Bind: A Privacy-Preserving Data Verification Protocol

Version: 3.1
Date: December 2025
Status: Specification

Executive Summary

Bind is a privacy-first decentralized data verification protocol that enables data issuers to publish attestations once—as cryptographic commitments—and allows holders to prove ownership and specific data points using zero-knowledge proofs, without revealing underlying sensitive information. Initial POC was completed on zkVerify; long-term deployment target is Aztec, a privacy-preserving rollup on Ethereum, once production-ready.

Key innovations:

1. **Privacy by design**: Only data commitments (hashes) appear on-chain; sensitive information remains off-chain.
2. **Selective disclosure**: Holders choose exactly which data fields to reveal for each use case.
3. **Envelope-based issuance**: Issuers can issue attestations without knowing the holder's wallet address, maximizing privacy.
4. **Schema-agnostic**: Supports ANY structured data (identity documents, credentials, supply chain records, sensor data, attestations, etc.).
5. **Selective disclosure with cryptographic proofs**: Using Noir circuits,

holders can prove possession of data without revealing sensitive fields.

1. Introduction and Motivation

1.1 The Problem

Today, data verification is fragmented and costly:

- **Privacy erosion**: Each verification exposes sensitive data to new entities, increasing breach risk. Users are repeatedly asked to hand over the same documents and personal details (IDs, bank statements, payslips, medical records), and every new copy becomes another place it can be lost or stolen.
- **Unlimited copies of sensitive data**: In the name of KYC, onboarding, compliance, and background checks, the same identity and financial information is collected and stored by dozens of different organizations. Each system has its own security posture, retention policy, and access controls, so a single person's most sensitive information ends up scattered across many systems that they do not control.
- **Limited provenance**: There is no cryptographic proof of data origin or authenticity without revealing all details. Verifiers often must "see everything" to trust anything.
- **Repeated verification**: Users must repeatedly verify the same data points across different platforms and applications. A degree, employment history, or government ID is re-checked over and over instead of being proven once and reused.
- **Operational cost**: Repeated backend verification and data audits are expensive and operationally burdensome. Institutions maintain custom verification workflows, support teams, and manual processes for tasks that are conceptually identical.

- **Vendor lock-in**: Verified data is trapped within the platform that issued or collected it, preventing reuse across different services and applications. Each new platform restarts the verification process from scratch, further spreading sensitive data into additional systems.

1.2 The Solution

Bind solves these problems by:

1. **Anchoring attestations on a verification chain** as privacy-preserving commitments, making them reusable across applications.
2. **Using zero-knowledge proofs** for selective disclosure, so holders prove "I have data point X" without revealing sensitive details.
3. **Supporting envelope-based issuance**, allowing issuers and holders to interact privately before any wallet address is revealed.
4. **Enabling any schema**, from identity documents to supply chain records to employment history and sensor data.
5. **Optimizing costs** via external verification and aggregation, amortizing verification costs across many use cases.

1.3 Why Bind Is Different

Bind stands out in several ways:

- **Privacy-first**: Built from the ground up to minimize data exposure, not as an add-on. Only cryptographic fingerprints and minimal metadata live on-chain; raw data never does.
- **Universal**: It works for any structured data across industries (education, employment, supply chain, medical, government, IoT, property, and beyond).
- **Reusable**: Verify once, reuse many times, across many applications and platforms. One attestation, infinite use cases.

- **Decentralized**: Eventual goal is to be governed by a DAO and public smart contracts, not a single company. Community decides which schemas, issuers, and governance rules apply.
- **Integration-friendly**: Works with both Web3 and Web2 systems using standard APIs. Existing infrastructure can adopt Bind without major changes.
- **Holder-controlled**: The holder's device is the locus of control. Holders generate proofs locally, choose what to reveal, and maintain cryptographic proof of their data without surrendering copies to centralized servers.

1.4 Vision

Bind aims to become the privacy-preserving verification layer for the internet.

Imagine:

- **Citizens** verify identity instantly without surrendering personal privacy to governments or corporations. No forged documents. Cryptographic proof.
- **Patients** verify health-related facts without exposing full medical histories. Vaccination status or eligibility is proven without centralizing sensitive data.
- **Students** prove their education once across employers, lenders, and licensing bodies. Universities stop answering repeated verification requests.
- **Workers** prove employment without exposing salary or sensitive HR data. Loan applications and background checks complete in minutes instead of days.

- **Supply chains** prove authenticity and compliance without revealing trade secrets or competitive data. Manufacturers, distributors, and retailers trust each other cryptographically.
- **Organizations** collaborate on shared data without trusting centralized intermediaries. Privacy, trust, and decentralization coexist.

Core principle: One attestation. Infinite use cases. Holder control. Complete privacy. Decentralized infrastructure.

2. How Bind Works

2.1 Three Simple Steps

Step 1: Issuance

A data issuer (university, employer, government agency, etc.) creates an attestation—a claim about data they have verified. Instead of storing this on a central database, Bind publishes a cryptographic fingerprint (commitment/hash) to the blockchain. The actual data stays private, off-chain.

What goes on-chain: A mathematical hash proving "Issuer X has verified data for Holder Y."
What stays private: The actual data values (e.g., GPA, salary, medical details).

Step 2: Claiming

A holder receives the attestation data off-chain from the issuer. They generate a zero-knowledge proof on their device (typically a few seconds). This proof cryptographically proves:

- "I own valid data matching this on-chain commitment."
- "This data has not been revoked."
- "This data has not expired."

The holder stores this proof privately.

Step 3: Verification

When a holder wants to verify their data (e.g., for a loan, job, or access), they present their zero-knowledge proof to the verifying organization. The verifier checks:

- Does this proof match the on-chain commitment?
- Is this proof valid and unrevoked?

The verifier never sees the underlying raw data—only that it is valid.

Selective disclosure: The holder controls which specific data points to reveal.

- Example: Prove "has a Bachelor's degree from University X" while hiding GPA.
- Example: Prove "employed at Company Y between 2020–2024" while hiding salary.

2.2 Privacy and Security Built In

Privacy principles:

- **Data-minimizing by design**: Only cryptographic fingerprints and minimal metadata live on-chain. Raw sensitive data never appears on any public ledger.
- **Holder-centric**: The holder's device is the locus of control for proofs

and disclosure decisions. No central server stores user data or tracks proof usage.

- **Selective disclosure**: Every verification can be tailored to expose only what is strictly necessary. Users decide field-level what to share.
- **Issuer privacy** (envelope flow): Issuers can issue attestations without knowing which wallet or person will ultimately claim them. Perfect privacy during issuance.

Security principles:

- **Issuer accountability**: Only approved issuers can create attestations for each schema type. Bad actors can be removed by governance.
- **Revocation**: Issuers can revoke attestations, and revoked entries automatically fail verification.
- **No single point of failure**: On-chain logic, decentralized governance, and local proof generation reduce reliance on centralized servers.
- **Auditable design**: The protocol design and governance decisions are transparent and can be reviewed by the community and third-party auditors.
- **Cryptographic soundness**: Proofs are mathematically proven correct. Forging a proof is computationally infeasible.

3. System Architecture

Bind operates across three conceptual layers. While the conceptual design is chain-agnostic, the current implementation runs on zkVerify and is intended to migrate to Aztec when feasible.

3.1 On-Chain Layer (Registry)

Purpose: Provide a shared "truth layer" for attestations.

This layer stores:

- **Attestation commitments**: Cryptographic hashes that represent verified data.
- **Schemas**: Definitions of what fields each attestation type contains (e.g., EDUCATION, EMPLOYMENT, SUPPLY_CHAIN).
- **Issuer lists**: Which organizations are authorized to issue which schema types.
- **Revocation records**: Whether specific attestations have been revoked.

This layer does **not** store raw personal data. It only stores the minimum cryptographic information needed to verify proofs.

3.2 Holder Layer (User Devices)

Purpose: Give holders full control over their proofs and privacy.

On the holder's device:

- Attestation data is received from issuers via secure off-chain channels.
- Zero-knowledge proofs are generated locally, using Bind's proving logic.
- The holder chooses what to prove and what to reveal each time.

No central server ever needs to store full user data or proofs.

3.3 Integration Layer (Apps and Services)

Purpose: Make Bind usable by real-world systems.

- **Web3 dapps** can verify proofs directly against the on-chain registry.

- **Web2 services** (banks, HR systems, SaaS platforms) integrate via REST APIs, sending proofs to a verification service that checks against the on-chain registry.

This layer allows Bind to plug into existing workflows with minimal disruption.

4. Types of Data Verified (Schemas)

Bind is **schema-agnostic**: any structured data can be modeled.

Examples:

- **Educational**: University, degree, graduation date, GPA (optional/hidden), program, honors.
- **Employment**: Company, title, start/end dates, department, compensation range (often hidden), performance indicators.
- **Supply Chain**: Product ID, origin, certifications, handling data, timestamps, temperature logs.
- **Medical**: Vaccination status, blood type, allergies, key lab results (with strict selective disclosure).
- **Identity**: Age, nationality, residency status, document validity (without exposing full identity details).
- **IoT/Sensor Data**: Device ID, calibration status, reading ranges, timestamps.
- **Property/Assets**: Ownership, address or asset ID, valuation range, lien status.

Schemas can be standardized and governed by the Bind community so that verifiers across industries can rely on consistent formats.

5. Governance and Economics (Conceptual)

- **DAO governance**:
  - Manages schemas, issuer lists, and protocol upgrades.
  - Can pause parts of the system in emergencies.
- **Economic model** (to be finalized):
  - Small fees for issuing, claiming, and verifying attestations.
  - Fees fund protocol maintenance, audits, and ecosystem development.
  - Potential incentives for early issuers, integrators, and ecosystem contributors.

Details of the DAO structure and token mechanics are intentionally left as open questions to be validated with partners.

6. Roadmap (Conceptual)

- **Phase 1 – Pilot on zkVerify**:
  - Deploy core Bind contracts and proof verification on zkVerify.
  - Run pilot integrations with selected issuers and verifiers.
  - Prove concept and refine protocols based on real-world feedback.

- **Phase 2 – Dual-Home / Migration Preparation**:
  - Align proof formats and contracts with Aztec's stack.
  - Begin mirroring commitments and revocations from zkVerify to Aztec testnet.
  - Expand schemas and integrations (medical, government, IoT).

- **Phase 3 – Primary Deployment on Aztec**:
    - Switch primary issuance and verification flows to Aztec mainnet.
    - Mature governance, broader ecosystem adoption, and cross-industry standardization.
    - Maintain zkVerify as an archival / interoperability layer if needed.

7. Conclusion

Data verification today is repeated, invasive, and centralized. Users' and companies sensitive information is scattered across dozens of organizations, each with different security standards, creating a massive attack surface and privacy risk.

Bind addresses this by enabling:

- **Reusable verification**: One proof, many contexts. Stop re-verifying the same data over and over.
- **Privacy by design**: Raw data remains off-chain and under holder control. Only cryptographic proofs and commitments are public.
- **User ownership**: Holders decide what to reveal, when, and to whom. Field-level control over every disclosure.
- **Universal applicability**: Works across industries and data types—from education and employment to supply chains and IoT.
- **Decentralized trust**: No single entity controls the verification layer. Community governs schemas, issuers, and protocol evolution.

The result: A world where data is verified once and trusted everywhere, while users maintain complete privacy control.

Bind is the infrastructure for a privacy-first data economy where verification is efficient, privacy is guaranteed, and holders are in control.

Appendix: Glossary

- **Attestation**: A verified claim by an issuer about some data (e.g., a degree, employment history, product origin).
- **Bind**: The protocol that "binds" verified data to the blockchain for reusable, privacy-preserving verification.
- **Commitment (hash)**: A one-way cryptographic fingerprint of data that proves it exists without revealing it.
- **Data point**: Specific field within a dataset (e.g., university name, start date, nationality).
- **Envelope**: Optional flow where issuers share data and a secret with holders without knowing their address.
- **Merkle root**: A compact cryptographic summary of many data items, used for efficient verification.
- **Noir**: A language and toolkit used to create zero-knowledge proof programs.
- **Zero-knowledge proof (ZKP)**: A proof that you know something (or that something is true) without revealing the thing itself.
- **Schema**: A structured definition of what fields an attestation contains.
- **Selective disclosure**: Ability for a user to reveal some fields while keeping others hidden in a proof.