nu_i.d.

Solution Data Sheet

At a Glance

NuID provides a modern authentication solution that enables your business to authenticate users without having to store or manage their authentication credentials.

- ✓ **Eliminates risk of credential breach**
- ✓ **No disruption to existing login UX**
- ✓ **Privacy-by-design decentralized identity framework**
- ✓ **Leverages zero knowledge cryptographic protocol**

5.9 BILLION

passwords were breached in the last 3 years

81% OF BREACHES

are caused by stolen or weak passwords

NuID eliminates the need to store passwords

Authentication is stuck in the "shared secret" paradigm. Your users need to share their passwords with you and you need to store them for authentication.

This model means that *every* website and application is storing duplicated identity information in centralized silos that attract cyber criminals like bees to honey.

If you are storing passwords, you are taking a huge risk: passwords and credentials have been targeted in **57% of cyberattacks and breaches**, and **81% of breaches** result from stolen or weak credentials.

NuID leverages zero knowledge cryptography and distributed ledger technology to eliminate the need for users to share their authentication secrets with anyone—something we call **trustless authentication**. Now you can manage login for your web and mobile applications without taking on the risk of storing their authentication credentials.

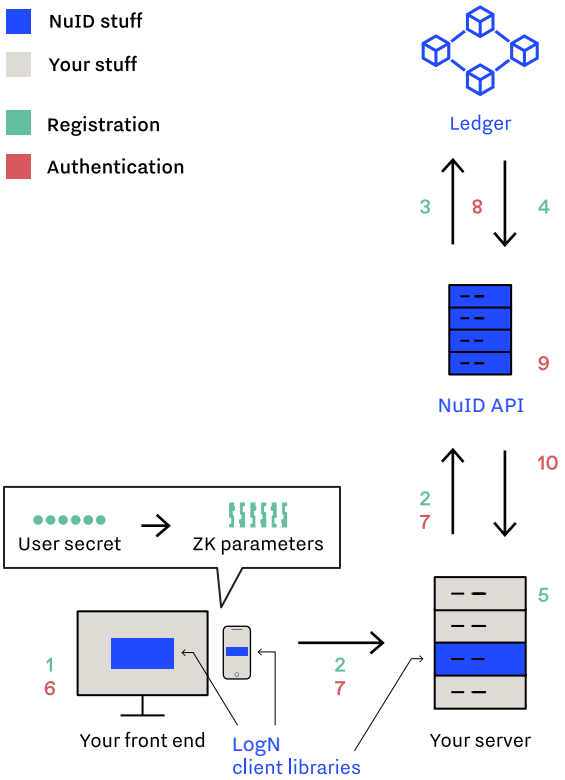
The modern login box

How is NuID different?
Instead of transmitting passwords from a device to your server to be verified for authentication, NuID uses zero knowledge cryptography that enables the user's device to "prove" the user provided the correct password. The actual password is never sent off-device, or stored anywhere, which means it can't be intercepted or stolen in a server-side breach.

User Experience
NuID eliminates the need to store credentials without requiring any changes to login UX. Unlike authentication solutions that rely on PKI, our SDK doesn't require users to store and manage private keys or download and use a separate authentication app. Your users can login with passwords, biometrics, and more, all without sharing any sensitive and private data.

info@nuid.io

nu_i.d.



Registration

1. User inputs a username/email and a new secret (e.g. password) and the LogN client libraries generate zk reference parameters from secret
2. Username and ZK parameters are sent to relying party and parameters are forwarded to NuID's service API
3. ZK parameters posted to ledger
4. Ledger returns the unique txid of where the parameters are located
5. Username is associated to the txid; registration complete

Authentication

6. User inputs their secret and the LogN client libraries generate one-time zero knowledge proof (ZKP) from the secret
7. ZKP is forwarded to NuID's service API along with txid associated to username
8. The service API uses txid to retrieve zk parameters from ledger
9. Verify ZKP against reference parameters
10. Authentication result sent to relying party

What is zero knowledge cryptography?

A zero knowledge proof (ZKP) is a cryptographic protocol which allows an individual to mathematically prove they know a piece of information, without sharing that information or anything else about it. The ZKP NuID uses is based on the Schnorr protocol described in the Internet Engineering Task Force's RFC 8235.

How does it work?

During registration, the client device is used to generate "public reference parameters" from the user's authentication secret, such as a password or biometric. These parameters are non-sensitive and can be shared openly, much like a public key. The reference parameters are immutably stored on a distributed ledger, and used to challenge the user during authentication.

When the user attempts to login, they input their credential on their device, and a zero knowledge proof of the credential is generated and sent to NuID's API. By running the proving algorithm to compare the proof sent from the client with the reference parameters stored on the ledger, the user can be authenticated without ever revealing private authentication data.

Why does NuID use a distributed ledger?

The NuID service uses a distributed ledger to store the public reference parameters generated from authentication secrets. These parameters are non-sensitive and can be safely shared publicly.

A distributed ledger provides critical tamper-proof and decentralized properties that remove the need for a centralized database and point of failure. Additionally, a public distributed ledger enables users to bring their own identity to multiple services, empowering users to own their own digital identities.

Deployment

The LogN client libraries simplify interacting with NuID's API, and can be integrated into almost any authentication flow. We provide lightweight client libraries for user-facing web and native applications.

The LogN SDK was designed to work as a thin layer within your IAM stack for customer or employee login. We offer our API as a fully-managed service on cloud infrastructure.

- ✓ **Immutable storage with no single point of failure**
- ✓ **Portability between services**
- ✓ **User ownership and privacy-by-design**