Lodestone

Technical Whitepaper

Cryptographic proof of presence for the physical world

Version 1.0
February 2026
Author: Jerimiah Ham, Groundspeak
Contact: jerimiah@geocaching.com

Table of Contents

1. Executive Summary

Lodestone is a secure NFC verification system that provides cryptographic proof of presence—verifiable evidence that a specific person physically interacted with a specific tag, captured at a specific point in that tag's cryptographic counter sequence. The system is designed for scenarios where internet connectivity cannot be guaranteed, enabling verification to occur hours, days, or even months after the physical interaction.

At its core, Lodestone leverages NXP's NTAG 424 DNA secure NFC tags, which provide hardware-level AES-128 encryption and tamper-evident rolling counters. What distinguishes Lodestone from simple "tap to verify" systems is its bidirectional security model: the user's mobile device writes a unique, user-specific one-time password (HOTP) to the tag, and the tag's cryptographic engine encrypts this payload along with its own authentication data before the device reads it back. This creates a verification bundle that only the server can decrypt and validate—proving both that the tag is genuine and that a specific user interacted with it at a specific point in its counter sequence.

Originally developed by Groundspeak to address the unique constraints of geocaching—where participants seek hidden containers in locations ranging from urban parks to remote wilderness—Lodestone's architecture applies broadly to any use case requiring tamper-proof, offline-capable proof of presence: asset tracking, supply chain verification, field service confirmation, or access logging in connectivity-challenged environments.

Key Highlights

• Offline-first architecture: Users can interact with tags for weeks or months before syncing, with full cryptographic integrity preserved

• Bidirectional authentication: Both the tag's identity and the user's interaction are cryptographically bound together

• Hardware-backed security: NTAG 424 DNA's secure element provides AES-128 encryption and non-resettable counters

• Anti-harvesting protection: One-time passwords and rolling counters prevent collecting tag data for later redistribution

2. Introduction

2.1 Background

Physical presence is one of the most fundamental human experiences, yet proving it digitally has remained surprisingly difficult. As more of life becomes mediated through screens and networks, the ability to reliably answer a simple question—"Was this person actually there?"—has become increasingly valuable.

Most existing approaches to presence verification assume continuous network connectivity. Real-time GPS check-ins, server-validated QR scans, and Bluetooth beacon systems all depend on the device communicating with a backend at the moment of interaction. This assumption holds in retail stores and airport lounges, but fails in exactly the environments where presence is most meaningful: hiking trails, remote landmarks, international travel, and the countless places where cellular coverage is unreliable or nonexistent.

Approaches that work offline tend to sacrifice security for convenience. A photograph can be shared. A QR code can be screenshotted and texted to a friend. GPS coordinates can be spoofed. Even basic NFC tags can be cloned or their data harvested and redistributed. These methods capture intent to verify but provide no cryptographic guarantee that a specific individual was physically present.

The challenge intensifies when verification must be deferred—when the proof of presence needs to remain valid and tamper-evident not just for seconds, but for days or weeks until connectivity is restored. This is the environment Lodestone was designed to address.

2.2 Problem Statement

Geocaching, the worldwide outdoor treasure-hunting game operated by Groundspeak, presents a particularly demanding proof-of-presence challenge. Over three million active players seek hidden containers across every continent, in locations ranging from urban parks to remote wilderness. A successful "find" has traditionally been recorded through a simple ritual: sign the physical logbook inside the cache, then log the find online later.

This honor-based system has served the community well for over two decades, but it inherently limits what's possible. Premium experiences, competitive events, achievement systems, and integrated rewards all require a level of verification trust that self-reporting cannot provide. The question becomes: how do you verify that someone was physically present at a cache location, potentially weeks before they have internet access, without trusting the user's device or the user's word?

Existing secure NFC technologies solve part of this problem. Tags like the NXP NTAG 424 DNA can cryptographically prove their own authenticity—the server can confirm that a scan came from a genuine tag, not a clone. But tag authentication alone leaves a critical gap: it proves the tag is real, but not who interacted with it. A single person could scan a tag and share the

resulting data with others. A network of cheaters could harvest tag responses and distribute them. The tag doesn't know or care who's holding the phone.

The missing piece is bidirectional authentication—a system where the user's identity is cryptographically bound to the tag's response, creating a verification bundle that proves both parties were present for that specific interaction.

2.3 Solution Overview

Lodestone closes this gap through a novel two-phase NFC interaction that leverages the NTAG 424 DNA's secure data messaging (SDM) capabilities in an unconventional way.

In the first phase, the user's device authenticates to the tag and writes a small payload containing a user-specific HOTP (HMAC-based One-Time Password). This HOTP is generated from a secret key stored in the device's hardware security module (Android Keystore or iOS Secure Enclave), meaning it cannot be extracted or shared—it can only be generated by that specific device.

In the second phase, the device reads the tag's SUN (Secure Unique NFC) message. Critically, the NTAG 424 DNA's SDM engine encrypts the file data—including the HOTP payload just written—using AES-128 with keys that only the tag and the verification server possess. The tag also increments its internal counter (which cannot be reset or rolled back) and generates a cryptographic MAC over the entire response.

The result is a self-contained verification bundle: encrypted proof of which tag was scanned, at what counter value, with which user's HOTP embedded inside. This bundle can be stored on the device indefinitely and transmitted to the server whenever connectivity is available. The server decrypts the payload, validates the tag's authenticity via the MAC, confirms the HOTP against the registered user's key, and checks that the counter value hasn't been used before.

No real-time connectivity required. No trust placed in the user's device after the initial key registration. No possibility of harvesting tag data for redistribution, because each response is bound to a specific user's one-time password that only their device could have generated.

The following sections detail the technical architecture that makes this possible.

3. Technical Architecture

3.1 System Overview

Lodestone consists of four primary components working together to create a complete proof-of-presence verification chain:

1. NFC Tags: NXP NTAG 424 DNA secure NFC tags, provisioned with unique cryptographic keys and configured for Secure Dynamic Messaging (SDM)

2. Mobile Application: Native iOS and Android applications that handle NFC communication, HOTP generation, and offline storage of verification bundles

3. Device Security Module: Platform-specific hardware security (Android Keystore or iOS Secure Enclave) that stores user HOTP keys in tamper-resistant silicon

4. Verification Server: Backend service that holds tag encryption keys, validates cryptographic proofs, and maintains the authoritative record of verified interactions

The system is designed around a principle of minimal trust: the mobile application is treated as an untrusted intermediary. It facilitates communication between the secure tag and the secure server, but cannot forge, replay, or redistribute valid verification bundles. All cryptographic operations that matter happen either in hardware-backed secure storage (on the mobile device) or on components the user cannot access (the tag's secure element and the server).

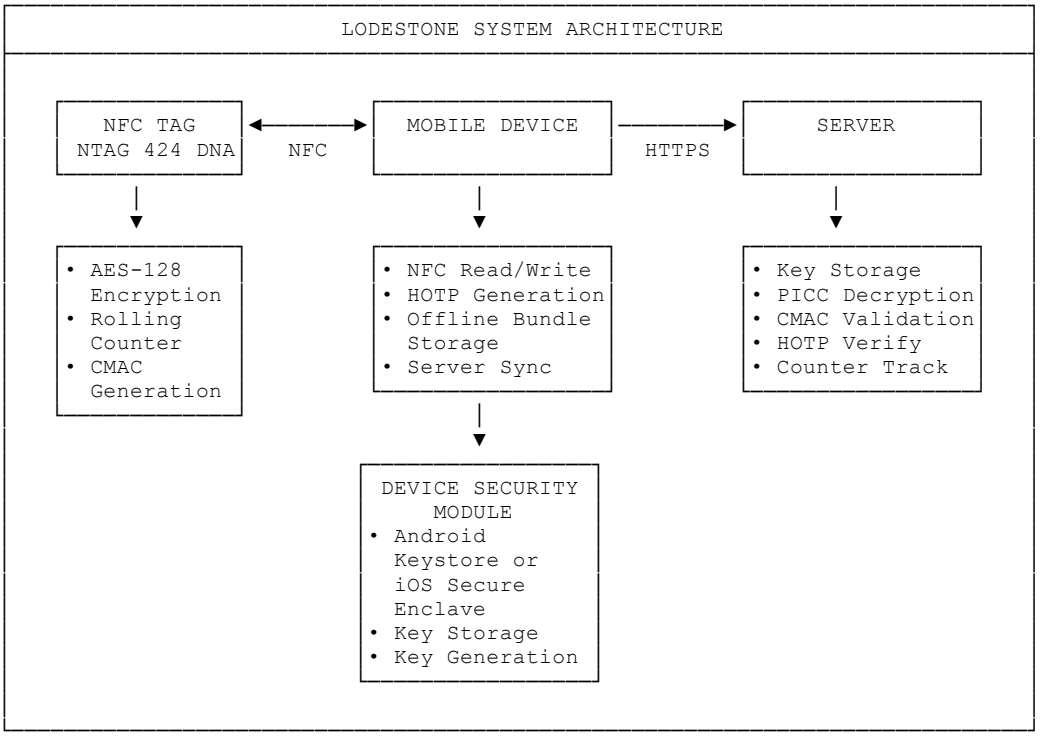


Diagram description for graphic designer: A four-component architecture diagram showing the NFC Tag, Mobile Device, Device Security Module, and Server. The NFC Tag connects bidirectionally to the Mobile Device via NFC protocol. The Mobile Device connects to the Server via HTTPS (this connection is dashed or lighter to indicate it's deferred/async). The Device Security Module is shown as a secure subsystem within or beneath the Mobile Device, connected with a secure/trusted link. Each component should list its key responsibilities as bullet points. Color coding suggestion: hardware-secured components (Tag, Device Security Module, Server) in one color family; the untrusted intermediary (Mobile Device application layer) in a neutral color.

3.2 Core Components

3.2.1 NFC Tag (NTAG 424 DNA)

The NXP NTAG 424 DNA serves as the physical anchor for proof of presence. Each tag contains a secure element that performs AES-128 encryption, maintains a non-resettable counter, and generates cryptographic message authentication codes (CMACs).

Key Architecture

Each tag is provisioned with a set of cryptographic keys serving distinct purposes:

Key Purpose	Function	Diversification
Master Key	Tag authentication and administrative operations	Unique per tag (derived from tag UID)
File Storage Write Key	Future capability for tag-owner-written data	Unique per tag (derived from tag UID)
NDEF URL Write Key	Controls modification of the URL template	Static across tags
Primary Encryption Key	Encrypts PICC data and file data in SUN messages	Static across tags
CMAC Key	Generates authentication MAC over SUN messages	Unique per tag (derived from tag UID)

The diversified keys ensure that compromising one tag does not expose the credentials needed to attack others. The static encryption key enables the server to decrypt the PICC data from any tag to extract the UID, which is then used to derive the tag-specific CMAC key for validation.

Secure Dynamic Messaging (SDM)

The tag is configured to use NXP's Secure Dynamic Messaging feature in an unconventional manner. Standard SDM use cases involve the tag generating a dynamic URL containing encrypted metadata about itself—useful for anti-counterfeiting and tap-counting applications.

Lodestone extends this by writing user-generated data to the tag's file storage area of the URL template immediately before reading the SUN message. The SDM engine encrypts this file data

alongside the standard PICC metadata, binding the user's payload to the tag's cryptographic response. This transforms the tag from a one-way authentication token into a bidirectional verification device.

Rolling Counter

The NTAG 424 DNA maintains an internal counter that increments with each successful SDM read operation. This counter cannot be reset or rolled back. It provides replay protection: even if an attacker captures a valid SUN message, that specific counter value can never be reused.

3.2.2 Mobile Application

The mobile application orchestrates the proof-of-presence interaction and manages offline storage until server synchronization is possible.

NFC Communication

The application performs a two-phase NFC transaction:

1. Write Phase: Authenticates to the tag using the NDEF URL write key and writes the complete URL template, including a file data section containing the user's HOTP and other metadata

2. Read Phase: Reads the tag's SUN message, which now contains the written payload encrypted by the tag's secure element along with the PICC data, counter value, and CMAC

Writing the complete URL on every interaction serves as a defacement mitigation strategy—if a malicious actor has modified the tag's URL template, the legitimate application overwrites it with the correct template.

Payload Construction

The file data payload written to the tag includes:

• HOTP Code: A one-time password computed from a derivative of the device's hardware-secured public key

• HOTP Index: The counter value used to generate this specific HOTP, enabling precise server-side validation

• Key Identifier Hint: A short prefix indicating which registered key the server should use for validation

• User Identifier: Non-sensitive identifier linking the interaction to a registered user

• Device Context: Device make, model, operating system version, and application version for diagnostic and analytics purposes

• Timestamp: Device-local time of interaction (treated as untrusted metadata, not used for verification)

This metadata is explicitly designed to contain no personally identifiable information (PII). The payload structure may evolve as operational requirements change.

Offline Storage

Successfully captured verification bundles are stored locally on the device. Each bundle is self-contained and includes all information the server needs for validation. Bundles can be transmitted immediately if connectivity is available, or queued indefinitely until the next successful sync.

3.2.3 Device Security Module

HOTP key security is critical to the system's integrity. If a user's HOTP credential could be extracted from their device and shared, the bidirectional authentication property would collapse—anyone with the credential could generate valid HOTPs.

Lodestone uses an asymmetric keypair architecture with platform hardware security modules:

• Android: Android Keystore, with preference for StrongBox (dedicated secure element) when available, falling back to TEE (Trusted Execution Environment) on devices without StrongBox

• iOS: Secure Enclave, Apple's hardware-isolated security coprocessor

The secure element generates and stores the private key of an asymmetric keypair. This private key never leaves the secure hardware in plaintext form. The application can request the public key and use a derivative of it as the seed for HOTP computation, but the underlying private key remains inaccessible—even to the application itself, even to a user intentionally trying to extract it.

The HOTP counter index is maintained in secure application storage and increments with each tag interaction. This index is written to the tag alongside the HOTP code, enabling the server to validate against the exact index used rather than searching a window of possibilities.

Key Registration

When a user sets up a new device, the application triggers keypair generation within the hardware security module. Registration with the server includes a verification step to prove the keypair is genuine and accessible:

1. The server sends a challenge payload to the application

2. The application requests the secure element to sign (encrypt) this payload using the private key

3. The signed response is returned to the server

4. The server verifies the signature using the registered public key

This challenge-response verification confirms that the public key corresponds to a real private key held in a hardware security module—not a fabricated key or one extracted from another

device. Once verified, the server stores the public key material for future HOTP validation.The server may periodically perform this verification step to ensure that the private key is still accessible (present on the device) and the keypair is still valid.

The system can support multiple devices per user account, enabling users to verify presence from different devices or transition between devices without losing verification capability. It can also support multiple users on a single device, keeping each user's keys separate.

3.2.4 Verification Server

The verification server is the ultimate authority on proof of presence. It holds the tag encryption keys, validates cryptographic proofs, and maintains the canonical record of all verified interactions.

Verification Pipeline

When the server receives a verification bundle, it performs the following validation sequence:

1. PICC Data Decryption: Decrypt the PICC data using the primary encryption key to extract the tag's UID and counter value

2. CMAC Key Derivation: Using the extracted UID, derive the tag-specific CMAC key

3. CMAC Validation: Verify the message authentication code to confirm the SUN message originated from a genuine NTAG 424 DNA tag and was not modified in transit

4. File Data Decryption: Decrypt the file data to recover the user's HOTP and metadata payload

5. HOTP Verification: Using the key identifier hint, locate the user's registered HOTP key and verify the one-time password is valid for the provided index

6. Counter Validation: Confirm the tag's counter value has not been previously recorded for this tag, preventing replay attacks

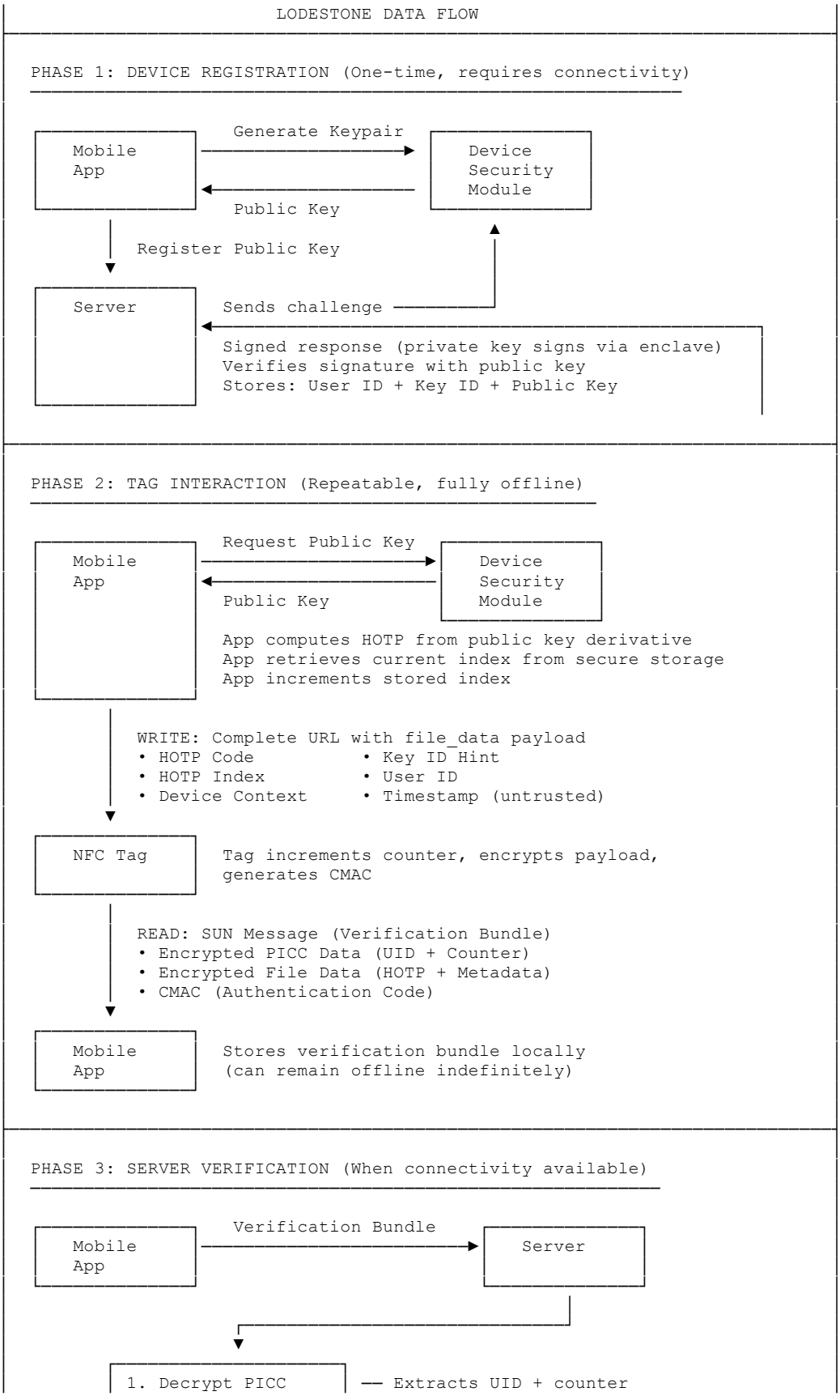
7. Record Creation: Store the verified interaction with the tag ID, user ID, tag counter value, and server-side timestamp

Counter Window Management

Because users may operate offline for extended periods and sync interactions out of chronological order, the server accepts counter values that have not been previously used for each tag. This accommodates legitimate usage patterns while still detecting obvious replay attempts.

3.3 Data Flow

The complete verification flow consists of three distinct phases: device registration (one-time setup), tag interaction (repeatable, offline-capable), and server verification (deferred until connectivity available).


LODESTONE DATA FLOW
PHASE 1: DEVICE REGISTRATION (One-time, requires connectivity)
Mobile App
Generate Keypair
Device Security Module
Public Key
Register Public Key
Server
Sends challenge
Signed response (private key signs via enclave)
Verifies signature with public key
Stores: User ID + Key ID + Public Key
PHASE 2: TAG INTERACTION (Repeatable, fully offline)
Mobile App
Request Public Key
Device Security Module
Public Key
App computes HOTP from public key derivative
App retrieves current index from secure storage
App increments stored index
WRITE: Complete URL with file_data payload
• HOTP Code
• HOTP Index
• Device Context
• Key ID Hint
• User ID
• Timestamp (untrusted)
NFC Tag
Tag increments counter, encrypts payload, generates CMAC
READ: SUN Message (Verification Bundle)
• Encrypted PICC Data (UID + Counter)
• Encrypted File Data (HOTP + Metadata)
• CMAC (Authentication Code)
Mobile App
Stores verification bundle locally
(can remain offline indefinitely)
PHASE 3: SERVER VERIFICATION (When connectivity available)
Mobile App
Verification Bundle
Server
1. Decrypt PICC
Extracts UID + counter

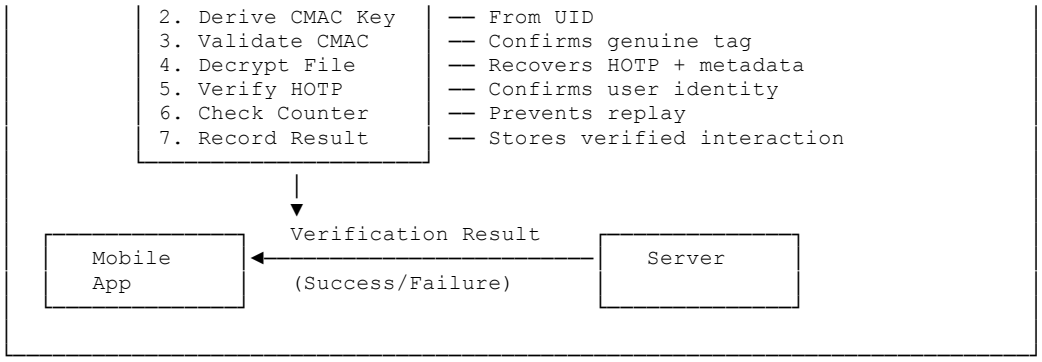


Diagram description for graphic designer: A three-panel vertical flow diagram showing the complete Lodestone data lifecycle. Panel 1 - Device Registration: Shows the mobile app requesting key generation from the Device Security Module (secure enclave icon), receiving a key reference back, then registering with the Server. This panel should be visually distinct as "one-time setup" and indicate that connectivity is required. Panel 2 - Tag Interaction: The largest panel, showing a circular or bidirectional flow. The app requests an HOTP from the security module, constructs a payload (show the payload contents in a callout box), writes to the NFC tag, the tag processes (counter increment, encryption, CMAC generation), and the app reads back the verification bundle (show bundle contents in a callout box). This panel should prominently indicate "NO CONNECTIVITY REQUIRED" and show the bundle being stored locally. Panel 3 - Server Verification: Shows the bundle being transmitted to the server when connectivity is available. The server's validation pipeline should be shown as a sequential checklist. The result flows back to the app. Include a visual indication that this phase can happen minutes, hours, days, or weeks after Phase 2.

4. Implementation Details

4.1 Technology Stack

Category	Technology	Purpose
NFC Hardware	NXP NTAG 424 DNA	Secure NFC tag with AES-128 encryption, rolling counter, and Secure Dynamic Messaging
Mobile (Android)	Kotlin, Android Keystore	Native application with hardware-backed key storage
Mobile (iOS)	Swift, Secure Enclave	Native application with hardware-backed key storage
Mobile (Integration)	React Native Native Module	Cross-platform integration option for real-time verification scenarios
Server Runtime	Java	Backend verification service
Cryptographic Foundation	johnnyb/ntag424-java (MIT license, Jonathan Bartlett)	NTAG 424 DNA cryptographic operations (SUN decryption, CMAC validation, LRP support)
Database	Relational database (ACID-compliant)	Transactional storage for counter replay protection and verification records

Platform Considerations

Native mobile applications (Android/Kotlin and iOS/Swift) provide full Lodestone functionality, including hardware-secured HOTP key generation and offline verification bundle storage. These platforms are required for complete offline-capable proof of presence.

A React Native native module is available for integration scenarios where real-time server connectivity can be assumed. This module handles NFC communication and captures the tag's SUN message, but omits the HOTP mechanism entirely. Verification relies solely on the tag's built-in counter to produce a unique SUN URL, with the server processing the interaction immediately using a counter window for replay protection. This approach enables straightforward integration for partners with existing React Native applications while accepting the constraint that verification must occur in real-time with network connectivity and lacks the user-binding properties of the full HOTP flow.

Database Requirements

The verification server requires a fully ACID-compliant relational database to provide the transactional guarantees essential for counter replay protection. When multiple verification bundles arrive simultaneously—potentially including replay attempts—the database must guarantee that counter value checks and insertions are atomic. Without strong transactional semantics, race conditions could allow duplicate counter values to be accepted.

4.2 Key Algorithms

4.2.1 HOTP Key Generation and Registration

Lodestone uses an asymmetric keypair architecture rather than traditional symmetric HOTP shared secrets. This design ensures that no secret material is ever transmitted during registration.

Keypair Generation

The device's hardware security module generates an ECDSA keypair. The private key is created within and never leaves the secure element. The application can request cryptographic operations using the private key (such as signing) but cannot extract the key material itself.

Proof of Ownership

During registration, the device must prove it actually possesses the private key corresponding to the public key being registered:

1. The application constructs a payload containing the player ID and current timestamp

2. The secure element signs this payload using SHA256withECDSA

3. The signature and public key are transmitted to the server

4. The server verifies the signature using the provided public key

This challenge-response proves the public key is backed by a real, accessible private key in a hardware security module—not a fabricated key or one copied from another device.

HOTP Seed Derivation

Rather than using the public key directly as an HOTP seed, both the server and mobile application independently derive a shared seed:

```
HOTP_SEED = SHA-256(public_key || player_id)
```

This derivation binds the HOTP seed to both the cryptographic keypair and the specific player identity, preventing key reuse across accounts.

4.2.2 HOTP Generation

HOTP code generation follows RFC 4226 with a modernized hash function:

• Algorithm: HMAC-SHA256 (rather than the RFC's original HMAC-SHA1)

• Input: The derived 32-byte seed and the current counter index

• Output: 6-digit numeric code

• Counter Management: The application maintains the counter index in secure storage, incrementing after each use. The index is transmitted alongside the HOTP code, enabling precise server-side validation.

The use of HMAC-SHA256 provides a larger security margin than SHA-1, aligning with current cryptographic best practices while remaining compatible with the RFC 4226 framework.

4.2.3 Secure Dynamic Messaging (SDM)

The NTAG 424 DNA's SDM engine performs encryption and authentication using AES-128:

PICC Data Encryption

The tag encrypts its own metadata (UID and counter value) using the Primary Encryption Key. This produces a ciphertext blob that only the server can decrypt, preventing observers from tracking tag identity or counter progression.

File Data Encryption

User-written payload data stored in the tag's file area is encrypted by the same SDM engine using the same Primary Encryption Key. This is the mechanism that binds the user's HOTP to the tag's cryptographic response—the payload written by the application becomes part of the tag's authenticated, encrypted output.

CMAC Validation

The tag generates a Cipher-based Message Authentication Code (CMAC) over the complete SUN message using the CMAC Key. This MAC serves two purposes:

1. Authenticity: Proves the message originated from a genuine NTAG 424 DNA tag

2. Integrity: Detects any modification to the message contents after generation

Because the CMAC Key is diversified per tag (derived from the tag's UID), the server must first decrypt the PICC data to extract the UID, then derive the tag-specific CMAC key, and finally validate the CMAC. A failed CMAC check indicates either a counterfeit tag, message tampering, or an unknown tag UID, and the verification is rejected without further processing.

4.3 Security Considerations

4.3.1 Leakage Resilient Primitive (LRP) Authentication

Lodestone tags are configured to use NXP's Leakage Resilient Primitive (LRP) mode for authentication rather than standard AES. LRP is designed to resist side-channel attacks—techniques where an attacker analyzes power consumption, electromagnetic emissions, or timing variations to extract cryptographic keys.

While side-channel attacks against NFC tags require sophisticated equipment and physical proximity, LRP activation has no performance cost or functional trade-off. Given that LRP is a one-way upgrade (tags cannot be reverted to standard AES mode once LRP is enabled), Lodestone enables this protection during provisioning as a defense-in-depth measure.

4.3.2 Random UID

NTAG 424 DNA tags support a Random UID feature that returns a randomized identifier during the NFC anti-collision phase, rather than the tag's true UID. This prevents passive tracking of tag movements by observers who might monitor NFC traffic without participating in the SDM protocol.

The tag's true UID is still available through the encrypted PICC data in SUN messages, where it can only be recovered by the verification server. Lodestone enables Random UID during provisioning to protect tag location privacy while maintaining full verification capability.

4.3.3 Key Diversification

Each tag's Master Key, File Storage Write Key, and CMAC Key are diversified—derived from a master secret combined with the tag's unique UID. This ensures that:

• Compromising one tag's keys does not expose credentials for any other tag

• An attacker with access to a single tag cannot derive keys for the broader deployment

• Key extraction attacks (however difficult) remain isolated to individual tags

The Primary Encryption Keys used for SDM file write permission (NDEF URL) and PICC data encryption remain static across tags. This enables the server to decrypt the PICC data (which contains the UID) before deriving the tag-specific CMAC key for validation.

4.3.4 Stealth Mode

Lodestone supports a "stealth mode" configuration that makes tags invisible to casual NFC interactions while remaining fully functional for dedicated applications.

Mechanism

Standard NFC tag discovery relies on reading the Capability Container (CC) file, which advertises the tag's capabilities and data structure. Stealth mode restricts read access to this file, causing the tag to appear non-responsive to generic NFC readers and operating system tag-discovery features.

An application with knowledge of the tag's structure can bypass the standard discovery process and communicate directly with the tag using ISO 14443-4 commands. The tag remains fully functional—it simply doesn't announce itself.

Use Cases

Stealth mode is valuable when tags should only be discoverable by authorized applications:

• Anti-tampering: Prevents casual users from discovering and potentially defacing tags

• Controlled experiences: Ensures interactions only occur through the official application

• Reduced interference: Eliminates accidental triggers from phones brushing past tags

Tags can be toggled between stealth and normal modes by an authorized provisioning application, allowing deployment flexibility.

4.3.5 URL Overwrite as Defacement Mitigation

The NDEF URL write key is considered semi-trusted—it must be known to the application to enable writing, which means it could potentially be extracted through reverse engineering. Rather than relying solely on write protection, Lodestone applications write the complete URL template on every interaction.

If a malicious actor has defaced a tag's URL (for example, redirecting it to a phishing site), the next legitimate interaction overwrites the defacement with the correct template before reading the SUN message. This self-healing behavior limits the impact window of URL defacement attacks.

4.3.6 Counter Replay Protection

The tag's rolling counter provides the foundation for replay protection, but server-side enforcement is equally critical. The verification server maintains a record of all counter values successfully verified for each tag.

Validation Rules

• Each counter value may only be accepted once per tag

• Counter values are checked atomically (database transaction) to prevent race conditions

• Out-of-order arrivals are permitted (supporting offline use with delayed sync)

• Extremely old counter values may trigger additional scrutiny or rejection, depending on deployment policy

Offline Tolerance

Because users may interact with many tags before syncing, and sync order is not guaranteed, the server cannot assume counter values arrive sequentially. The system accepts any previously-unseen counter value, relying on the cryptographic binding (tag UID + counter + user HOTP) to validate each interaction independently.

4.3.7 Hardware-Backed Key Isolation

The decision to store HOTP private keys exclusively in hardware security modules (Android Keystore/StrongBox, iOS Secure Enclave) provides defense against multiple threat vectors:

• Application compromise: Malware with access to application storage cannot extract keys

• Device backup extraction: Keys are marked non-exportable and excluded from backups

• User collusion: Even a user intentionally attempting to extract and share their key cannot do so

• Rooted/jailbroken devices: While hardware security guarantees are reduced on compromised devices, extraction remains non-trivial compared to software-only storage

This architectural choice accepts a trade-off: if a user loses access to their device, their HOTP key is unrecoverable. The system handles this through key rotation—users can register new devices, and the server tracks multiple active keys per account. In fact, regular key rotation (after every 1000 tag interactions) is standard practice.

5. Use Cases & Applications

5.1 Primary Use Case: Geocaching Verification

Geocaching—the worldwide outdoor treasure-hunting activity with over three million active participants—presents the defining use case for Lodestone. Players seek hidden containers ("geocaches") in locations ranging from urban parks to remote wilderness, traditionally logging their finds through physical logbooks and online self-reporting.

The Verification Gap

The honor-based system has sustained the geocaching community for over two decades, but it fundamentally limits what's possible:

• Premium cache experiences requiring verified completion

• Competitive events with standings based on verified finds

• Achievement systems tied to physical accomplishments

• Integration with rewards programs or sponsors

• Detection of systematic cheating that undermines community trust

Lodestone Integration

A Lodestone tag deployed inside or near a geocache container transforms a self-reported find into a cryptographically verified interaction:

1. The player taps the tag with the geocaching application

2. The application writes the player's HOTP and metadata, then captures the encrypted verification bundle

3. The bundle is stored locally until connectivity is available

4. Upon sync, the server verifies the tag's authenticity, the player's identity, and the uniqueness of this interaction

5. The verified find is recorded and credited to the player's account

This flow works identically whether the player syncs thirty seconds later on a city sidewalk or three weeks later after returning from a backcountry expedition.

5.2 Additional Applications

Lodestone's proof-of-presence architecture extends beyond geocaching to any scenario requiring tamper-proof verification of physical interaction:

• Adventure Lab Experiences: Location-based storytelling and discovery activities where verified presence at each waypoint advances the narrative

• Field Service Verification: Confirmation that technicians physically visited equipment locations, valuable for compliance and service-level agreements

• Supply Chain Checkpoints: Tamper-evident verification that goods passed through specific handling points

• Access Logging: Recording physical presence at secure or regulated locations without requiring network infrastructure on-site

• Event Attendance: Verified participation in physical events, check-in stations, or course completions

• Tourism and Cultural Sites: Authenticated visits to landmarks, museums, or heritage sites for gamification or rewards programs

The offline-first architecture is particularly valuable in scenarios where connectivity cannot be assumed: remote industrial sites, international travel, outdoor recreation, and developing regions with limited infrastructure.

5.3 Case Study: MEGA Event Pilot Program (Summer 2026)

Case Study: Large-Scale Field Validation

Context

Geocaching MEGA events are community gatherings that attract thousands of participants to a single location for a weekend of activities, workshops, and group caching excursions. These events represent an ideal testing environment for Lodestone: high participant volume, diverse device types, motivated users willing to provide feedback, and controlled deployment conditions.

Pilot Design

In summer 2026, Groundspeak will deploy Lodestone tags at select MEGA events integrated with the Adventure Lab application. This pilot uses the React Native native module, providing real-time verification without the full HOTP offline capability. The simplified deployment allows the team to validate:

• Technical Performance: NFC read/write reliability across hundreds of device models, tag durability in field conditions, server throughput under concentrated load

• User Experience: Tap interaction success rates, error messaging clarity, time-to-verification perception, accessibility considerations

• Operational Logistics: Tag provisioning workflows, placement strategies, damage and loss rates, staff training requirements

Expected Outcomes

The pilot will generate both quantitative metrics (success rates, latency distributions, error categorization) and qualitative feedback (user surveys, staff observations, community forum

discussion). This data will inform refinements before broader deployment and validate assumptions about real-world usage patterns at scale.

Path to Full Deployment

Insights from the MEGA event pilots will guide the transition to full offline-capable Lodestone integration in the flagship Geocaching application. This will bring cryptographic proof-of-presence to the game of geocaching, enabling new categories of verified experiences, premium cache types, competitive formats, and community features that were previously impossible under the honor-based system.

6. Performance & Benchmarks

6.1 Performance Metrics

Lodestone's performance characteristics span three domains: NFC interaction speed, server verification throughput, and offline storage efficiency.

NFC Interaction Timing

The complete Lodestone tap interaction—authenticate, write payload, read SUN message—must complete within platform-imposed time constraints. iOS in particular enforces strict NFC session timeouts (approximately 500ms for foreground operations).

Laboratory testing on iOS devices demonstrates the full read-write cycle completing in 115–148 milliseconds, well within platform limits and imperceptible to users as anything other than instantaneous.

Operation	Measured Duration	Platform Constraint
Full Lodestone interaction (iOS)	115–148 ms	~500 ms timeout
Tag authentication	~30 ms	—
Payload write	~40 ms	—
SUN message read	~45 ms	—

Note: Timing varies by device NFC hardware, tag positioning, and environmental factors. Values represent controlled laboratory conditions.

Server Verification

Server-side verification of a single bundle involves PICC decryption, CMAC key derivation, CMAC validation, file data decryption, HOTP verification, and database operations. Estimated performance targets for production deployment:

Metric	Estimate	Notes
Single verification latency	< 100 ms	End-to-end, excluding network transit
Expected sustained load	< 60 verifications/minute	Typical production usage
Burst capacity	1,000+ verifications/minute	Event scenarios with concentrated activity

Current geocaching usage patterns suggest sustained verification rates well below system capacity. The architecture supports significant growth without redesign.

Offline Storage

Each verification bundle is compact and self-contained:

Metric	Value
Bundle size	~500 bytes
Typical device capacity	10,000+ pending bundles
Sync payload (100 bundles)	~50 KB

Storage efficiency enables extended offline operation without meaningful impact on device resources.

Tag Durability

NTAG 424 DNA tags are designed for long-term deployment:

Metric	Specification
Data retention	50 years
Read/write endurance	200,000–500,000 cycles

These specifications exceed the practical lifetime of most deployment scenarios. A tag scanned 100 times per day would reach 200,000 cycles after approximately 5.5 years of continuous heavy use.

6.2 Comparative Analysis

Lodestone addresses a verification gap that existing approaches cannot fill. The following comparison illustrates why alternative methods fall short for offline-capable, tamper-proof proof of presence.

Approach	Offline	Tamper Resistant	User Binding	Replay Protected	Complexity
GPS Check-in	No	No	Yes	No	Low
QR Code Scan	No	No	No	No	Low
Photo Verification	Yes	No	Partial	No	Low
RFID (125kHz/13.56MHz)	Yes	No	No	No	Low
iButton (1-Wire)	Yes	No	No	No	Low
Basic NFC (NDEF)	Yes	No	No	No	Low
Secure NFC (SUN only)	Yes	Yes	No	Yes	Medium

Lodestone	Yes	Yes	Yes	Yes	Medium

GPS Check-in

GPS-based verification requires the device to report its coordinates to a server at the moment of interaction. This fails in areas without connectivity and is trivially spoofed using widely available GPS simulation tools. Location accuracy also degrades in urban canyons, dense forests, and indoor environments—exactly the places where interesting verification targets are often located.

QR Code Scan

QR codes are inexpensive and easy to deploy, but offer no security properties whatsoever. A QR code can be photographed, shared, posted online, or reproduced. One person scanning a code can distribute it to thousands. QR codes also require server connectivity at scan time to have any verification value.

Photo Verification

Requiring photographic evidence of presence adds friction and introduces subjective human review. Photos can be shared between users, taken by one person and submitted by another, or in some cases fabricated entirely. Photo verification also raises privacy concerns and accessibility issues.

RFID (125kHz and 13.56MHz)

Traditional RFID technologies are widely deployed in guard tour systems and access control. However, these legacy technologies were designed for convenience, not security. Modern cloning devices for 125kHz proximity cards (such as those using the EM4100 protocol) are inexpensive and widely available—a card can be duplicated in seconds. Even 13.56MHz MIFARE Classic cards, once considered secure, have well-documented vulnerabilities that enable cloning with commodity hardware.

For verification purposes, RFID shares the same fundamental weakness as QR codes: the data can be captured and reproduced, providing no assurance that the original token was present.

iButton (1-Wire)

iButton devices have been a mainstay of guard tour and time-and-attendance systems for decades. Their rugged stainless steel packaging provides physical durability, but offers no cryptographic protection. Standard iButton devices broadcast a static serial number that can be read by any compatible reader—and reproduced by programmable iButton emulators readily available online.

The guard tour industry has largely relied on the assumption that employees lack the technical sophistication or motivation to clone tokens. This assumption fails in scenarios with higher stakes or more motivated adversaries.

Basic NFC (NDEF)

Standard NFC tags storing static data (URLs, text, identifiers) offer no cryptographic protection. The data can be read and cloned to another tag, or simply recorded and replayed. Basic NFC tags are useful for convenience features but provide no verification integrity.

Secure NFC (SUN Message Only)

Tags like the NTAG 424 DNA with Secure Dynamic Messaging provide cryptographic proof that a genuine tag was scanned, with replay protection via the rolling counter. This is a significant improvement—the server can verify tag authenticity and detect replayed scans.

However, standard SUN implementation lacks user binding. The tag proves it was scanned, but not by whom. A single user could scan the tag, capture the SUN message, and share it with others. A coordinated group could harvest SUN messages from tags and redistribute them. The tag cannot distinguish between legitimate users and data mules.

Lodestone

Lodestone closes this gap by introducing bidirectional authentication. The user's device writes a hardware-secured HOTP to the tag before reading the SUN message. The tag's cryptographic engine encrypts this user-specific payload, binding it to the tag's response. The resulting verification bundle proves:

1. A genuine NTAG 424 DNA tag was involved (CMAC validation)

2. The interaction occurred at a specific point in the tag's counter sequence (replay protection)

3. A specific registered user's device generated the HOTP (user binding)

This combination of properties—offline capability, tamper resistance, user binding, and replay protection—is unique to Lodestone's architecture.

6.3 Scalability

Tag Deployment

Lodestone scales horizontally with tag deployment. Each tag operates independently with its own counter sequence and diversified keys. There is no coordination required between tags, no shared state to manage, and no limit on deployment density.

Server Architecture

The verification server is stateless with respect to individual verification requests—all necessary information is contained in the verification bundle. This enables horizontal scaling through standard load balancing. Database operations are the primary scaling constraint, addressed through:

• ACID-compliant transactions ensuring counter replay protection under concurrent load

• Index optimization for tag UID and counter value lookups

• Archival strategies for historical verification records

Offline Accumulation

The system is designed to tolerate arbitrary delays between tag interaction and server verification. Users returning from extended offline periods may sync hundreds of verification bundles simultaneously. The server processes these as independent verification requests, with no assumption about arrival order or timing.

7. Roadmap & Future Development

7.1 Current Status

The Lodestone platform foundation is operational and has been validated through laboratory testing and controlled field deployments.

Deployed Capabilities

• Core proof-of-presence verification pipeline: tag provisioning, mobile capture, server validation

• Secure tag interactions designed for outdoor reliability and extended offline operation

• Native mobile libraries for iOS and Android with hardware-backed HOTP key management

• React Native integration module for real-time verification scenarios

• Server infrastructure with ACID-compliant transaction processing for counter replay protection

Early integrations and experience pilots with select partners have established baseline performance characteristics and validated the end-to-end verification flow under field conditions.

7.2 Planned Development

Near-Term: Pilot Expansion (2026)

The immediate development focus centers on expanded real-world validation:

• Large-scale event pilots at geocaching MEGA events, testing system performance with hundreds of concurrent users across diverse device types

• Experience pattern development for Adventure Lab and location-based storytelling applications

• Operational tooling improvements: streamlined provisioning workflows, partner onboarding documentation, deployment best practices

• User experience refinement informed by quantitative metrics and qualitative feedback from pilot participants

Medium-Term: Application Expansion

Following successful pilot validation, development will extend Lodestone's reach to additional use cases:

• Return visit and collection mechanics: verification patterns that reward sustained engagement with locations over time

• Tourism and destination applications: city-wide trails, landmark verification, cultural heritage site integration

• Educational contexts: verified field trips, outdoor learning curricula, nature exploration programs

• Enhanced privacy patterns: verification techniques that minimize unnecessary user data exposure while maintaining cryptographic integrity

7.3 Research Directions

Active exploration is underway in several areas that may influence future platform capabilities:

• Multi-tag verification sequences for complex, multi-location experiences

• Extended offline tolerance patterns for extreme-duration expeditions

• Community-driven experience formats emerging from pilot feedback

• Integration frameworks for third-party platforms requiring verified presence capabilities

Development priorities will be informed by real-world usage patterns observed during pilot deployments rather than speculative feature planning. The platform will evolve deliberately, with new capabilities validated through controlled pilots before broad release.

8. Conclusion

Lodestone represents a novel approach to proof of presence: cryptographic verification that a specific individual physically interacted with a specific location, captured in a tamper-evident bundle that remains valid indefinitely without requiring network connectivity at the moment of interaction.

The system achieves this through an unconventional application of the NTAG 424 DNA's Secure Dynamic Messaging capabilities. By writing a hardware-secured, user-specific one-time password to the tag immediately before reading its encrypted response, Lodestone creates a bidirectional authentication binding that neither standard NFC approaches nor traditional secure tag implementations can provide.

Technical Differentiation

Where existing verification technologies force a choice between offline capability and security, Lodestone delivers both:

• Hardware-backed user binding: HOTP keys generated and stored in device secure elements (Android Keystore, iOS Secure Enclave) cannot be extracted or shared, even by the device owner

• Cryptographic tag authentication: AES-128 encryption with LRP protection and CMAC validation ensures tag genuineness and message integrity

• Deterministic replay protection: Non-resettable tag counters combined with server-side counter tracking prevent verification bundle reuse

• Indefinite offline tolerance: Self-contained verification bundles require no assumptions about sync timing or network availability

Practical Validation

Developed to address the demanding requirements of geocaching—where verification must work reliably in wilderness conditions, across extended offline periods, and at scale across a global community—Lodestone's architecture has been validated through laboratory testing and early field pilots. The Summer 2026 MEGA event pilot program will extend this validation to large-scale, real-world conditions.

Broader Applicability

While geocaching provides the proving ground, Lodestone's proof-of-presence architecture addresses a fundamental verification gap with applications across industries: field service confirmation, supply chain integrity, access logging, tourism engagement, and any scenario where tamper-proof evidence of physical presence creates value.

The combination of offline capability, cryptographic integrity, and user-specific binding—achieved through commodity secure NFC hardware and standard mobile platform

security features—positions Lodestone as a practical foundation for verified physical-world interactions.

Partnership & Licensing Inquiries

Groundspeak welcomes inquiries from organizations interested in:

• Pilot participation: Join upcoming pilots to evaluate Lodestone for your verification use cases

• Platform integration: Explore integration of Lodestone proof-of-presence into existing applications and workflows

• Licensing: Discuss licensing arrangements for deploying Lodestone technology in new domains

To learn more or begin a conversation, please contact:

Groundspeak, Inc.

Email: bizdev@geocaching.com

Website: https://www.geocaching.com

Appendix A: Glossary

Term	Definition
ACID	Atomicity, Consistency, Isolation, Durability—a set of properties guaranteeing reliable database transaction processing. Lodestone requires ACID compliance to ensure counter replay protection under concurrent verification requests.
AES	Advanced Encryption Standard—a symmetric block cipher used for encryption. The NTAG 424 DNA uses AES-128 (128-bit keys) for SDM encryption and authentication.
Android Keystore	Android's hardware-backed secure storage system for cryptographic keys. Keys stored in Keystore cannot be extracted from the device, even by the application that created them.
CMAC	Cipher-based Message Authentication Code—a cryptographic MAC algorithm based on a block cipher. The NTAG 424 DNA generates a CMAC over SUN messages to prove authenticity and integrity.
ECDSA	Elliptic Curve Digital Signature Algorithm—an asymmetric cryptographic algorithm used for digital signatures. Lodestone uses ECDSA keypairs for HOTP key registration and proof of device ownership.
HMAC	Hash-based Message Authentication Code—a mechanism for calculating a message authentication code using a cryptographic hash function. HOTP codes are generated using HMAC-SHA256.
HOTP	HMAC-based One-Time Password—a one-time password algorithm defined in RFC 4226. Lodestone uses HOTP to generate unique, verifiable codes bound to specific user devices.
iButton	A 1-Wire device manufactured by Maxim Integrated, packaged in a stainless steel enclosure. Commonly used in guard tour systems but vulnerable to cloning due to lack of cryptographic protection.
ISO 14443	The international standard for proximity contactless smart cards, defining physical characteristics, radio frequency interface, and transmission protocols. NTAG 424 DNA operates as an ISO 14443 Type 4 tag.
LRP	Leakage Resilient Primitive—an authentication mode available on NTAG 424 DNA that provides enhanced protection against side-channel attacks compared to standard AES authentication.
NDEF	NFC Data Exchange Format—a standardized data format for storing and exchanging information on NFC tags. Lodestone uses NDEF URL records configured for SDM.
NFC	Near Field Communication—a set of communication protocols enabling two electronic devices to communicate within approximately 4 cm of each other.
NTAG 424 DNA	An NFC tag IC manufactured by NXP Semiconductors featuring AES-128 encryption, Secure Dynamic Messaging, rolling counters, and cryptographic authentication. The hardware foundation of Lodestone.
PICC	Proximity Integrated Circuit Card—the ISO 14443 term for a contactless smart card or tag. In Lodestone context, "PICC data" refers to the tag's encrypted metadata (UID and counter) within a SUN message.

RFC 4226	The IETF specification defining the HOTP algorithm. Lodestone implements RFC 4226 with HMAC-SHA256 rather than the original HMAC-SHA1.
RFID	Radio-Frequency Identification—a technology using electromagnetic fields to identify and track tags attached to objects. Legacy RFID systems (125kHz proximity cards, MIFARE Classic) lack the cryptographic protections of modern secure NFC.
SDM	Secure Dynamic Messaging—an NTAG 424 DNA feature that generates encrypted, authenticated messages containing tag metadata and file contents. Each SDM read produces a unique, verifiable response.
Secure Enclave	Apple's hardware-isolated security coprocessor present in iOS devices. Cryptographic keys generated in the Secure Enclave cannot be extracted, even by the operating system.
StrongBox	Android's designation for a dedicated hardware security module (separate from the main processor's TEE). Provides stronger isolation guarantees than TEE-only Keystore implementations.
SUN	Secure Unique NFC—NXP's marketing term for the dynamic, cryptographically authenticated messages generated by SDM-enabled tags. Each tap produces a unique SUN message.
TEE	Trusted Execution Environment—an isolated processing environment within a device's main processor, providing security guarantees stronger than the main operating system but weaker than dedicated hardware security modules.
UID	Unique Identifier—a factory-programmed identifier assigned to each NFC tag. The NTAG 424 DNA's 7-byte UID is used for key diversification and tag identification.

Appendix B: References

Standards and Specifications

1. M'Raihi, D., Bellare, M., Hoornaert, F., Naccache, D., and O. Ranen. "HOTP: An HMAC-Based One-Time Password Algorithm." RFC 4226, Internet Engineering Task Force, December 2005. https://datatracker.ietf.org/doc/html/rfc4226

2. International Organization for Standardization. "ISO/IEC 14443: Identification cards — Contactless integrated circuit cards — Proximity cards." ISO/IEC, 2018.

3. National Institute of Standards and Technology. "Advanced Encryption Standard (AES)." FIPS PUB 197, November 2001. https://csrc.nist.gov/publications/detail/fips/197/final

4. National Institute of Standards and Technology. "Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication." NIST Special Publication 800-38B, May 2005. https://csrc.nist.gov/publications/detail/sp/800-38b/final

NXP Documentation

5. NXP Semiconductors. "NTAG 424 DNA and NTAG 424 DNA TagTamper: Product Data Sheet." Rev. 3.3, 2023. https://www.nxp.com/docs/en/data-sheet/NT4H2421Gx.pdf

6. NXP Semiconductors. "AN12196: NTAG 424 DNA and NTAG 424 DNA TagTamper Features and Hints." Application Note, Rev. 1.5, 2023. https://www.nxp.com/docs/en/application-note/AN12196.pdf

7. NXP Semiconductors. "AN12321: Symmetric Originality Signature with NTAG 424 DNA." Application Note, 2020.

Platform Security Documentation

8. Google. "Android Keystore System." Android Developers Documentation. https://developer.android.com/training/articles/keystore

9. Apple. "Secure Enclave." Apple Platform Security Guide. https://support.apple.com/guide/security/secure-enclave-sec59b0b31ff/web

Open Source Libraries

10. johnnyb/ntag424-java: Java library for NTAG 424 DNA operations. MIT License. (https://github.com/johnnyb/ntag424-java)

Document Version 1.0 — DRAFT — February 2026

Groundspeak, Inc. — Confidential and Proprietary