ZKX HELIX™

SUPERIOR TECHNOLOGY. SUPERIOR SECURITY.

HELIX KEY BENEFITS

- Enables the definition and enforcement of zero-trust access policies
- Polices are enforced by patented MFA to maximize operational security
- Enhanced security ties a user to a specific device
- User sessions are continually authenticated in the background
- Phishing-resistant: private keys cannot be stolen

INTRODUCTION

Authentication technology has desperately needed an overhaul for almost a decade now. Our overreliance on inferior technology, manifested by inferior solutions, has resulted in continual, predictable, and ever-repeating exploitations of our cyber infrastructure that increasingly harm individual dependent users and decreasingly impact the bottom-line of those organizations that peddle and mismanage these dated engines of failure.

ZKX Helix solves these problems. ZKX Helix is comprised of several patented technologies, totally unique to the Helix environment. These Helix-exclusive technologies enhance password authentication, conceal authentication credential data from mischief and snooping, and minimize the new and emerging attack surfaces for digital identity and authentication.

MAXIMIZE SECURITY WITH USER-DEVICE PAIRING

Helix achieves superior security by employing a unique chain of hashing algorithms combined with a traditional interactive zero-knowledge proof (ZKP) of identity. In short, when a user enrolls in the Helix MFA platform, a series of private keys are generated and associated with both that user and the specific device they're enrolling with. The corresponding public keys (the mates to the private keys) are sent to their organization's credential storage server (or platform). These public keys are just that – public – and pose no risk to the enrolled user (or the organization) if they were ever to become leaked or otherwise compromised.

When an enrolled user needs to satisfy a certain access policy to gain access to a network resource (i.e., authenticate themselves and prove their claimed identity) — Helix will ensure that that user not only produces the correct authentication credentials (e.g., PIN, password, NFC token, etc.) but also that they do so with the correct device. Assuming that the user/device pair passes all of Helix's checks, they will be deemed in accordance with the policy at hand and granted access to their requested resource.

CONTINUAL AND TRANSPARENT AUTHENTICATION

Once authenticated with with Helix, the software begins an automated, continual session for that user — a cryptographically rigid attestation of that user's (and their device's) verified identity. As the user navigates the network, they will no longer be burdened with constant prompts to reauthenticate as they bounce from fileshares to e-mail to intranet to anything else. Instead, transparent, automatic zero-knowledge authentication transactions are occurring with each privileged action they take, simultaneously ensuring that the user isn't constantly bombarded with interruptions of their process in the name of "cybersecurity" while maintaining the breadth of their lateral movement – making sure they're only getting access to what they need and nothing more.

REVERSE APPLICATION OF MASKS

When a user signs up with the Helix platform, numerous private keys are created for them and associated with their user identity and their device identity. As a user registers their authentication credentials with Helix, a standardized hashing algorithm intakes data related to each individual authentication credential. This credential data is used as input for the hashing algorithm, which results in totally unique, fixed-length outputs – one output for each input. These outputs (called "hashes") are then delimited across an arbitrary boundary.

Each delimitation of each individual hash is then resupplied to the same generating hash function, resulting in a linear increase in the volume of hashed strings that result from a single credential. Each of these double-hashed, credential-derived threads of data is then used to mathematically manipulate (i.e., "mask") a single of the user's (and device's) private keys, creating a cryptographically inert object – an object whose derivation stemmed from private data, but is not inherently private itself – called a key fragment.

Because this process derived multiple key fragments from a single authentication credential, multiple private keys can be mapped to (i.e. masked by) a single authentication credential. The key fragments are then stored on the user's device. Since the key fragments are cryptographically inert, none of the actual private keys are ever at risk of leaking or exposure, even if the user's device is stolen, compromised, or infected with malware.

When the user (on their same device) needs to authenticate themselves, they supply the same credential to Helix, which then generates the same arbitrary number of double-hashed outputs, which are then mathematically applied to the key fragments in a manner reverse of how they were applied to the raw private keys. This reverse application of the masks to the key fragments reconstructs the original private keys, which are then used to sign ZKP transactions before ultimately being destroyed, leaving again just the key fragments stored on the device.

A PRACTICAL EXAMPLE OF THIS UNIQUE ALGORITHM

Suppose that a user is enrolling in Helix using just their password. The administrator responsible for configuring this instance of Helix has determined that each authentication credential will be mapped to eight private keys, each 256-bits in length. As the user enrolls and registers their password with Helix, the password is used as input to a standardized hash function. If this hash function's output is 256 bits long, the Helix algorithm can bucketize this one 256-bit output into eight 32-bit long outputs. Each of these eight delimitations are resupplied to the same hash function now as inputs, resulting in eight 256-bit long totally unique hash outputs. This enables Helix to mathematically transform the eight private keys into eight key fragments, which are then stored on the user's device.

Later, when this user needs to authenticate (on the same device) they will be challenged on their knowledge or possession of the eight private keys that correspond to eight public keys, held by the server responsible for issuing authentication challenges. However, the user does not possess the private keys, only fragments produced by the keys. The user, when prompted to do so, inputs their password to Helix, which generates the same eight double-hashed threads of data (known as "masks"). These masks are now applied to the key fragments in a fashion reverse to how they were applied to the raw private keys. This reverse application of mask to key fragment results in the original private key, which is then used to sign the issued authentication challenge (a ZKP challenge) and is then subsequently destroyed, leaving only the cryptographically inert key fragment on the user's device.

IMMUNE TO CONVENTIONAL CYBERATTACKS

This patented method of mapping an arbitrary number of private keys to a single authentication is incredibly powerful, making it impossible for an adversary to exploit common attack avenues.

- Helix totally prevents private keys from being compromised, as they are not stored on either server nor device.
- Helix MFA is extremely phishing-resistant, as an adversary with just the user's password is effectively neutered without the user's specific device.
- Helix exponentially increases the work burden required for an adversary to successfully cheat an authentication transaction (as what is traditionally a one-to-one relationship – I have one password and my organization has one hash to verify it; now becomes a one-to-n relationship – I have one password and my organization has n keys to verify it). In order to successfully cheat this system, an adversary must successfully guess or spoof every single private key mapped to the user's password – not just the password (or its hash) itself.

These examples are only a fraction of the protection offered by Helix, as its core driving force of the ZKP affords users and organizations even more assurances that their identity and credential data are safe and uncompromisable.

ZERO KNOWLEDGE PROOFS

Zero-Knowledge Proofs (ZKPs) are not just a buzzword, but a designation. A proof of knowledge can only be designated as a true ZKP if it meets all of three criteria:

- **Completeness:** If a statement is true, one can always be convinced it is true
- **Soundness:** If a statement is false, one will never be convinced it is true
- **Zero-Knowledge:** No single bit (1 or 0) of information is revealed about the statement, only its veracity

If any proof of knowledge does not meet all three of these criteria, it cannot be considered a zero-knowledge proof.

The protection offered by ZKPs ensure that no entity (not even the authentication server that is issuing authentication challenges and verifying the legitimacy of user responses) can ever become privy to what the user's private key data actually is – only if it is legitimate or not. This applies as well to adversaries that may be sniffing the communication channels between client and server in an attempt to learn meaningful authentication secrets from unprotected data streams.

Furthermore, even if an adversary is impersonating the Helix authentication server itself, they still do not learn any single bit of data about the substance of the user's private keys – all due to the protection of ZKPs. This affords users of the Helix system unrivaled levels of security, privacy, and data custody.

ZKPs have different protocols (or rules) by which they operate – specific cadences of information flow that dictate exactly how information can be verified under the protections of zero-knowledge. One side effect of the ZKPs employed by Helix is that they are arbitrarily repeatable (i.e., they iterate) and can be used to calculate a precise measure of confidence in a statement's claimed truthfulness. This measure of confidence (i.e., "confidence score") is a function of how many private keys are challenged and how many iterations of ZKP are conducted.

For example, a user that authenticates in Helix with a store of ten private keys (which themselves are the composite of the user authentication credential(s) and the device's key fragments) and five iterations of ZKP achieves the same level of calculated confidence as a user that authenticates with five keys over ten ZKP rounds. While five, ten, twenty, or even more iterations of an authentication protocol may sound costly in terms of bandwidth consumption, Helix transactions on average consume 1.5 kB data to achieve "twelve-nines" (99.9999999999%) of confidence in any given user/device combination.

ACCESS POLICIES

In Helix, network administrators are able to define access policies on the basis of three main parameters that Helix uses to drive its security functions: device identity, authentication artifact, and calculated confidence. By combining these three adjustable and customizable parameters with the technical security aspects of the Helix protocol and system, Helix provides users, administrators, and organizations writ-large with true next-generation zero-trust security that deploys wherever it is needed and dutifully supports the requirements placed on it. From resource access management in the enterprise, to merging disparate zero-trust networks at the disconnected edge – Helix technologies enhance any variety of network operations with a higher, more effective standard of cybersecurity.

Technology that is safe, simple, and superior. Helix is security how it should be.