down or photograph it. It does not reveal your vote.

Checking your vote

Open **Check my vote**. Your phone checks the public record on its own and shows one of four answers:

You see	It means
Your vote was counted	It is in the record, unchanged, and inside the verified final total for your seat
Your vote is safely saved	Voting has not closed yet; come back after the count
This older vote was replaced	You voted again later; the newer vote is the one that counts
Something is wrong	A check failed. Call the helpline with your receipt number

Under "More details" are three further checks: at least 4 of 5 guardians approved the block, this is your final vote, and only one voting pass was ever issued in your name.

If someone forces you

Vote the way they demand if you must, then later vote again privately. Only your last vote counts, and nobody, including the person who watched you, can tell whether you changed it.

The test vote

On the confirm screen, "Not sure this app is honest? Do a test vote first" shows exactly what the app locked inside the ballot. That test ballot is destroyed and never counted, and you then vote for real. Because the app cannot tell a test from a real vote, a dishonest app would be caught.

Help for everyone

Control	Where	What it does
हिन्दी / English	Header and bottom bar	Switches the whole voter journey
Read aloud	Bottom bar	Reads the screen slowly in the chosen language, using the phone's own voice
Text size A- to A++	Top bar	Up to 1.3 times larger text everywhere
Light / Dark	Top bar and bottom bar	Colour theme; follows the phone until you choose
Help	Bottom bar	Helpline button and plain answers to the four most common questions

6. Officials' guide

Each official works from their own desk in the Officials' portal, and after every action the desk shows the block number and transaction hash that sealed it. Each desk also links to the election's public timeline.

Election Commission (`eci`)

1. **Create an election.** Pick a template or fill in the ID, type, name, jurisdiction and polling day, then tick the constituencies. The list only offers constituencies of the right kind for the type.
2. **Watch readiness.** Each election card shows the nine lifecycle steps with who owns each one. The step that is waiting glows gold.
3. **Open the polls.** Enabled only when the issuer has published keys, all five trustees have dealt, and every list is final. Opening fixes the election's public encryption key.
4. **Close the polls.** Any ballot already queued is sealed first; after this, every ballot is refused.

Credential Issuer (`issuer`)

1. For each new election press **Publish blind-signing keys**. The server generates one RSA-2048 key per constituency; only the public halves go on the ledger.

2. While polls are open the issuer works automatically: it checks each voter's signed eligibility statement, refuses a second pass for the same person, and blind-signs. The desk shows how many passes were issued.

Returning Officer (`ro.pune` , `ro.mumbai` , `ro.delhi`)

1. For each constituency in your district: enter the candidate's name, choose the party (its symbol is fixed) or Independent (choose a free symbol: bell, kite, umbrella or lantern), then **Accept nomination**.
2. After scrutiny press **Finalise list**. NOTA is added automatically and the list can no longer change.
3. After polls close and at least 3 trustees have unlocked the totals, press **Declare result**. The desk shows how many decryptions are published so far.

Trustee (`trustee.*`)

1. **Key ceremony**, during setup: press **Deal my sealed key shares**. Your desk shows which of the five trustees have dealt.
2. **Unlock the totals**, after polls close: press **Verify the ledger and unlock the totals**. Your node first re-verifies the entire ledger from block 0; if any block fails, it refuses and says why. Otherwise it publishes one proven decryption share per constituency.
3. Any 3 of the 5 trustees are enough; the other two are a safety margin if someone is unavailable or refuses.

Guardian (`guardian.*`)

The desk shows the node's status, its height against the network, and a log of every block it was asked to sign, with the reason for any refusal.

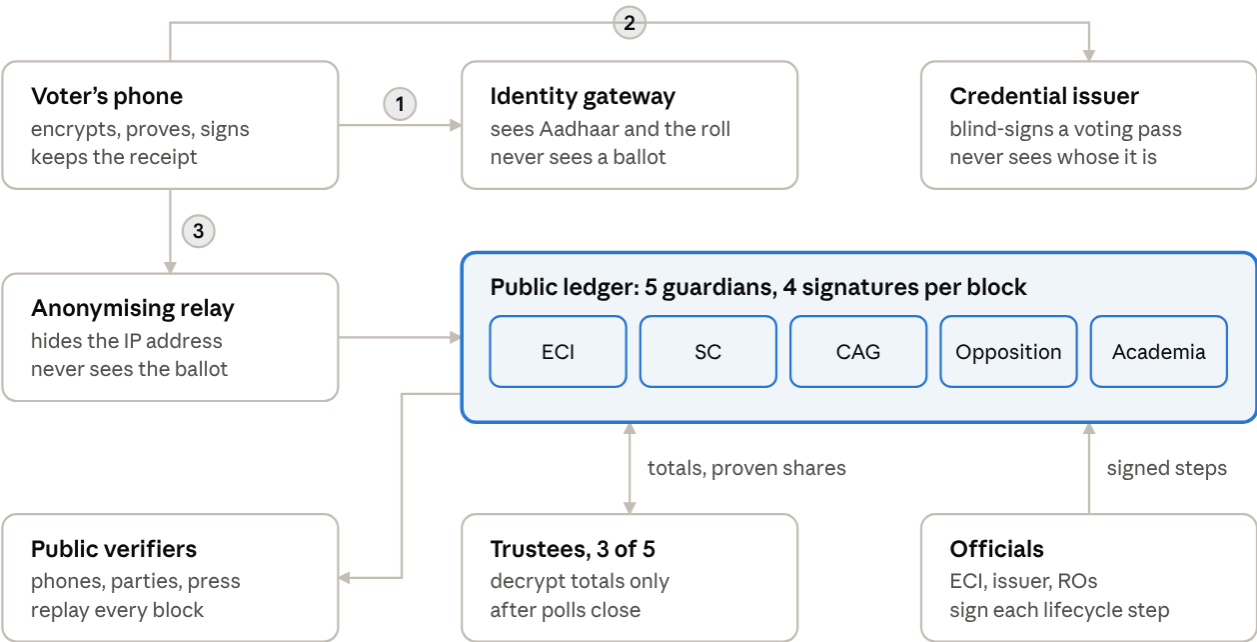
Button	What happens
Switch node off	The node stops signing and storing blocks. With 4 others healthy, the network continues
Switch node back on	It fetches every block it missed and re-verifies each one (signatures, hashes and every transaction) before appending
Full audit of my copy	Replays the node's entire copy from genesis, checking every signature, proof and decryption
Repair from honest peers	Only shown if tampering was detected; replaces the local copy with the verified majority copy

If two guardians are off, only 3 of 5 remain, below the quorum of 4: the ledger pauses safely and queued transactions wait until a node returns. It never accepts a block with fewer than 4 signatures.

7. System architecture

The system is split so that no component sees both who a voter is and what they voted: identity lives only in the gateway, eligibility passes through the issuer as a blinded value, and the ledger only ever holds anonymous encrypted ballots.

Identity and ballot never meet in one place



architecture · who talks to whom, and what each part can see

Numbers 1 to 3 follow a voter: Aadhaar login at the gateway, a blind-signed pass from the issuer, then the encrypted ballot through the relay onto the ledger, which everyone else only reads or writes signed steps to.

Components

Component	Runs where (production)	Holds secrets	Sees
Voter's phone or browser	The voter's device	Anonymous key, receipt codes, optional note of the choice	Everything about their own vote, nothing about others

Component	Runs where (production)	Holds secrets	Sees
Identity gateway	AUA data centre connected to UIDAI	HMAC pepper, gateway signing key	Aadhaar number, name, age, address; never ballots
Credential issuer	ECI data centre, HSM	RSA private keys per constituency	A per-election pseudonym and a blinded number; never the anonymous key or the ballot
Guardian nodes ×5	Each institution's own data centre	Its block-signing key	Every transaction, all encrypted
Trustee stations ×5	Each trustee's own offline machine or HSM	Its signing key, sealing key, key shares	Encrypted totals only, after polls close
Official desks	Each office, with smart card or HSM	That official's signing key	Their own role's data
Public API / website	CDN + stateless API servers	None	Public ledger data

What happens on the network when someone votes

1. Phone → gateway: Aadhaar number and OTP. The gateway returns a signed eligibility statement listing, per election, the constituency and a random pseudonym.
2. Phone creates a fresh anonymous key, blinds it, and sends the statement plus the blinded value to the issuer. The issuer returns a blind signature. The phone removes the blinding and now holds a valid, untraceable voting pass.
3. Phone encrypts the vote with the election key, adds zero-knowledge proofs, signs with the anonymous key, and posts it to the ledger with no login, cookie or referrer attached (production: through an anonymising relay).
4. Guardians check and seal it. The phone fetches a Merkle proof and the block's signatures, and checks them itself.

Code map

Path	What it contains
<code>src/crypto/group.ts</code>	ristretto255 group arithmetic (libsodium WebAssembly)
<code>src/crypto/elgamal.ts</code>	Exponential ElGamal, 0-or-1 proofs, sum proof, Benaloh audit
<code>src/crypto/threshold.ts</code>	Feldman dealings, share verification, partial decryption, Lagrange combination
<code>src/crypto/blindsig.ts</code>	RSA full-domain-hash blind signatures
<code>src/crypto/merkle.ts</code>	Merkle roots and inclusion proofs
<code>src/crypto/protocol.ts</code>	Every transaction type, the validation rules, block format, full chain replay
<code>src/crypto/voter.ts</code>	The voter-side steps: credential, ballot
<code>src/server/identity.ts</code>	Identity gateway (mock UIDAI)
<code>src/server/issuer.ts</code>	Credential issuer
<code>src/server/trustees.ts</code>	Trustee key ceremony and decryption
<code>src/server/ledger.ts</code>	Guardian nodes, BFT rounds, outage and attack drills
<code>src/server/operations.ts</code>	Official actions, logins, timeline, demo autopilot
<code>src/server/api.ts</code>	HTTP routes, grouped by party
<code>src/web/</code>	The website; <code>verifier.worker.ts</code> re-verifies the whole ledger in the browser
<code>scripts/test-crypto.ts</code>	End-to-end lifecycle and attack tests

The same `src/crypto` code runs in the guardians, the trustees and the voter's browser, so the check a voter runs is exactly the check the guardians ran.

8. Cryptography

Seven standard building blocks do all the work; none is novel, and each is chosen so that its failure would be visible to a public verifier rather than silent.

Building block	Used for	Standard / library
Blind RSA signature (full-domain hash)	Anonymous voting passes	RSA-2048; production: RFC 9474 RSABSSA
Exponential ElGamal on ristretto255	Encrypting each ballot	libsodium ristretto255
Disjunctive Chaum-Pedersen proofs	Proving each ballot slot holds 0 or 1 and the slots sum to 1	Fiat-Shamir with SHA-512
Benaloh challenge	Letting voters test that the app encrypted their real choice	Reveal randomness, re-encrypt, compare
Joint-Feldman DKG with sealed shares	Creating the election key without anyone holding it	X25519 sealed boxes for shares
Threshold decryption (3 of 5)	Unlocking only totals, with proofs	Lagrange interpolation in the exponent
Ed25519, SHA-256, Merkle trees	Signing blocks and transactions, chaining and inclusion proofs	libsodium, noble-hashes

Anonymous voting pass

The phone makes a new key pair and computes $m = H(\text{"credential", election, constituency, anonymous key})$. It picks a random r and sends $m \times r^e \bmod n$ to the issuer. The issuer, after checking eligibility, returns $(m \times r^e)^d$. The phone divides by r and obtains m^d , a valid signature on a key the issuer never saw. Because r is random, every blinded request looks alike, so the issuer cannot link a signature on the ledger back to the request it answered.

Encrypted ballot

For a constituency with k options the phone produces k ciphertexts. Slot i encrypts 1 for the chosen option and 0 for the others:

$$E(m; r) = (g^{\{r\}}, \backslash; g^{\{m\}} \cdot h^{\{r\}})$$

Each slot carries a proof that it is an encryption of 0 or of 1, without revealing which. The product of all slots carries a proof that it encrypts exactly 1. Together these show the ballot holds exactly one vote, so a voter cannot vote twice in one ballot or cast a negative vote. Proofs are bound to the election, constituency and anonymous key, so a ballot cannot be copied under another pass.

Counting without opening ballots

ElGamal is additive in the exponent: multiplying every ballot's slot i gives an encryption of the total votes for option i . Any three trustees each publish $D_j = A^{x_j}$ with a proof that they used their real share x_j . Anyone combines them:

$$g^M = B \cdot \prod_{j \in S} D_j^{\lambda_j}$$

The count M is then recovered by trying 0, 1, 2 ... up to the number of ballots. No individual ballot is ever decrypted.

Key ceremony

Each trustee j picks a random polynomial of degree 2, publishes its three coefficients multiplied onto the base point (commitments), and seals $f_j(i)$ to every trustee i 's X25519 key. Each trustee opens its shares, checks each one against the dealer's public commitments, and adds them to get its key share. The election's public key is the sum of every dealer's first commitment. The secret key itself is never computed anywhere.

Ledger

Each block commits to the previous block's hash, the Merkle root of its transactions, its height, time and proposer. It is valid only with valid Ed25519 signatures from at least 4 of the 5 guardians listed in the constitution. Changing one byte of any past ballot changes the Merkle root and the block hash, and the attacker cannot produce four guardians' signatures on the new hash.

9. Data storage

Each party stores only its own data, in its own place, and the only shared data is the public ledger, which contains no personal information. In the prototype each party's store is one JSON file under `data/`, so the separation is visible on disk.

Party	Prototype file	Contains	Public?	Production store
Each guardian	<code>data/validator-<id>.json</code>	Its signing key and its own full copy of the ledger	Ledger yes, key no	Append-only database (for example RocksDB or PostgreSQL with WORM backup) on that institution's servers; key in an HSM
Each trustee	<code>data/trustee-<id>.json</code>	Signing key, sealing key, one key share per election	No	Offline HSM or air-gapped machine; shares destroyed after the result is final
Each official	<code>data/official-<id>.json</code>	That official's signing key	No	Smart card or HSM token held by the person
Credential issuer	<code>data/issuer.json</code>	RSA private keys per constituency; issuance log (pseudonym, constituency, random receipt, time)	Receipts and counts only	HSM for keys; database for the log
Identity gateway	<code>data/gateway.json</code>	Signing key and HMAC pepper	No	HSM; UIDAI's own audit trail sits with UIDAI
Voter's device	Browser localStorage	Anonymous key, voting pass, receipt codes, optional note of the choice	No, never leaves the device	Phone secure enclave or passkey

What the public ledger holds

- The constitution: public keys of every guardian, trustee and official; parties, symbols and the constituency map.

- For each election: its definition, the issuer's public keys, the trustees' public commitments and sealed shares, candidate lists, open and close.
- Ballots: anonymous key, blind-signature credential, encrypted slots, proofs, signature. No name, Aadhaar, IP address or timestamp beyond the block's.
- Decryption shares with proofs, and declared results with the officer who declared them.

What is never stored anywhere

- A link from a person to an anonymous key: the issuer only ever saw a blinded number.
- The election's decryption key: it exists only as five shares.
- A plaintext ballot: only encrypted slots are stored, and only totals are decrypted.

Retention

The ledger is kept permanently as the official record. Trustee key shares should be destroyed once the election petition period ends, so that no future group of trustees could decrypt individual ballots. The issuer's pseudonym-to-receipt log is needed only to answer "was a pass issued in my name" and can be reduced to counts after the petition period.

10. Zero-trust model

Zero trust here means that every claim anyone makes (an official, an operator, this website) can be checked by someone else with mathematics instead of taken on their word, and that no single party holds enough to cheat or to see a vote.

Party	Knows who you are	Knows that you took part	Knows how you voted	Can change a result
Identity gateway	Yes	No	No	No
Credential issuer	A per-election pseudonym only	That a pass was issued to that pseudonym	No	No
Gateway + issuer together	Yes	Yes (turnout)	No	No
Any guardian	No	No	No	No, needs 4 of 5

Party	Knows who you are	Knows that you took part	Knows how you voted	Can change a result
4 guardians colluding	No	No	No	They could sign a false block, but it fails every public audit (section 11)
Any trustee	No	No	No	No
3 trustees colluding	No	No	Only if they also broke the rule to decrypt single ballots	No
Election Commission	No	No	No	No, cannot sign ballots, decrypt, or count
Returning Officer	No	No	No	No, a wrong count fails validation
The voter	Yes	Yes	Yes, and can check it was counted	Can replace their own vote until close

Why the government cannot see your vote

- The government's officials never hold the decryption key: it is split among five institutions that include the opposition, the judiciary and academia.
- Even with the key, only totals are published; guardians do not accept a decryption of anything but the aggregate for a closed constituency.
- Even with every ballot decrypted, a ballot carries an anonymous key that nobody can link to a person, because the issuer signed it blind.

Why the government cannot change your vote

- A ballot is signed by the voter's anonymous key; nobody else can produce a replacement.
- Every block needs 4 of 5 independent guardian signatures, and each guardian re-checks every proof.
- Every voter's phone and any public auditor can replay the whole ledger from the genesis block, whose fingerprint is published in the Gazette.

Why you cannot sell or be forced into your vote

- The receipt proves your ballot is counted, not whom it is for. The encryption randomness is deleted after casting.
- You can re-vote privately; only the last ballot counts, and a coercer cannot tell whether you did.

11. Threats and compromise: what if it is hacked

The design assumes that any one component will eventually be hacked, including through a zero-day nobody knows about yet, and makes sure that a breach of one part can neither change a result unnoticed nor reveal how anyone voted. What a breach can do is deny service or leak turnout, and both are visible.

Attacks the prototype demonstrates and refuses

All of these are exercised by `npm test` or the "Try to cheat" panel:

Attack	Outcome	Why
Insert a vote from a person who does not exist	Refused by all honest guardians	No issuer blind signature for that key
Submit a real ballot twice	Refused	Its sequence number is not higher than the last one for that pass
Put two votes inside one ballot	Refused	Zero-knowledge sum proof fails
Edit a past ballot on a guardian's disk	Detected at once; that node takes itself offline	Merkle root no longer matches the signed block
Edit and recompute every hash	Detected	New hashes carry no valid guardian signatures
Returning Officer acts for another district	Refused	Signer not authorised for that district
Officer declares made-up counts	Refused	Recomputed decryption does not match
Trustee publishes a false decryption share	Refused	Its Chaum–Pedersen proof fails against that trustee's public key

Attack	Outcome	Why
Declare using a share never published	Refused	Every share must already be on the ledger
Open polls before the key ceremony	Refused	Readiness check fails
Two guardians switched off	Ledger pauses, nothing lost	Quorum of 4 not reached; queued transactions wait

Component by component

If an attacker fully controls...	They can	They cannot	How it is noticed	Recovery
One guardian	Refuse to sign; serve a doctored copy to people who ask that node	Get any block accepted; change any past block	Voters' phones and auditors cross-check copies; the doctored copy fails replay; its self-audit and the others' logs show refusals	Take it offline, restore from the other four copies, rotate its key by constitutional amendment
Two guardians	Halt new blocks (safety over liveness)	Change anything	Ledger stops; everyone sees it	Bring either back; the backlog seals automatically
Four or five guardians	Sign an invalid block	Make it pass public replay: every block is re-checked by phones, trustees and auditors, not only by guardians	Trustees refuse to decrypt; any replay names the bad block; Gazette-published block hashes diverge	Election declared void for affected constituencies; guardian institutions answerable in court

If an attacker fully controls...	They can	They cannot	How it is noticed	Recovery
Credential issuer	Mint passes for fake voters (ballot stuffing); refuse real voters	Link passes to people; see votes	Passes issued exceed the electoral roll in a constituency; voters find no receipt of their own, or find a receipt they never requested; ballots exceed issuance log count	Revoke the constituency's key, re-run with a new issuer key; UIDAI-signed authentication logs show which issuances were genuine
Identity gateway	Impersonate voters (take passes in their names); refuse service	See votes	A voter who later logs in is told a pass already exists; issuance counts exceed real authentications in UIDAI's own log	Revoke affected passes; voters re-vote at a booth, which overrides online ballots
Up to 2 trustees	Refuse to take part	Decrypt anything	Their absence is visible on the desk	The other 3 decrypt
3 or more trustees colluding	Decrypt individual ballots (not only totals)	Link a ballot to a person; change a count	Only if they publish; nothing else is at risk	Production mitigation: a verifiable mixnet so trustees only ever decrypt shuffled ballots
The website or API servers	Serve a fake page or fake data	Fool a phone that pins the genesis hash and checks proofs itself; forge any signature	The phone shows "public record changed" or a failed check	Serve from mirrors; the checks run in the voter's own browser

If an attacker fully controls...	They can	They cannot	How it is noticed	Recovery
A voter's phone (malware)	Change that voter's choice before encryption	Do it without risk: a test vote reveals it	Test votes (Benaloh challenge) by any voter; independent verifier apps	Voter re-votes from a clean device or at a booth
The network (IP tracking)	Link a credential request to a ballot by timing and IP address	Read the ballot	Not detectable by the voter	Production: anonymising relay and random delay before casting

A zero-day in the software

A zero-day in the portal's own code, a library, the operating system or the browser is the most realistic serious threat. The defences are layered so that one flaw does not break the guarantees:

1. **Diversity.** Each guardian institution runs its own build on its own infrastructure; production should use at least two independent implementations of the validation rules, so one bug cannot make all nodes accept the same bad block.
2. **Verification outside the system.** Results are only trusted once independent replays (the trustees' own, auditors', political parties', the voter's phone) agree. A bug that makes guardians accept a bad ballot also makes those replays reject it, unless the same bug exists in all of them.
3. **Paper and booths.** Remote voting sits beside polling booths with EVM and VVPAT. A booth vote overrides an online ballot, so an attacked online channel can be switched off without disenfranchising anyone.
4. **Response plan.** On discovery: freeze new transactions (two guardians switch off), publish the last good block hash, patch, have each guardian re-audit from genesis, and resume. Ballots already sealed remain valid and countable because they are self-verifying.
5. **Bug bounty and public code.** The whole codebase, including this document, should be public before any election, with a funded bounty and third-party audits (for example CERT-In empanelled auditors and academic reviewers).

12. Production deployment blueprint

In production each trust domain is a separate organisation with its own people, servers, keys and network, and the only thing they share is the public ledger protocol. The prototype's single process maps onto the parts below.

Infrastructure per institution

Institution	Runs	Minimum setup
Each of the 5 guardian institutions	One validator node plus a hot standby	Two data centres; HSM for the block key; append-only database with daily off-site backups; BFT engine such as CometBFT, HotStuff or Besu QBFT carrying these validation rules
Each of the 5 trustees	An air-gapped trustee station	HSM or offline machine; used only for the key ceremony and the count; ceremony filmed and observed by party agents
Election Commission	Official desks, credential issuer, public API	Issuer keys in an HSM; officials use smart cards; public API is stateless behind a CDN
Authentication User Agency	Identity gateway	Licensed AUA/KUA link to UIDAI; face authentication for liveness; electoral roll (ERONET) for eligibility
Anonymising relay operator	Oblivious HTTP relay	Run by an organisation independent of ECI, so it sees IP addresses but not ballots, while the ledger sees ballots but not IP addresses
Political parties, media, academia	Independent verifiers	Anyone can run the public replay and publish block hashes

Key ceremonies

1. **Constitution ceremony**, once: each guardian, trustee and official generates keys on their own HSM. Only public keys are collected into the genesis block. All five guardians sign it. Its hash is published in the Gazette of India and in newspapers.

2. **Election key ceremony**, per election: trustees deal on the ledger, as in the prototype. Observers watch; complaints about a bad share are raised on the ledger. Production uses the Pedersen / Gennaro DKG variant, which also removes key bias.
3. **Key rotation**: guardian and official keys are rotated by a constitutional amendment transaction signed by the old set, never by editing files.

Scaling to a national election

- About 97 crore electors and roughly 60 crore votes over several weeks of phased polling.
- Shard by state: one ledger per state plus a national ledger that records each state's block hashes.
- Ballot verification (about 6 ms per option per ballot on one core) is parallel: guardians verify on many cores, and zero-knowledge proofs can be batch-verified.
- Storage: roughly 10 KB per ballot, so about 6 TB per general election, well within one institution's storage.
- The count is cheap: homomorphic tallies mean trustees decrypt only a few thousand constituency totals, not 60 crore ballots.

Operations and monitoring

Signal	Alert when	Who acts
Guardian heights diverge	Any node more than 2 blocks behind for 1 minute	That institution's on-call
Refused transactions	A burst of refusals from one source	ECI security desk
Issuance versus roll	Passes issued in a constituency exceed 100% of its roll, or rise abnormally fast	ECI and the RO
Public replay	Any independent verifier reports a failed block	All guardians; the count halts until resolved
Genesis hash	Any mirror serves a different genesis	Public warning, mirror taken down

Security process

- Open-source code, reproducible builds, published audit reports.

- Third-party audits before each election cycle; CERT-In empanelled auditors plus academic cryptographers.
- Public bug bounty; a responsible-disclosure channel run outside ECI.
- A mock national election at scale before first use, with public red-team participation.

13. Legal, accessibility and known limitations

The cryptography is ready for a pilot; the law, the Aadhaar arrangement and the offline alternatives are not, and each needs a decision outside engineering.

Legal and regulatory

- **Representation of the People Act 1951 and Conduct of Election Rules 1961** would need amendment to allow remote electronic voting, ledger-based counting and declaration from verified decryptions.
- **Aadhaar Act 2016 and the Puttaswamy judgment (2018)**: Aadhaar authentication must be voluntary and purpose-limited. Voters without Aadhaar, or who decline it, need an equal path (EPIC card with in-person verification).
- **Digital Personal Data Protection Act 2023**: the gateway and issuer are data fiduciaries; their logs need purpose limits and deletion schedules (section 9).
- **Indirect elections** such as Rajya Sabha, Legislative Councils and the President use different electorates and, for Rajya Sabha, an open ballot for party MLAs. They need their own templates and electorate lists, not Aadhaar eligibility.

Accessibility and inclusion

- Implemented: Hindi and English, read aloud, text up to 1.3 times larger, light and dark themes, large touch targets, one question per screen, keyboard navigation and skip link, reduced-motion support.
- Needed: all 22 scheduled languages, screen-reader testing with blind voters, assisted voting for people who cannot read, a feature-phone or kiosk channel, and booth voting for anyone without a smartphone.

Known limitations

Limitation	Effect	Mitigation
Malware on a voter's phone	Could change that voter's choice	Test votes, independent verifier apps, booth override

Limitation	Effect	Mitigation
Coercion in the room until polls close	Remote voting cannot fully prevent it	Re-voting, booth override, short remote-voting window
Gateway and issuer colluding	Learn who took part (turnout), not votes	Issue passes to every eligible voter in advance, or use anonymous credentials (BBS+)
Three trustees colluding	Could decrypt single ballots	Verifiable mixnet; destroy shares after the petition period
Network metadata	IP and timing could link request and ballot	Anonymising relay, random delay
Losing the phone	Its voting pass is gone; a new one is refused	Production: re-authentication that revokes the old pass on the ledger, or a passkey backed up by the phone's platform
Prototype performance	About 0.5 s per voter on one laptop	Parallel verification and batching in production

14. API reference and glossary

The API is grouped by the party that would serve it in production; official routes need the `x-session` header returned by `/api/auth/login`, and the ballot route deliberately reads no identity at all.

Method and path	Who	Purpose
GET <code>/api/overview</code>	Public	Constitution, elections, readiness and counts
GET <code>/api/elections/:id/timeline</code>	Public	Every lifecycle transaction with block, actor and hash
GET <code>/api/elections/:id/decryption</code>	Public	Which trustees have published shares per constituency
GET <code>/api/results/:id</code>	Public	Declared counts, declaring officer, decrypting trustees
POST <code>/api/identity/otp</code> , POST <code>/api/identity/verify</code>	Voter → gateway	Aadhaar login; returns the signed eligibility statement

Method and path	Who	Purpose
POST /api/issuer/credential	Voter → issuer	Blind-sign a voting pass
GET /api/issuer/log/:id	Public	Issuance receipts and counts
POST /api/ledger/submit	Anonymous	Submit a ballot (ballots only; no login)
GET /api/ledger/tx/:id, GET /api/ledger/latest/:e/:key	Public	Where a ballot landed, with Merkle proof; latest ballot for an anonymous key
GET /api/ledger/chain?node=	Public	The full raw ledger, optionally one guardian's copy
GET /api/ledger/nodes, GET /api/ledger/events	Public	Guardian status and network log
POST /api/auth/login	Officials	Username and passcode; returns a session
POST /api/eci/elections, /api/eci/open, /api/eci/close	ECI	Create, open, close
POST /api/issuer-admin/publish	Issuer	Publish blind-signing keys
POST /api/ro/candidate, /api/ro/finalize, /api/ro/declare	Returning Officer	Nominate, finalise, declare
POST /api/trustee/deal, /api/trustee/decrypt	Trustee	Key ceremony; unlock totals
GET /api/guardian/me; POST /api/guardian/pause, /resume, /audit, /repair	Guardian	Node status and operations
POST /api/demo/fullrun, /api/demo/autopilot, /api/demo/simulate, /api/demo/tamper, /api/demo/rogue	Demo only	Automation and attack drills

Glossary

Term	Meaning
Ballot	An encrypted vote: one encrypted slot per candidate, plus proofs and the voter's anonymous signature
Blind signature	A signature made on a hidden value, so the signer cannot later recognise what it signed
Constitution	The genesis block: every party's public key, parties, symbols and constituency map
Credential / voting pass	An anonymous key signed blind by the issuer; proves "eligible voter of this constituency" and nothing else
Guardian	A ledger node run by an independent institution; checks and signs blocks
Key ceremony (DKG)	How the trustees create the election key so that nobody ever holds the whole of it
Merkle proof	A short list of hashes that proves one transaction is inside a block
NOTA	None of the Above, added to every list when it is finalised
Quorum	4 of 5 guardian signatures needed for a block
Receipt code	The first characters of the ballot's hash; proves inclusion, not the vote
Threshold	3 of 5 trustees needed to decrypt a total
Trustee	A key-holder with one share of each election's decryption key
Zero-knowledge proof	A proof that something is true (this ballot holds exactly one vote) that reveals nothing else