AffixIO Technical Paper · WP-027
June 2026
affix-io.com

AFFIXIO WHITE PAPER · WP-027

Verifiable Elections Without Voter Privacy Compromise: Zero-Knowledge Proof Architecture for Ballot Integrity

How cryptographic voting systems can prove every vote was counted correctly without ever revealing who voted for whom.

AffixIO | United Kingdom | affix-io.com | June 2026

ABSTRACT

Democratic elections face a structural tension: verifying every vote was counted correctly requires transparency, while protecting each voter's choice requires secrecy. Today most large-scale elections resolve this through procedural controls rather than mathematical guarantees. Zero-knowledge proof cryptography offers a different model: a publicly auditable proof that all votes were correctly counted, no ineligible votes admitted, no eligible votes excluded, and the tally arithmetically correct, without revealing any individual ballot choice. This paper describes the cryptographic architecture that makes this possible, including ZK range proofs, verifiable mixnets, nullifier-based double-vote prevention, and post-quantum ballot signing with ML-DSA-65. It

> also describes how AffixIO's existing attestation infrastructure provides building blocks applicable to this domain, and what would be required before any such system could be deployed in real elections.

CONTENTS

SECTION 1

The Integrity-Privacy Tension in Elections

Any democratic election must satisfy two properties that pull in opposite directions: every vote must be counted accurately and verifiably, and no individual voter's choice must ever be revealed. These two demands are not merely in tension by convention; they are structurally antagonistic. Revealing who voted for whom would allow perfect verification of the count, but would also expose voters to coercion, social pressure, and retribution. Concealing every trace of how votes were cast provides robust privacy, but makes it impossible to audit whether the count was conducted honestly.

Current systems address this by separating the processes rather than resolving the tension. Paper ballots provide auditability through their physical existence; the secret ballot provides privacy through the anonymisation of the paper record at the point of casting; and the combination is achieved through procedural controls administered by election officials, scrutineers, and observers. The chain of custody for physical ballot papers, the conduct of the count, and the publication of results are all governed by legal frameworks and enforced by human oversight rather than mathematical proof.

The weakness of procedural controls is that they depend on trust in the people and institutions administering the process. In well-established electoral systems with strong independent oversight that trust may be well-founded and durable. But trust is not proof. When confidence in electoral administration erodes, for whatever reason, the tools available to restore it are limited: re-running the election is expensive and legally complex; independent audits of procedural compliance are slower and harder to communicate than mathematical evidence; and the absence of cryptographic proof means that the integrity of the count can be asserted but not demonstrated to sceptics who have chosen not to trust the administrators.

Zero-knowledge proof cryptography offers a different model. Rather than requiring trust in the administrators, it transfers the burden of verification to publicly checkable mathematics. An election system built on ZK proofs can generate publicly auditable evidence that every vote was correctly counted, that no ineligible votes were included, that no eligible votes were excluded, and that the final tally is the correct arithmetic sum of all counted ballots. Critically, all of this evidence can be provided without revealing any individual voter's ballot choice. The privacy and the verifiability are both guaranteed by the same mathematical construction rather than by competing procedural controls.

This paper describes the cryptographic architecture that makes this possible. It is important to state at the outset that this is an engineering description of what the mathematics can provide. It is not a commentary on any specific election, electoral system, or political context. The properties described here are properties of cryptographic protocols; they are as apolitical as a hash

function. Whether any particular democratic institution chooses to adopt this architecture is a political and administrative decision entirely outside the scope of this paper.

SECTION 2

What an Election System Actually Needs to Prove

Before designing a cryptographic voting system it is necessary to be precise about what the system actually needs to prove. The academic literature on verifiable voting has converged on four distinct properties, sometimes referred to collectively as the verifiability chain. Each property addresses a different point in the ballot lifecycle where something could go wrong, either through error or through deliberate manipulation.

Cast-as-intended

Cast-as-intended means that the voter's device correctly encoded their ballot choice: if the voter selected candidate A, the cryptographic encoding that left the voter's device represents a vote for candidate A and not for someone else. This property is particularly important in electronic voting contexts, where the voter interacts with software rather than marking a physical paper. Malicious or buggy software could silently change a voter's choice at the point of encoding. A cast-as-intended verification mechanism allows the voter to independently confirm, using tools separate from the voting device, that what was recorded matches what was intended.

Recorded-as-cast

Recorded-as-cast means that the ballot submitted by the voter is the same ballot that the system received and stored. A system that accepted ballots but then silently substituted different ones before storing them would appear to function correctly but would not be recording the electorate's actual choices. Recorded-as-cast verification typically involves the voter receiving a

cryptographic receipt at the time of voting, which they can later compare against the public record of stored ballots to confirm that their ballot appears unchanged.

Counted-as-recorded

Counted-as-recorded means that all stored ballots were included in the final tally, without modification or duplication. A system that stored ballots correctly but then selectively excluded some from the count, or included the same ballot multiple times, would produce an incorrect result even if all the individual storage operations were honest. This property requires that the tally process is verifiably complete and non-duplicating with respect to the set of stored ballots.

Tallied-as-counted

Tallied-as-counted means that the published result correctly reflects the arithmetic sum of all counted ballots. This is the property that manual recounts are designed to verify. In a ZK system it is achieved by providing a publicly verifiable proof that the published totals are the correct sums of the encrypted ballot values, without decrypting any individual ballot.

Current paper ballot systems with manual counting satisfy recorded-as-cast (the physical ballot is the record) and provide auditable evidence for counted-as-recorded through manual recount procedures. They rely on procedural controls for cast-as-intended (the voter marks the paper themselves and can inspect it before submission) and tallied-as-counted (the count is conducted openly with observers present). ZK voting systems can provide cryptographic proofs for all four properties simultaneously, replacing procedural assurances with mathematical ones.

PROPERTY	WHAT IT GUARANTEES	PAPER BALLOT MECHANISM	ZK MECHANISM
Cast-as-intended	Device encoded the voter's choice correctly	Voter marks paper directly; inspects before submission	Commitment scheme with voter-verifiable challenge
Recorded-as-cast	Stored ballot matches submitted ballot	Physical ballot is the record	Cryptographic receipt matched against public bulletin board
Counted-as-recorded	All stored ballots included; none duplicated	Manual recount with observers	Verifiable shuffle proof over full ballot set
Tallied-as-counted	Published result is correct arithmetic sum	Open count with scrutineers	ZK range proof over encrypted tally

SECTION 3

The Existing State of Election Auditing

The practice of auditing elections has developed considerably over the past two decades, and it is important to understand what existing approaches achieve and where their limitations lie before describing what cryptographic systems could add. The current landscape is characterised by a range of techniques at different levels of statistical and mathematical rigour, with wide variation across jurisdictions.

Risk-limiting audits

Risk-limiting audits (RLAs) are a statistical sampling approach developed primarily by Philip Stark at the University of California, Berkeley. An RLA draws a random sample of ballots and examines them, continuing to draw samples until the statistical evidence is strong enough to support the reported outcome with a defined level of confidence, or until a full hand recount has been conducted. The defining property of an RLA is that if the reported outcome is wrong, the audit will detect this with at least the specified probability, which is set in advance by the election administrator. Several

jurisdictions have adopted RLAs as a standard post-election practice. RLAs provide rigorous statistical confidence, but not mathematical certainty: there remains a small probability, set by the confidence parameter, that an incorrect outcome would not be detected.

Ballot image audits

Some jurisdictions now scan all paper ballots during counting and publish the resulting images, allowing any member of the public to independently verify the scanning software's interpretation of each ballot. This approach is a significant improvement in transparency over opaque optical scanning without published images. However, it provides verifiability only for the scanning stage; it does not address whether the published images are accurate representations of the physical ballots, nor whether the count of those images was conducted correctly. It is also dependent on the publication infrastructure being trustworthy.

Direct Recording Electronic machines with paper trails

Direct Recording Electronic (DRE) voting machines that produce a voter-verified paper audit trail (VVPAT) create a physical record that voters can inspect before confirming their vote. The paper trail can be audited independently of the electronic record. However, this approach requires manual inspection of the physical paper trail, which is labour-intensive, and the electronic record and the paper trail can diverge in ways that are difficult to detect systematically without a full recount of the paper trail. The security of DRE systems with VVPATs depends heavily on the manufacturing quality of the machines and the security of their software.

End-to-End Verifiable systems in deployment

End-to-End Verifiable (E2E-V) cryptographic voting systems have been deployed in a small but growing number of real elections. The Helios system has been used for elections run by the International Association for Cryptologic Research (IACR) and for student and faculty elections at several European universities since 2008. The Belenios system, developed at INRIA in France, has been used for French professional and associational elections. The STAR-Vote system was designed for county-level deployment in the

United States and demonstrated significant design work, though it has not yet been deployed at scale in governmental elections. These deployments demonstrate that E2E-V systems are technically workable, but the transition to mainstream governmental electoral use has not yet occurred.

The gap between what is cryptographically possible and what is widely deployed is not primarily a technical gap. The cryptographic tools to make elections publicly verifiable without compromising voter privacy have existed for over two decades. The gap is primarily regulatory, logistical, and educational: existing electoral systems are embedded in legal frameworks, administered by established institutions, and operate with equipment and procedures that have been certified and procured through long bureaucratic processes. Cryptographic improvements require changes to all of these layers simultaneously. This paper describes what the cryptography can provide; Section 11 addresses what the non-cryptographic barriers are.

> **Note on scope:** This paper does not evaluate or comment on the adequacy of any particular jurisdiction's current auditing practices. It describes the general landscape of available techniques and what cryptographic additions could provide. Adequacy assessments are matters for electoral authorities, independent reviewers, and democratic accountability processes.

SECTION 4

ZK Range Proofs for Vote Tallies

In a valid election, each ballot contains a vote for exactly one candidate (in a single-choice system) or a valid combination of values (in preferential, approval, or score systems). The tally must reflect the sum of valid votes. A cryptographic election system needs to prove two things about the tally without decrypting individual ballots: first, that each ballot contains a valid vote (a value within the permitted range); and second, that the published tally is the correct sum of all those values. Zero-knowledge range proofs address both of these requirements.

Pedersen commitments and their properties

A Pedersen commitment allows a party to commit to a value without revealing it, while retaining the ability to prove later that the committed value has specific properties. Pedersen commitments are homomorphic: the commitment to the sum of two values is the product of the commitments to each value individually. This homomorphic property is what makes ZK tallying possible. A voter can commit to their vote, the election authority can compute a commitment to the tally by multiplying the ballot commitments together, and the result is a commitment to the correct sum, all without any individual ballot being decrypted.

Bulletproofs and range verification

Bulletproofs, introduced by Bunz et al. in 2018, are a class of ZK proof that allows a party to prove that a committed value lies within a specified numerical range without revealing the value itself. Applied to voting: a voter can produce a Bulletproof demonstrating that their encrypted ballot encodes a value of exactly 0 or exactly 1 for each candidate (in a single-choice system), and that the values across all candidates sum to exactly 1. This proof is compact, publicly verifiable, and requires no interaction with the verifier after it is generated. The election authority can publish the Bulletproofs alongside the encrypted ballots; any member of the public with access to the published verification parameters can independently verify every proof.

What range proofs prevent

Range proofs prevent two classes of attack that are otherwise difficult to detect in encrypted ballot systems. Ballot stuffing, where an attacker submits ballots encoding values greater than the maximum permitted, would increase the tally for a candidate without a corresponding legitimate vote; a range proof that no ballot contains a value above the permitted maximum makes this mathematically impossible. Negative-vote attacks, where a ballot encoding a negative value is submitted to reduce another candidate's tally, are similarly prevented by range proofs that verify all values are non-negative. These attacks are not merely theoretical; they represent real risks in any system that operates over encrypted ballot representations without formal validity proofs.

Public verifiability without trusted setup

Bulletproofs require no trusted setup ceremony, which is an important practical advantage in an electoral context. Trusted setup ceremonies generate public parameters that must be kept secure; if the secrets generated during setup are ever revealed, the proof system can be used to generate fake proofs. In electoral contexts, the requirement for a trusted setup creates a single point of failure and a significant ceremonial overhead. Bulletproofs avoid this by using only standard cryptographic assumptions without a setup phase, meaning that any party can verify any proof using publicly available parameters without needing to trust the party that generated those parameters.

The range proof approach applies directly to the tallied-as-counted property described in Section 2. Once every ballot has an associated range proof confirming it contains a valid vote, and the homomorphic tally computation has been verified, the published totals are mathematically guaranteed to be the correct sum of valid ballots. This guarantee holds regardless of the trustworthiness of the election administrators, the counting software, or the audit observers; it depends only on the mathematical soundness of the proof system.

SECTION 5

Mixnet-Based Anonymity with Public Verifiability

Range proofs and homomorphic tallying can prove that the tally is correct, but they do not by themselves sever the link between individual ballots and individual voters. If the system knows which ballot was submitted by which voter (as it must, in order to enforce one-vote-per-person, as described in Section 7), then a record associating voter identities with ballot positions exists and could potentially be used to infer how individuals voted. Verifiable mixnets are the cryptographic mechanism that breaks this link while preserving the ability to verify that the same set of ballots that entered the mixing process came out unchanged.

How a mixnet works

A mixnet is a cryptographic system that takes a set of encrypted inputs, applies a random permutation to their order, re-encrypts each one (so that the ciphertext changes even though the plaintext it hides is the same), and outputs the resulting set. The critical property is that the link between any given input and any given output is computationally infeasible to reconstruct without knowing the permutation that was used. Because the ciphertexts are re-encrypted, even an observer who can compare input ciphertexts to output ciphertexts cannot identify which output corresponds to which input. The permutation itself is discarded after use and never published.

Verifiable shuffle proofs

A plain mixnet provides anonymity but no verifiability: an observer cannot tell whether the output set contains the same ballots as the input set, or whether some have been dropped, duplicated, or substituted. A verifiable mixnet addresses this by requiring the mix operator to produce a ZK proof of correct shuffling: a proof that the output set is a valid permutation and re-encryption of the input set, without revealing the permutation itself. This proof is publicly verifiable by anyone. Multiple constructions of verifiable shuffle proofs exist in the academic literature, including those by Furukawa and Sako, by Neff, and by Bayer and Groth; all provide the same fundamental guarantee with different efficiency trade-offs.

Multiple mix nodes for decentralised trust

In practice, a mixnet for an election would typically be operated by multiple independent mix nodes rather than a single operator. Each node receives the output of the previous node, applies its own verifiable shuffle, and passes the result to the next. The final output is the set of ballots that has been shuffled by all nodes in sequence. The security property of a multi-node mixnet is that the anonymity of the final output is preserved as long as at least one of the mix nodes is honest: even if all other nodes collude, the permutation applied by the honest node is sufficient to sever all input-output links for that node's shuffled set. This means the trust requirement is distributed: no single operator needs to be fully trusted, only one needs to be honest.

Applied to voting, the sequence of operations is as follows. Voters submit encrypted ballots to the election authority, which publishes them on a public bulletin board. The ballot set is then passed through the mixnet, which shuffles and re-encrypts them in a publicly verifiable way. The shuffled, anonymised ballots are then decrypted by a threshold decryption process (requiring cooperation from multiple key holders, none of whom can decrypt alone) and the plaintext votes are tallied. The verifiable shuffle proof guarantees that the decrypted set contains exactly the same votes as the submitted encrypted set, and the range proofs on the encrypted ballots guarantee that all votes in the set are valid. Combining these two proof types provides a complete publicly verifiable audit trail from ballot submission to final tally.

Threshold decryption

Threshold decryption is a closely related technique that complements mixnets in this architecture. Rather than holding a single decryption key (which would be a single point of failure and a privacy risk if misused), the decryption capability is shared among multiple key holders using a threshold scheme, such that decryption requires the cooperation of a minimum number of holders (for example, three of five). No individual key holder can decrypt any ballot unilaterally. The threshold decryption is typically applied to the shuffled, anonymised ballot set after the mixnet has completed its operation, so that even the key holders cannot link decrypted ballots to voter identities.

SECTION 6

Post-Quantum Signing for Ballot Roots

Election records occupy an unusual position in the landscape of digital records: they must be trusted for decades, not years. Constitutional challenges, historical analysis, judicial proceedings, and recount requests can all require access to ballot records well after the election that generated them. A signature that proves the integrity of a ballot root must therefore

remain trustworthy not just for the weeks or months following an election, but potentially for the decades during which the records might be needed as evidence.

The quantum threat to long-lived signatures

Classical digital signature schemes including RSA and ECDSA are vulnerable to quantum computers running Shor's algorithm. A cryptanalytically relevant quantum computer would be able to derive private signing keys from the corresponding public keys, allowing it to forge signatures on any document including historic ballot roots. The timeline for such quantum computers is uncertain, with estimates varying considerably across the research community, but the consensus direction of travel is clear: the window of safety for classical signatures on long-lived records is narrowing. For records that must remain trustworthy for twenty or thirty years from the point of creation, the quantum risk is not a future problem but a present one: signatures created today under classical schemes may become forgeable before the records cease to be relevant.

ML-DSA-65 and NIST FIPS 204

The National Institute of Standards and Technology finalised its post-quantum digital signature standard in August 2024 as NIST FIPS 204, standardising the Module-Lattice-Based Digital Signature Algorithm (ML-DSA) at three security levels. ML-DSA-65 is the middle parameter set, providing security equivalent to AES-192 against quantum attacks, and is the level that AffixIO's attestation infrastructure uses for audit record signing (described in WP-002). ML-DSA-65 generates signatures of approximately 3,309 bytes and public keys of approximately 1,952 bytes, substantially larger than classical counterparts, but these sizes are manageable in contexts where signatures are applied to Merkle roots rather than to individual records.

Merkle-anchored ballot roots

The efficient application of ML-DSA-65 to election records uses the same Merkle tree architecture described in WP-011 for AI governance audit records. Rather than signing each ballot individually, the entire ballot set is committed to a Merkle tree: each ballot is a leaf, and the root of the tree is a single hash

value that cryptographically commits to the complete set of ballots. Signing the Merkle root with ML-DSA-65 produces a single post-quantum signature that covers all ballots simultaneously. Any subsequent change to any ballot, addition of a ballot, or removal of a ballot, would produce a different Merkle root and invalidate the signature. Any third party can verify the signature on the root and then verify any individual ballot's inclusion in the committed set by verifying its Merkle path, without needing to access all other ballots.

Why post-quantum signing is necessary for election records

The long-lived nature of election records makes post-quantum signing not merely desirable but necessary for any election system designed for deployment in the coming years. A ballot root signed with ECDSA today would be protected for as long as ECDSA remains computationally secure against quantum attack; a ballot root signed with ML-DSA-65 today would remain protected regardless of future quantum computing advances, because ML-DSA-65's security rests on the hardness of Module Learning With Errors, which has no known quantum speedup. For institutions designing election infrastructure intended to remain in service for ten to twenty years, adopting ML-DSA-65 for ballot root signing from the outset is significantly simpler than migrating to post-quantum signatures after the infrastructure is deployed and operational.

This directly extends AffixIO's existing post-quantum attestation infrastructure to the electoral context. The Merkle-anchored, ML-DSA-65-signed audit records that AffixIO produces for AI governance purposes use the same architectural pattern that would be appropriate for ballot root publication and tally certificate generation in a ZK election system. The electoral application of this infrastructure would require additional development specific to the voting context, but the core cryptographic building block is production-proven.

SECTION 7

Double-Vote Prevention with Spent Registries

A fundamental requirement of any election is that each eligible voter can vote exactly once. Enforcing this in a cryptographic system that preserves voter anonymity is a non-trivial problem. The naive approach, maintaining a list of which voters have voted, breaks anonymity if that list is linkable to ballot choices. The voter privacy literature has developed several approaches to this problem; the nullifier-based approach is the most directly applicable to ZK proof architectures.

The nullifier concept

A nullifier is a value derived deterministically from a voter's credential using a one-way function, such that the same credential always produces the same nullifier, but the nullifier cannot be used to recover the credential. When a voter exercises their credential to vote, they publish the nullifier to a public registry. The ZK proof they submit with their ballot proves, among other things, that the nullifier was correctly derived from a valid credential, without revealing the credential itself. If the voter attempts to vote a second time using the same credential, they will produce the same nullifier, and the registry will detect the duplicate and reject the second ballot. If they attempt to construct a different nullifier from the same credential, the ZK proof will fail, because the derivation function is one-way and deterministic.

The spent registry as a public bulletin board

The spent registry is a public, append-only list of nullifiers. It is published throughout the election period and is readable by anyone. Voters, observers, and independent auditors can inspect the registry to verify that it contains no duplicates (each nullifier appears at most once) and that its size matches the reported number of votes cast. The registry reveals nothing about who voted for whom, because nullifiers are not linkable to voter identities or ballot choices by any party that does not hold the original credential, and the ZK proofs ensure that no credential can produce more than one valid nullifier per election.

Eligibility proofs alongside nullifier proofs

The nullifier mechanism handles double-vote prevention, but a complete system also needs to verify that each voter holds a legitimate electoral credential, that is, that they are eligible to vote. This is handled by an eligibility ZK proof submitted alongside the ballot. The voter proves, in zero knowledge, that their credential is a member of the set of valid electoral credentials (represented as a Merkle tree of registered voters), without revealing which credential they hold. The combination of the eligibility proof and the nullifier proof provides both eligibility enforcement (only registered voters can generate valid credentials) and one-vote-per-person enforcement (each credential can only generate one valid nullifier per election), without linking any ballot to any specific voter.

Credential issuance and its security assumptions

The security of the nullifier-based double-vote prevention mechanism depends on the integrity of the credential issuance process. If fraudulent credentials are issued (for example, by registering non-existent voters), those credentials can be used to submit additional ballots that will pass the eligibility and nullifier checks. The ZK proof layer can only enforce the properties of the credentials it receives; it cannot independently verify that the voter registration process was conducted honestly. This means that the security of a ZK voting system rests on two pillars: the mathematical soundness of the ZK proofs, and the integrity of the electoral roll from which credentials are derived. The first pillar can be verified by anyone; the second requires the same administrative oversight that underpins any other electoral system.

This nullifier architecture mirrors AffixIO's existing spent-proof registry described in WP-014 for double-spend prevention in digital asset contexts. The underlying mechanism, a ZK proof of credential validity combined with a public nullifier registry, is identical in structure. The electoral application adds the specific requirement that credentials be derived from electoral registration data, which introduces the dependency on the integrity of the electoral roll described above.

SECTION 8

End-to-End Verifiable (E2E-V) Voting Systems

End-to-End Verifiable (E2E-V) voting systems are the umbrella category for election systems that combine all of the cryptographic properties described in Sections 4 through 7 into a coherent architecture that voters and observers can verify from ballot casting through to final tally. The term "end-to-end" refers to the span of the verifiability guarantee: from the end of the voter's interaction with the system through to the end point of the published result.

The three verification properties of E2E-V

Cast-as-intended verification in E2E-V systems is typically achieved through a challenge mechanism at the point of voting. The voter's device produces an encrypted ballot. The voter can then choose, with some probability, to challenge the ballot rather than cast it. If challenged, the device reveals the randomness used to encrypt the ballot, allowing the voter (or an app on a separate device) to verify that the encrypted value correctly encodes the intended choice. The device knows there is a probability that any given ballot will be challenged, and therefore cannot systematically substitute the voter's choice without being caught with high probability across the electorate. Challenged ballots are spoilt and a fresh ballot is offered.

Recorded-as-cast verification is achieved through a cryptographic receipt that the voter takes away from the polling interaction. The receipt is a commitment to their ballot that appears on the public bulletin board once the ballot has been received and recorded. The voter, or an agent acting on their behalf, can check the bulletin board later and confirm that a ballot matching their receipt is present in the published set. If it is absent, or if the published commitment does not match their receipt, they have evidence of a recording failure that can be raised with the election authority or with independent observers.

Counted-as-cast verification is provided by the publicly verifiable proofs over the complete ballot set: the verifiable shuffle proofs (demonstrating that the anonymisation process preserved all ballots), the range proofs (demonstrating that all ballots encode valid votes), and the tally computation (demonstrating that the published totals are the correct sums). Any member

of the public, with access to the public bulletin board and a verification tool, can independently re-verify all of these proofs and confirm that the tally is consistent with the recorded ballot set.

Deployed E2E-V systems

Helios, designed by Ben Adida and first deployed in 2008, is the earliest widely-known E2E-V system. It has been used for elections run by the International Association for Cryptologic Research (IACR), for elections at several European universities including UCLouvain and the University of Surrey, and for some professional organisation elections. Helios operates via a web interface and produces a publicly verifiable set of proofs on a public bulletin board. Its primary limitation for large-scale governmental use is that it does not provide strong coercion resistance, a property discussed in Section 11.

Belenios, developed at INRIA Lorraine in France, extends the Helios architecture with improved trustee key management and has been used for a range of French professional and associational elections. The French data protection authority (CNIL) has approved its use for certain categories of election, making it one of the few E2E-V systems with formal regulatory endorsement for real elections.

STAR-Vote was a research and design effort led by researchers at the University of Texas at Austin and intended for deployment in Travis County, Texas. It incorporated an in-person voting terminal, a paper ballot for the voter to take away, and a cryptographic bulletin board with publicly verifiable proofs. The design work was substantial and has influenced subsequent E2E-V research, though the system was not ultimately deployed at scale in governmental elections. It remains the most detailed design for an E2E-V system intended for mainstream governmental use.

The complementary role of E2E-V

E2E-V systems do not replace electoral administration; they complement it by providing a mathematical audit layer that operates independently of the administrative process. Election officials still manage the registration of voters, the operation of polling facilities, the conduct of the election day

process, and the official declaration of results. The E2E-V layer provides a parallel mathematical record that any party can verify against the official results. Agreement between the official result and the independently verifiable proof provides strong evidence of integrity; discrepancy provides evidence of a problem that can be investigated.

SECTION 9

How AffixIO's Infrastructure Applies

AffixIO does not build election software and does not operate elections. The cryptographic infrastructure described in this paper represents the type of components that could be constructed using AffixIO's API and the underlying cryptographic primitives it provides. This section describes the technical correspondence between AffixIO's existing production capabilities and the architectural components that a ZK election system would require, so that organisations evaluating such a system can understand which building blocks already exist in production-proven form.

Merkle-tree audit infrastructure and ballot roots

AffixIO's Merkle-tree audit infrastructure, described in detail in WP-011, produces tamper-evident audit records for AI governance and compliance contexts. The same architectural pattern, building a Merkle tree over a set of records and signing the root with ML-DSA-65, is directly applicable to ballot root publication and tally certificate generation in a ZK election system. The production implementation handles the Merkle tree construction, root computation, ML-DSA-65 signing, and publication of verification bundles that allow any third party to verify any record's inclusion without access to the complete record set. These are exactly the operations required for publicly verifiable ballot root anchoring.

Post-quantum signing and long-lived records

AffixIO's production signing pipeline uses ML-DSA-65 (NIST FIPS 204) for all attestation records, with keys held in FIPS 140-2 Level 3 certified HSMs. This directly addresses the long-lived signature requirement described in Section

6. An electoral application of this infrastructure would sign ballot Merkle roots with the same ML-DSA-65 keys and produce the same type of verification bundle, with the additional electoral-specific metadata (election identifier, ballot period, jurisdiction) included in the signed payload. The post-quantum guarantee already present in AffixIO's production attestation infrastructure would transfer directly to the ballot root.

Nullifier-based spent registry

The nullifier-based double-vote prevention mechanism described in Section 7 mirrors AffixIO's existing spent-proof registry described in WP-014. The production implementation maintains an append-only registry of nullifiers, verifies that submitted nullifiers have not previously appeared in the registry, and provides ZK proofs of non-inclusion. The electoral adaptation would require modifying the credential format to reflect electoral registration data rather than digital asset ownership, and adding the electoral roll Merkle tree as the basis for eligibility proofs. The core nullifier registry mechanism, including the duplicate detection logic and the ZK proof verification, is already production-proven.

ZK eligibility proofs

AffixIO's API produces eligibility proofs in healthcare and identity attestation contexts, proving that a party satisfies specified criteria without revealing the underlying data. The ZK circuit structure for proving Merkle set membership, which is the core operation in voter eligibility verification, is the same type of circuit used in these existing contexts. An electoral credential verification circuit would prove that the voter's credential is a leaf in the electoral roll Merkle tree; this is structurally identical to proving that a patient identifier is a member of an eligible patient set or that an identity document is a member of a verified document registry.

Scope of what would be required beyond existing infrastructure

The building blocks described above are real and production-proven in their current application contexts. However, it would be misleading to suggest that connecting them would straightforwardly produce a deployable electoral system. The electoral-specific development required is substantial. Verifiable

mixnet construction is not currently part of AffixIO's production infrastructure and would require significant new engineering. Cast-as-intended verification mechanisms for voter-facing devices require specialised user interface design and security analysis. The threshold key management required for distributed decryption of ballot sets is a different operational pattern from AffixIO's current single-institution signing infrastructure. Independent security review, formal verification of the complete protocol, and regulatory engagement are all prerequisites before any deployment in real elections. This paper describes the technical architecture; it does not represent a product ready for electoral deployment.

SECTION 10

Regulatory and Standards Landscape

Election administration is primarily a national and sub-national regulatory matter. There is no single global standard for cryptographic election systems, and the regulatory requirements for electronic or cryptographically enhanced voting vary considerably across jurisdictions. This section describes the principal regulatory frameworks that are relevant to any organisation considering the deployment of ZK-based election infrastructure.

United States: EAC and the VVSG

In the United States, the Election Assistance Commission (EAC) is the federal body responsible for certifying voting systems against the Voluntary Voting System Guidelines (VVSG). The VVSG are voluntary rather than mandatory; states are free to adopt more or less stringent requirements. VVSG 2.0, adopted in February 2021, introduced a significant new principle of software independence: a voting system should be designed so that an undetected change or error in the software cannot cause an undetectable change in election results. This principle is closely aligned with the goals of E2E-V systems, but VVSG 2.0 does not mandate cryptographic E2E-V as the mechanism for achieving software independence. States that wish to deploy

cryptographically verifiable voting systems would need to seek EAC certification of the specific system under VVSG 2.0 requirements, which would require extensive testing and documentation.

European Union: Council of Europe recommendations and member state practice

The Council of Europe's Recommendation CM/Rec(2017)5 on standards for e-voting sets out security requirements for remote electronic voting, covering transparency, auditability, and protection of voter privacy. The Recommendation requires that e-voting systems be auditable, that voters be able to verify that their vote was cast as intended and recorded as cast, and that the secrecy of the vote be protected throughout the process. These requirements are consistent with E2E-V system design. Several EU member states have experience with electronic voting: Estonia operates a nationally deployed remote internet voting system that has been used in national elections since 2005, though it uses a different cryptographic architecture that does not provide full E2E-V properties. Other member states have piloted or considered electronic voting with varying degrees of cryptographic sophistication.

NIST interoperability standards and IEEE P1622

NIST Special Publication 1500-100, which defines common data formats for election systems, and the IEEE P1622 standard for voting system data interoperability, provide technical standards for how voting system data should be structured and exchanged. These standards are relevant to any ZK election system because the ballot data formats, audit log formats, and result publication formats all need to interoperate with existing electoral administration infrastructure. A ZK system that produced cryptographic proofs in a proprietary format that could not be verified with standard tools would have limited practical value; the proof formats need to be standardised and the verification tools publicly available.

Academic security models

The academic cryptographic community has developed several formal security models for verifiable elections. Universal Verifiability is the property that anyone, using only the published election record, can verify that the tally is the correct sum of valid cast ballots. Individual Verifiability is the property that each voter can verify that their own ballot was included in the tally. Coercion Resistance is the property that a voter cannot prove to a third party how they voted, even if they want to, which prevents vote-buying and coercion schemes. These three properties are distinct and not all E2E-V systems achieve all three simultaneously; coercion resistance in particular requires additional protocol complexity beyond what most deployed systems provide. ZK-based systems can be formally analysed against these properties using established security models, which provides a basis for evaluating specific designs against the academic state of the art.

Any organisation working towards deployment of a ZK election system in a real electoral context would need to engage with the regulatory framework of the specific jurisdiction, seek approval from the relevant electoral authority, and demonstrate compliance with applicable technical standards. This is a multi-year process even for well-resourced organisations with strong technical capabilities. The regulatory landscape is not a barrier to eventual deployment, but it is a significant factor in the timeline for any real-world application.

SECTION 11

What Would Need to Change for Deployment

The cryptographic components described in this paper are mature and, in several cases, already deployed in non-electoral contexts. The path from these components to a deployable election system is real but substantial. This section describes the principal categories of change required, not to discourage development in this direction, but to provide an honest account of the work involved so that organisations considering it can plan accordingly.

Regulatory approval and certification

Any cryptographic voting system intended for use in real elections would need approval from the relevant electoral authority before deployment. In the United States this means EAC certification against VVSG requirements; in EU member states it means compliance with national electoral law and any applicable Council of Europe recommendations; in other jurisdictions the specific requirements vary. Certification typically requires formal security analysis by accredited security laboratories, source code review, functional testing, and a period of deployment in lower-stakes elections (such as party primaries or local elections) before use in major national elections. This process takes years even when it proceeds smoothly, and it involves legal as well as technical review because changes to voting systems often require legislative changes as well as regulatory approval.

Voter accessibility and education

E2E-V systems provide their verifiability benefits only if voters actually use the verification mechanisms. A voter who does not check their ballot receipt against the bulletin board, or who does not use the challenge mechanism to verify cast-as-intended, receives no personal benefit from the cryptographic audit layer, though they still benefit from the universal verifiability that others exercising the system provide. Achieving meaningful verification participation rates requires both accessible verification tools (typically web applications that require minimal technical knowledge) and voter education about why verification matters and how to do it. Designing verification interfaces that are usable by the full range of voters, including those with limited digital literacy, limited English language proficiency, or visual impairments, is a significant design and testing challenge. International accessibility standards (WCAG 2.1 and equivalent national requirements) apply to electoral systems in many jurisdictions.

Coercion resistance

The systems described in this paper do not provide strong coercion resistance. A coercer who instructs a voter to vote in a particular way and then demands proof cannot easily verify that their instruction was followed (because the ballot receipt does not reveal the ballot content). However, the voter also cannot use the receipt to prove to the coercer how they voted, because the receipt is a commitment that hides the content. This provides

some protection against coercion, but it is not the formal coercion-resistant property defined in the academic literature. Formally coercion-resistant protocols, including the Civitas system and the JCJ protocol of Juels, Catalano, and Jakobsson, provide stronger guarantees by allowing voters to cast a fake ballot that overrides their original vote, so that even voters who have been compelled to vote under observation can subsequently cast a genuine vote. These protocols are significantly more complex to implement and have not been deployed at scale. For any jurisdiction where coercion is a material concern, the choice of E2E-V protocol needs to take this property into account.

Transition planning and decentralised administration

Existing elections worldwide are administered by thousands of decentralised jurisdictions with different equipment, legal requirements, administrative procedures, and budgets. A single county in the United States may operate different equipment from its neighbours; a single country in the European Union may have different requirements from adjacent countries. Transitioning to cryptographically verifiable systems requires long-term planning that addresses equipment procurement (typically on cycles of five to ten years), software certification (which requires lead times), staff training, and legal changes in jurisdictions where changes to voting system type require legislative action. This transition planning is entirely separable from the cryptographic design work, but it is a prerequisite for any large-scale deployment and it operates on timescales that dwarf the cryptographic development timeline.

Internet voting as a distinct question

Some discussions of ZK election systems conflate ZK-enhanced voting with internet voting. These are distinct questions. ZK proof techniques can be applied to in-person voting systems (where voters interact with a cryptographic terminal at a polling station) as well as to remote internet voting systems. The privacy and verifiability properties described in this paper are properties of the cryptographic protocol and apply in both contexts. However, remote internet voting introduces additional security considerations that are independent of the ZK proof layer, including the security of the voter's device, the security of the network over which votes

are transmitted, and the risk of large-scale malware attacks on the voter population. The security requirements for a remote internet voting system are substantially more demanding than those for an in-person cryptographic voting terminal, and the academic security community has historically been cautious about recommending large-scale internet voting deployment even with cryptographic enhancements.

SECTION 12

Conclusion

The cryptographic tools to make elections publicly verifiable without compromising voter privacy have existed for over two decades. What has not existed, until recently, is the combination of practical deployment infrastructure, post-quantum security, and accessible audit presentation that would make them usable at scale. That combination is now within reach. ZK range proofs using Bulletproofs are mature and do not require a trusted setup. Verifiable mixnets have well-understood academic foundations and have been deployed in real elections at small to medium scale. Nullifier-based double-vote prevention is a proven pattern applied in other ZK proof contexts. Post-quantum ballot signing with ML-DSA-65 is deployable today using production-proven infrastructure. These are not experimental ideas; they are engineering components with known properties and known limitations.

The deployment gap, the distance between what is cryptographically possible and what is widely deployed in governmental elections, is primarily regulatory, logistical, and educational rather than cryptographic. The path to closure requires long-term engagement with electoral authorities, patient navigation of certification processes, accessible tooling for voter verification, and sustained voter education. These are not cryptographic problems; they are institutional and social problems, which means they are solvable by the institutions and communities that administer elections, given sufficient political will and resources.

The value proposition is straightforward. An election whose integrity can be mathematically verified by any member of the public, using publicly available tools and published data, requires less trust in the individuals administering it. That is not a political statement; it is an engineering property. Mathematical proof is not partisan; it does not favour any party, candidate, or outcome. It favours accuracy. Whether any particular democratic institution chooses to invest in this infrastructure is a decision for democratic processes to make. But the infrastructure exists, it works, and the cost of adopting it continues to fall as the underlying components mature and standardise.

AffixIO's post-quantum attestation and audit infrastructure provides building blocks that are applicable to the electoral domain: Merkle-tree-anchored audit records, ML-DSA-65 post-quantum signing, nullifier-based spent registries, and ZK eligibility proofs. These components are in production use in governance and identity attestation contexts, and their application to electoral infrastructure is a matter of adaptation rather than invention. The electoral-specific development, security review, and regulatory engagement required to deploy them in real elections is significant and well beyond the scope of this whitepaper. The mathematical foundation is ready; the institutional work remains.

Related reading

- WP-002: Post-Quantum Attestation with ML-DSA-65
- WP-011: Merkle Tree Audit Architecture for AI Decision Systems
- WP-014: Double-Spend Prevention with ZK Nullifier Registries
- WP-001: Cryptographic AI Governance

Frequently asked questions

Does ZK voting mean online voting?

Not necessarily. ZK proof techniques can be applied to both in-person voting, where voters use a cryptographic terminal at a polling station, and remote voting. The privacy and verifiability properties described in this paper are properties of the cryptographic protocol, not of the voting channel. A polling-station terminal that produces a verifiable encrypted ballot and a voter receipt can provide E2E-V properties without any internet connection from the voter's perspective.

Can a ZK election system be hacked?

ZK voting systems shift the trust model: rather than trusting the election administration to count correctly, voters and observers can verify the mathematical proofs. However, the security of the system depends on correct implementation, secure key ceremonies for generating the public parameters, and the integrity of the voter credential issuance process. No technology eliminates all risk. ZK systems make certain categories of attack, including silent ballot substitution and undetected tally manipulation, mathematically impossible. Other categories of risk, including compromise of voter credential issuance, software supply chain attacks on voting terminals, and coercion of voters, remain and require different mitigations. A ZK election system is more auditable than a purely procedural one, but it is not immune to all forms of attack.

Is AffixIO proposing to run elections?

No. AffixIO provides cryptographic infrastructure components for AI governance, identity attestation, and audit trails. Electoral applications of these components would require extensive additional development specific to the voting context, independent security review by specialists in election security, regulatory approval from the relevant electoral authorities, and a sustained programme of testing in progressively higher-stakes deployments before any use in a major governmental election. This whitepaper describes the technical architecture of what a ZK election system would look like, not a product that is ready for electoral deployment.

AffixIO | affix-io.com | hello@affix-io.com

All whitepapers | Download PDF

- About
- Solutions
- Legal
- Trust & Security

Contact

truth layer | yes | no | proof