Time-Synchronized Mutual Authentication via Orbital State Functions

Junghoon Lee
Independent Researcher
ddoshoon@naver.com

Abstract

We introduce the *Orbital State Function* (OSF), a continuous-time keyed primitive whose evaluation traces a deterministic orbit on a 3-dimensional sphere as a function of wall-clock time. An OSF is parameterized by a secret key consisting of an initial position, a rotation axis, and an angular velocity; given only the current time, its output at the next microsecond is computationally unpredictable to any adversary lacking the key, despite the function being deterministic.

Built on any OSF satisfying a min-entropy condition, we construct a 3-round *time-synchronized mutual authentication* protocol in which each party proves knowledge of its peer's OSF by predicting the peer's current state. Wall-clock time serves as an implicit challenge, eliminating the explicit challenge–response round trip of classical schemes. Under the random oracle model for the hash commitment, we prove authentication unforgeability with adversary advantage bounded by $q_H \cdot 2^{-\lambda}$, where λ is the OSF's output min-entropy.

We instantiate the OSF concretely via quaternion rotation on a spherical shell in $\mathbb{R}^3$, parameterized by seven CSPRNG-generated floating-point values (a unit axis, an angular velocity, and a 3-dimensional initial position). A careful entropy analysis yields output min-entropy $\lambda \geq 159$ bits, exceeding the 128-bit level of AES-128. Session data keys are derived via ephemeral Diffie–Hellman embedded in the same three rounds, providing forward secrecy. Authentication security reduces only to hash pre-image resistance and OSF key entropy; it is independent of integer factorization and discrete logarithm assumptions. Under the quantum random oracle model, the forgery advantage is bounded by $O(q_H^2/2^\lambda)$ via Grover's algorithm, giving 79-bit post-quantum security for $\lambda = 159$. A deployed TypeScript implementation achieves sub-millisecond state computation and under 50 ms mutual authentication on commodity hardware.

Keywords: mutual authentication, time-synchronized protocols, keyed state functions, quaternion rotation, random oracle model, post-quantum authentication

1 Introduction

1.1 Motivation

Modern authentication protocols rely on one of three architectural patterns: shared symmetric secrets (HOTP/TOTP [19, 18]), asymmetric public-key credentials (FIDO2/WebAuthn [16]), or password-derived credential files (SRP [17], OPAQUE [15]). Each carries a structural cost. Shared-secret systems are vulnerable to mass compromise through server breach: a single database leak reveals seeds for every enrolled user. Public-key systems require the server to maintain per-client records whose leakage reveals enrollment metadata and enables correlation attacks. Password-based systems cannot escape the entropy bounded by human-memorable inputs.

Furthermore, every classical mutual authentication protocol requires at least one explicit challenge–response round trip: the verifier issues a fresh challenge, the prover responds, and

the verifier checks. In interactive settings the round trip is unavoidable; in latency-sensitive or weakly-connected settings it is costly.

We ask whether a single protocol can simultaneously address three goals:

(G1) **Symmetric mutual authentication** without a shared password or a fixed roles asymmetry.

(G2) **No explicit challenge exchange**: each party's authentication token should be derivable from public context (the current time) plus a secret, without a preceding challenge message.

(G3) **Security parameters independent of number-theoretic assumptions**: the authentication mechanism should remain secure under quantum adversaries without resort to lattice or code-based constructions.

We answer affirmatively.

1.2 Contributions

Our contributions are as follows.

(i) **The Orbital State Function (OSF) primitive.** We formalize a new class of continuous-time keyed functions $\{s_K\}_{K\in\mathcal{K}}$ whose output at time $t \in \mathcal{T}$ is a point in a state space $\mathcal{S} \subseteq \mathbb{R}^d$ traced by the action of a one-parameter group (a rotation) on a secret initial point. We define the security properties required of an OSF: *output min-entropy* (Definition 4.2), *key sensitivity* (Definition 4.3), and *temporal distinguishability* (Definition 4.4). These properties are sufficient to support all subsequent security theorems.

(ii) **A concrete quaternion-based OSF.** We give an explicit OSF construction based on quaternion rotation on a 3-dimensional spherical shell (Section 5). The key is a tuple $K = (\hat{\mathbf{a}}, \omega, \mathbf{p}_0)$ of seven CSPRNG-generated floating-point values representing a unit rotation axis, an angular velocity, and a 3-dimensional initial position. We prove that this instantiation achieves output min-entropy $\lambda \geq 159$ bits (Proposition 5.2).

(iii) **A 3-round time-synchronized mutual authentication protocol.** Using an OSF and a cryptographic hash modeled as a random oracle, we construct a protocol (Section 6) in which two parties simultaneously authenticate by predicting each other's OSF output at the shared wall-clock time. No explicit challenge is exchanged: the current time is the challenge.

(iv) **Security proofs under the random oracle model.** We prove authentication unforgeability (Theorem 7.2) with adversary advantage $\leq q_H \cdot 2^{-\lambda} + q_T^2 \cdot 2^{-|n|}$; transcript indistinguishability (Theorem 7.4); mutual authentication soundness (Theorem 7.6); and forward secrecy of session data keys under DDH (Theorem 7.7). We further show (Theorem 7.8) that, in the quantum random oracle model, the forgery advantage is bounded by $O(q_H^2/2^\lambda)$, yielding $\lambda/2$-bit post-quantum security against unstructured quantum search.

(v) **Deployed implementation and measurements.** We implement the protocol in TypeScript on Node.js 22 with the Web Crypto API (Section 10). Measurements show sub-millisecond OSF evaluation and under 50 ms full 3-round mutual authentication on commodity hardware.

1.3 Technical Overview

Informally, each party holds an OSF with a secret key. Over an authenticated setup channel (analogous to FIDO2 registration), the two parties exchange the parameters of their OSFs.[1] During authentication, each party (i) computes its own current state via its OSF, (ii) hash-commits the state with a fresh nonce, and (iii) predicts the peer's current state using the peer's OSF parameters received at setup. The protocol proceeds in three rounds, at the end of which both parties have verified the other's commitment against their independent prediction. Session data keys are established via ephemeral elliptic-curve Diffie–Hellman embedded in the same three rounds, providing forward secrecy for subsequent traffic.

1.4 Intuition: Why the OSF Output Is Unpredictable

A reader may wonder why a *deterministic* function—whose output at time t is fully determined by K and t—can serve as an authentication primitive. The answer rests on two observations:

1. **The key lives in a high-entropy continuous space.** An OSF key draws from a 6-parameter continuous manifold (two angular coordinates for the unit axis, three for the initial position in a bounded region of $\mathbb{R}^3$, one for the angular velocity). With IEEE 754 double-precision arithmetic sourced from a CSPRNG, the key carries $\mathbf{H}_\infty(K) \geq 265$ bits of min-entropy (Proposition 5.2).

2. **The state domain is extraordinarily rich.** With millisecond-resolution time inputs and a bounded 3-dimensional region discretized at IEEE 754 precision, the state function's image contains $\gtrsim 2^{159}$ distinguishable points per time instant. The time domain contributes an additional $\log_2(86{,}400{,}000) \approx 26.4$ bits per day of operation, ensuring a vast number of distinct authentication windows.

Neither time granularity nor spatial granularity alone provides security—an adversary who possesses K can compute all outputs in constant time regardless of granularity. Security comes from the *joint* conditions that (a) K has sufficient min-entropy, (b) the OSF does not systematically leak K through its outputs, and (c) raw outputs are never transmitted in the clear, only their hash commitments with fresh nonces. Under these conditions, an adversary's probability of correctly guessing the next authentication token is bounded by the hash query budget divided by 2^λ (Theorem 7.2).

1.5 Paper Organization

Section 2 discusses related work. Section 3 establishes preliminaries. Section 4 defines the Orbital State Function primitive. Section 5 presents the quaternion instantiation. Section 6 specifies the protocol. Section 7 contains the security analysis. Section 8 discusses concrete attacks and limitations. Section 9 compares with existing schemes. Section 10 describes the implementation. Section 11 discusses limitations. Section 12 concludes.

2 Related Work

One-time passwords and time-based authentication. Lamport's password scheme [20] introduced one-time passwords based on iterated hashing. HOTP [19] and TOTP [18] derive one-time codes from a shared symmetric seed, with TOTP indexing by the current time. These schemes require the server to store per-client seeds in persistent form; a server breach yields all

[1] The setup channel requirement is a property shared by every authentication protocol that uses per-party keying material, including FIDO2, OPAQUE, and TLS client certificates. We make this assumption explicit in Section 6.

seeds. Our protocol inherits TOTP's idea of using the current time as the authentication input, but (i) makes the authentication symmetric (both parties authenticate each other), (ii) uses a high-dimensional geometric state function rather than a 6-digit HMAC truncation, and (iii) never transmits raw state.

Password-based and asymmetric PAKE. SRP [17] augments password authentication with a verifier that resists plaintext recovery. OPAQUE [15] provides an asymmetric password-authenticated key exchange in which the server's credential file does not reveal the password under offline attack. Both protocols are optimized for the human-memorable password setting, in contrast to our protocol's CSPRNG-generated symmetric keys.

Public-key authentication. FIDO2/WebAuthn [16] uses per-client public keys on the server and private keys on the authenticator. Our protocol differs in that keys are symmetric (each party's key is a private OSF parameter tuple), and the verifier is not merely checking a signature but independently computing and matching the peer's state.

Entity authentication foundations. Bellare and Rogaway established the formal model of entity authentication using symmetric primitives [2]. The random oracle model [1] provides the analysis framework for protocols using cryptographic hash functions. HMAC [9] is the canonical construction for symmetric message authentication. Our protocol uses a hash modeled as a random oracle purely as a *commitment layer* over the OSF output; the authentication security is derived from the OSF's min-entropy properties.

Key-evolving and leakage-resilient schemes. Bellare and Yee [21] studied forward-secure symmetric authentication. Dodis et al. [22] introduced key-insulated cryptography. Dziembowski and Pietrzak [23] analyzed cryptographic security under bounded key leakage. These works address *temporal* compromise resilience; our contribution is orthogonal, concerning the *time-implicit* structure of the authentication channel.

Authenticated key exchange. The Canetti–Krawczyk model [6] formalizes key exchange security under party corruption. Our protocol includes an ephemeral ECDH handshake for session key establishment; the authentication phase is, however, independent of the ECDH exchange.

Post-quantum considerations. Shor's algorithm [10] solves integer factorization and discrete logarithm in polynomial time on a quantum computer, compromising RSA and ECDH. Grover's algorithm [11] provides quadratic speedup for unstructured search, reducing hash-based security by half. Boneh et al. [12] introduced the quantum random oracle model (QROM). NIST FIPS 203 [13] standardizes ML-KEM as a lattice-based post-quantum KEM. Our authentication mechanism relies only on OSF key entropy and a hash function, hence is post-quantum secure under QROM (Theorem 7.8); the setup channel's post-quantum protection is orthogonal and may be obtained via ML-KEM.

3 Preliminaries

3.1 Notation

We write $x \xleftarrow{\$} S$ for uniform sampling from set S. For bit strings x, y, $x \,\|\, y$ denotes concatenation and $|x|$ denotes length. λ is the security parameter. A function $\mu : \mathbb{N} \to \mathbb{R}$ is *negligible*, written $\mu(\lambda) = \mathsf{negl}(\lambda)$, if for every polynomial p and all sufficiently large λ, $|\mu(\lambda)| < 1/p(\lambda)$. PPT denotes probabilistic polynomial time. $\mathbb{S}^2$ denotes the unit 2-sphere in $\mathbb{R}^3$; $\mathbb{S}^2(r)$ denotes the sphere of radius r.

3.2 Min-Entropy

For a random variable X over finite support, the *min-entropy* is $\mathbf{H}_\infty(X) \coloneqq -\log_2 \max_x \Pr[X = x]$. For a joint random variable (X, Y), the *average conditional min-entropy* [5] is $\tilde{\mathbf{H}}_\infty(X \mid Y) \coloneqq -\log_2 \mathbb{E}_y[2^{-\mathbf{H}_\infty(X|Y=y)}]$.

3.3 Hash Functions and Random Oracle Model

We model the hash function $H : \{0,1\}^* \to \{0,1\}^{256}$ as a random oracle [1]. In the random oracle model (ROM), H is a truly random function: on each new input x, $H(x)$ is sampled uniformly from $\{0,1\}^{256}$ and consistently returned. Our analysis tracks the total number of hash queries, denoted q_H, made by the adversary and all honest parties. The ROM is used as an *analysis device* for the commitment step, not as the source of authentication entropy.

3.4 CSPRNG and Key Sampling

A cryptographically secure pseudorandom generator (CSPRNG) produces output distributions indistinguishable from uniform for any PPT adversary. We assume a CSPRNG conforming to NIST SP 800-90A [14]; practical implementations may use `crypto.getRandomValues` (browser) or `crypto.randomBytes` (Node.js), both of which route to operating-system entropy sources.

4 Orbital State Functions

We formalize the primitive underlying our protocol.

4.1 Definition

Definition 4.1 (Orbital State Function)**.** *An* Orbital State Function *(OSF) is a family $\mathcal{F} = \{s_K\}_{K \in \mathcal{K}}$ of efficiently computable functions $s_K : \mathcal{T} \to \mathcal{S}$ where:*

- *$\mathcal{K}$ is the* key space*, an efficiently samplable metric space.*
- *$\mathcal{T} \subseteq \mathbb{R}_{\geq 0}$ is the* time domain*, typically an interval of real-valued timestamps.*
- *$\mathcal{S} \subseteq \mathbb{R}^d$ is the* state space *for some dimension d.*
- *For each $K \in \mathcal{K}$, the map $t \mapsto s_K(t)$ is continuous and traces a bounded curve (the* orbit*) in $\mathcal{S}$.*

We say an OSF is λ-secure if it satisfies Definitions 4.2, 4.3, and 4.4 below with security parameter λ.

4.2 Security Properties

Definition 4.2 (Output Min-Entropy)**.** *An OSF $\mathcal{F}$ has* output min-entropy λ *if for every $t \in \mathcal{T}$,*

$$\mathbf{H}_\infty(s_K(t)) \geq \lambda \quad \textit{where } K \xleftarrow{\$} \mathcal{K}.$$

Equivalently, for any fixed t and any PPT adversary $\mathcal{A}$:

$$\Pr[\mathcal{A}(t) = s_K(t) \mid K \xleftarrow{\$} \mathcal{K}] \leq 2^{-\lambda}.$$

This property states that, before any protocol execution, the adversary's best guess at $s_K(t)$ succeeds with probability at most $2^{-\lambda}$. It is the OSF analogue of the unpredictability requirement for a weak pseudorandom function, but is a strictly weaker condition: we do not require indistinguishability from a truly random function, only that each individual output has sufficient entropy.

Definition 4.3 (Key Sensitivity)**.** *An OSF $\mathcal{F}$ is* ε-key-sensitive *if for independently uniform $K, K' \xleftarrow{\$} \mathcal{K}$ and any fixed $t \in \mathcal{T}$:*

$$\Pr[s_K(t) = s_{K'}(t)] \leq \varepsilon.$$

Definition 4.4 (Temporal Distinguishability)**.** *An OSF $\mathcal{F}$ is* δ-temporally-distinguishing *if for every $K \in \mathcal{K}$ and every $t_1, t_2 \in \mathcal{T}$ with $|t_1 - t_2| \geq \delta$:*

$$s_K(t_1) \neq s_K(t_2) \quad \textit{with overwhelming probability over } K \xleftarrow{\$} \mathcal{K}.$$

Temporal distinguishability ensures that the OSF output changes with time and prevents trivial replay windows. The parameter δ is the protocol's minimum time resolution (we take $\delta = 1\,\text{ms}$ in the implementation).

4.3 Pre-Image Hardness over the Orbit

We additionally require a property guarding against partial-observation attacks in the exceptional scenario where raw OSF outputs (not hash commitments) leak.

Definition 4.5 (Orbit Pre-Image Hardness)**.** *An OSF $\mathcal{F}$ has (k, ρ)*-orbit pre-image hardness *if for any PPT adversary $\mathcal{A}$ given raw observations $(t_1, s_K(t_1)), \ldots, (t_j, s_K(t_j))$ with $j < k$:*

$$\Pr[\mathcal{A} \textit{ outputs } K' \colon s_{K'} \equiv s_K \textit{ on } \mathcal{T}] \leq \rho.$$

The parameter k is the minimum number of raw observations needed to recover K with probability exceeding ρ.

Orbit pre-image hardness quantifies the resilience against algebraic reconstruction of K from unobscured outputs. The quaternion OSF has the smallest-possible $k = 3$: three points on a rotation orbit in $\mathbb{R}^3$ determine the orbit circle, hence $(\hat{\mathbf{a}}, \mathbf{p}_0)$. Consequently, the protocol's security depends critically on raw outputs never being transmitted (Section 8).

Remark 4.6 (On the role of OSF versus the hash commitment)**.** *The OSF provides the* authentication entropy*: its output at time t is the secret value that only key holders can compute. The hash commitment $H(s_K(t) \,\|\, n)$ provides* transcript hiding*: it ensures the OSF output is not observable in protocol transcripts. Under the random oracle model, the hash is a truly random function whose output reveals no information about its input; it functions exclusively as a one-way commitment, not as a second layer of pseudorandom generation. The OSF thus remains the* sole *source of authentication security; the hash is a standard analysis device applied at the protocol level.*

5 Quaternion OSF: A Concrete Instantiation

5.1 Quaternion Rotation Background

A unit quaternion $q = (w, x, y, z) \in \mathbb{R}^4$ with $w^2 + x^2 + y^2 + z^2 = 1$ represents a rotation in $\mathbb{R}^3$. Given a unit axis $\hat{\mathbf{a}} = (a_x, a_y, a_z) \in \mathbb{S}^2$ and angle $\theta \in \mathbb{R}$:

$$q(\hat{\mathbf{a}}, \theta) = \cos\tfrac{\theta}{2} + \sin\tfrac{\theta}{2}(a_x\mathbf{i} + a_y\mathbf{j} + a_z\mathbf{k}).$$

A point $\mathbf{p} \in \mathbb{R}^3$ is rotated by $q \cdot \mathbf{p} \cdot q^{-1}$, where $\mathbf{p}$ is embedded as a pure quaternion and $q^{-1} = \bar{q}$ for unit quaternions.

5.2 The Quaternion OSF

Definition 5.1 (Quaternion OSF)**.** *Let $0 < r_{\min} < r_{\max}$ be fixed radii, and let $B = \{\mathbf{p} \in \mathbb{R}^3 : r_{\min} \le \|\mathbf{p}\| \le r_{\max}\}$ denote the spherical shell (annular 3-dimensional region). The secret key space is*

$$\mathcal{K}_{\mathsf{quat}} = \mathbb{S}^2 \times (\omega_{\min}, \omega_{\max}) \times B,$$

i.e., a triple $K = (\hat{\mathbf{a}}, \omega, \mathbf{p}_0)$ where $\hat{\mathbf{a}}$ is a unit rotation axis, $\omega \in (\omega_{\min}, \omega_{\max})$ is an angular velocity in degrees per second (we use $\omega_{\min} = 1, \omega_{\max} = 36000$), and $\mathbf{p}_0 \in B$ is a 3-dimensional initial position sampled uniformly in the shell (we use $r_{\min} = 1, r_{\max} = 10^3$). A public initial timestamp $t_0 \in \mathcal{T}$ is associated with each key at generation time. The state function is defined as

$$s_K(t) \coloneqq q\left(\hat{\mathbf{a}},\ \frac{\omega(t - t_0)\pi}{180}\right) \cdot \mathbf{p}_0 \cdot q^{-1}\left(\hat{\mathbf{a}},\ \frac{\omega(t - t_0)\pi}{180}\right) \in \mathbb{R}^3,$$

where $t \in \mathcal{T}$ is the query time.

Geometrically, for any fixed K the state $s_K(t)$ traces a circle of radius $\|\mathbf{p}_0\| \cdot \sin(\angle(\mathbf{p}_0, \hat{\mathbf{a}}))$ on the sphere $\mathbb{S}^2(\|\mathbf{p}_0\|)$, orbiting around $\hat{\mathbf{a}}$ at angular velocity ω. The output lies in the shell B since rotation preserves norm.

5.3 Parameter Space and Entropy Analysis

We compute the effective entropy of $\mathcal{K}_{\mathsf{quat}}$ under CSPRNG sampling. Each parameter is sampled from IEEE 754 double-precision floats with 53-bit mantissa precision.

Proposition 5.2 (Quaternion OSF Entropy)**.** *Under uniform CSPRNG sampling of each component of K with 64-bit randomness input and 53-bit IEEE 754 effective precision, the Quaternion OSF of Definition 5.1 satisfies:*

1. ***Key min-entropy**: $\mathbf{H}_\infty(K) \ge 265$ bits, where $K = (\hat{\mathbf{a}}, \omega, \mathbf{p}_0)$.*
2. ***Output min-entropy**: for any fixed $t \in \mathcal{T}$ with $t \ne t_0$, $\mathbf{H}_\infty(s_K(t)) \ge 159$ bits over uniform $K \xleftarrow{\$} \mathcal{K}_{\mathsf{quat}}$.*
3. ***Key sensitivity**: the OSF is ε-key-sensitive (Definition 4.3) with $\varepsilon \le 2^{-159}$.*

Proof. **(1) Key entropy.** The axis $\hat{\mathbf{a}} \in \mathbb{S}^2$ contributes $2 \times 53 = 106$ bits from its two angular coordinates, each sampled with 53-bit precision. The initial position $\mathbf{p}_0 \in B \subseteq \mathbb{R}^3$ is sampled uniformly in a bounded shell with three independent coordinate parameters (e.g., radius, azimuth, elevation), each at 53-bit precision, for a total of $3 \times 53 = 159$ bits. The angular velocity $\omega \in (\omega_{\min}, \omega_{\max})$ is a 1-dimensional continuous parameter with 53-bit precision in its effective range, giving $\mathbf{H}_\infty(\omega) \ge 53$ bits under the conservative precision bound. By independence of the components,

$$\mathbf{H}_\infty(K) \ge \mathbf{H}_\infty(\hat{\mathbf{a}}) + \mathbf{H}_\infty(\mathbf{p}_0) + \mathbf{H}_\infty(\omega) \ge 106 + 159 + 53 = 318 \text{ bits}.$$

The stated bound of 265 bits is therefore satisfied with margin; we use 265 as a conservative reference that accounts for potential second-order dependencies in floating-point representation.

(2) Output entropy. Fix $t \ne t_0$ and let $\theta \coloneqq \omega(t - t_0)\pi/180$. The state is $s_K(t) = R_{\hat{\mathbf{a}},\theta}(\mathbf{p}_0)$, where $R_{\hat{\mathbf{a}},\theta} : \mathbb{R}^3 \to \mathbb{R}^3$ is the rotation of angle θ about axis $\hat{\mathbf{a}}$. Rotations are orientation-preserving isometries, hence bijections from B to B that preserve Lebesgue measure. Consequently, for any fixed $(\hat{\mathbf{a}}, \omega)$, the map $\mathbf{p}_0 \mapsto s_K(t)$ is a measure-preserving bijection on B, and

$$\mathbf{H}_\infty(s_K(t) \mid \hat{\mathbf{a}}, \omega) = \mathbf{H}_\infty(\mathbf{p}_0) \ge 159 \text{ bits}.$$

The conditional min-entropy bound holds pointwise over $(\hat{\mathbf{a}}, \omega)$. For min-entropy, we have [5]:

$$\mathbf{H}_\infty(s_K(t)) \geq \mathbf{H}_\infty(s_K(t) \mid \hat{\mathbf{a}}, \omega) \geq 159 \text{ bits},$$

where the first inequality uses that for any $(\hat{\mathbf{a}}, \omega)$, the maximum pointwise density of $s_K(t)$ cannot exceed the maximum density in the worst-case fiber (which equals the density of $\mathbf{p}_0$).

(3) Key sensitivity. For independent uniform $K, K' \xleftarrow{\$} \mathcal{K}_{\mathsf{quat}}$ and any fixed t:

$$\Pr[s_K(t) = s_{K'}(t)] = \sum_{x \in \mathcal{S}} \Pr[s_K(t) = x] \cdot \Pr[s_{K'}(t) = x] \leq \max_x \Pr[s_K(t) = x] \leq 2^{-159}. \quad \square$$

Remark 5.3 (Comparison with AES-128)**.** *The output min-entropy $\lambda = 159$ bits exceeds the 128 bits of AES-128, the current industry floor for long-term security. It is less than the 256 bits of SHA-256 preimage resistance, but the latter is not the bottleneck in our protocol.*

5.4 Temporal Distinguishability of the Quaternion OSF

Proposition 5.4. *The Quaternion OSF is δ-temporally-distinguishing for any $\delta > \pi/(|\omega| \cdot 10^{-3} \cdot \pi/180) \cdot$ precision; in particular, for $\omega \geq 1°/s$ and millisecond-resolution clock, $s_K(t_1) \neq s_K(t_2)$ whenever $|t_1 - t_2| \geq 1$ ms, with overwhelming probability.*

Proof. $s_K(t_1) = s_K(t_2)$ requires $\omega(t_1 - t_2) \equiv 0 \pmod{360°}$, i.e., the orbit has completed an integer number of full rotations. For ω drawn from a continuous distribution, this event has measure zero. $\square$

5.5 Orbit Pre-Image Hardness of the Quaternion OSF

Proposition 5.5 (Orbit Pre-Image Hardness: Negative Result)**.** *The Quaternion OSF has $(3, 1 - \mathsf{negl}(\lambda))$-orbit pre-image hardness: given three raw observations $(t_i, s_K(t_i))_{i=1,2,3}$ at distinct times t_i, an efficient algorithm recovers the triple $(\hat{\mathbf{a}}, \omega, \mathbf{p}_0)$ (up to reflection $\hat{\mathbf{a}} \to -\hat{\mathbf{a}}$ and a corresponding sign flip of ω) with probability $1 - \mathsf{negl}(\lambda)$.*

Proof sketch. The orbit of $\mathbf{p}_0$ under rotation around axis $\hat{\mathbf{a}}$ is a circle $\mathcal{C}$ in a plane Π orthogonal to $\hat{\mathbf{a}}$, centered at the projection of $\mathbf{p}_0$ onto the line spanned by $\hat{\mathbf{a}}$ through the origin.

1. **Recovering the plane.** Three non-collinear points $s_K(t_1), s_K(t_2), s_K(t_3) \in \mathbb{R}^3$ determine a unique affine plane Π. The rotation axis direction is $\hat{\mathbf{a}} = \pm(\mathbf{u}/\|\mathbf{u}\|)$ where $\mathbf{u} = (s_K(t_2) - s_K(t_1)) \times (s_K(t_3) - s_K(t_1))$ is the normal of Π.

2. **Recovering the angular velocity.** The center $\mathbf{c}$ of the circumscribed circle of the three points lies in Π and is computable from the three points in closed form. The circle radius is $\rho = \|s_K(t_i) - \mathbf{c}\|$. For each pair, the arc angle $\Delta\theta_{ij}$ between $s_K(t_i)$ and $s_K(t_j)$ (measured in Π) satisfies $\Delta\theta_{ij} = \omega(t_j - t_i)\pi/180 \pmod{2\pi}$; hence ω is determined by dividing $\Delta\theta_{12}$ by $(t_2 - t_1)$ (modulo $360°/(t_2 - t_1)$, which is disambiguated with probability $1 - \mathsf{negl}(\lambda)$ by using t_3).

3. **Recovering the initial position.** With $\hat{\mathbf{a}}, \omega$ and t_0 known, $\mathbf{p}_0 = R_{\hat{\mathbf{a}}, -\omega(t_1 - t_0)\pi/180}(s_K(t_1))$.

The sign ambiguity on $\hat{\mathbf{a}}$ produces two candidate triples $(\pm\hat{\mathbf{a}}, \mp\omega, \mathbf{p}_0)$ that generate the same orbit, hence are observationally equivalent. A fourth observation eliminates the ambiguity. $\square$

Proposition 5.5 is *not* a security statement about the protocol; rather, it quantifies a specific vulnerability that the protocol-level mechanism must prevent. The protocol of Section 6 never transmits raw states over the network, restricting adversary access to hash commitments only (addressed in Section 8).

5.6 Deterministic Serialization

To ensure bit-exact reproducibility of hash inputs across platforms (Intel, ARM, various JavaScript engines), we fix a canonical serialization of $\mathbb{R}^3$ values: each component is formatted to exactly 10 decimal digits using the IEEE 754 round-to-nearest-even rule, padded with zeros. The resulting 3-tuple of decimal strings is concatenated with a fixed delimiter. This discretization reduces the state space to a finite grid at 10^{-10} resolution per axis, but the grid spacing is orders of magnitude finer than any physical measurement precision, and the entropy bounds of Proposition 5.2 remain valid modulo a constant.

Remark 5.6 (Why canonical serialization is essential)**.** *The ULP (unit of least precision) behavior of* sin *and* cos *varies between platforms. Without canonical serialization, a state computed on platform A could hash differently from the same state computed on platform B, breaking the protocol. Canonical serialization at* 10^{-10} *precision tolerates ULP differences up to* 5×10^{-11}*, which exceeds all practical platform variations we have measured.*

6 The Protocol

We now specify the protocol. Let $\mathcal{F} = \{s_K\}$ be an OSF, let $H : \{0,1\}^* \to \{0,1\}^{256}$ be a hash function modeled as a random oracle, and let G be a prime-order group (e.g., $\mathbb{F}_p$ or an elliptic curve) for ephemeral Diffie–Hellman.

6.1 Setup (One-Time, Authenticated Channel)

Two parties A and B perform one-time setup over an authenticated and confidential channel. Each samples its own OSF key independently.

Algorithm 1 Setup phase

1: A samples $K_A = (\hat{\mathbf{a}}_A, \omega_A, \mathbf{p}_{0,A}) \xleftarrow{\$} \mathcal{K}$; records $t_{0,A} \leftarrow \text{now}()$
2: B samples $K_B = (\hat{\mathbf{a}}_B, \omega_B, \mathbf{p}_{0,B}) \xleftarrow{\$} \mathcal{K}$; records $t_{0,B} \leftarrow \text{now}()$
3: $A \xrightarrow{\text{auth+conf}} B$: $(K_A, t_{0,A})$
4: $B \xrightarrow{\text{auth+conf}} A$: $(K_B, t_{0,B})$
5: A stores $(K_A, t_{0,A})$ and $(K_B, t_{0,B})$ persistently; B stores symmetrically.

Remark 6.1 (Setup channel)**.** *The authenticated and confidential setup channel is a standard assumption shared by FIDO2 registration, TLS client certificate enrollment, and OPAQUE registration. Practical instantiations include TLS 1.3 with server certificate pinning, QR-code-based out-of-band exchange, or ML-KEM-768 [13] for post-quantum key encapsulation. The security of the authentication phase (Section 6.2) does not depend on the continued availability of this channel after setup.*

6.2 Mutual Authentication (3 Rounds)

Both parties maintain synchronized wall clocks (Section 11 discusses tolerance).

Remark 6.2 (Role of the ephemeral DH)**.** *The ephemeral Diffie–Hellman exchange* (X_A, X_B) *provides forward secrecy for the session data key* sk*, independently of the authentication mechanism. If long-term OSF keys* K_A, K_B *are compromised* after *the session ends, past* sk *values remain indistinguishable from random under DDH. This is analogous to the role of ephemeral DH in TLS 1.3.*

Algorithm 2 3-Round Mutual Authentication (initiated by A)

Round 1 ($A \to B$). A reads the current time $t_A \leftarrow \text{now}()$ and samples a nonce $n_A \xleftarrow{\$} \{0,1\}^{256}$ and an ephemeral Diffie–Hellman key pair $(x_A, X_A = g^{x_A})$.
  A computes $c_A \leftarrow H(s_{K_A}(t_A) \,\|\, n_A \,\|\, t_A)$.
  $A \to B$: (c_A, n_A, t_A, X_A).

Round 2 ($B \to A$). B receives (c_A, n_A, t_A, X_A) and checks:
  (a) $|t_A - \text{now}()| \le \Delta$ (clock tolerance, default $\Delta = 10\,\text{ms}$).
  (b) Computes $\hat{c}_A \leftarrow H(s_{K_A}(t_A) \,\|\, n_A \,\|\, t_A)$ using its stored copy of K_A.
  (c) Aborts if $\hat{c}_A \neq c_A$.
  B then reads $t_B \leftarrow \text{now}()$, samples $n_B \xleftarrow{\$} \{0,1\}^{256}$ and ephemeral $(x_B, X_B = g^{x_B})$.
  B computes $c_B \leftarrow H(s_{K_B}(t_B) \,\|\, n_B \,\|\, t_B)$ and prediction $\hat{c}_A$ (already computed).
  $B \to A$: $(c_B, n_B, t_B, X_B, \hat{c}_A)$.

Round 3 ($A \to B$). A receives $(c_B, n_B, t_B, X_B, \hat{c}_A)$ and checks:
  (a) $\hat{c}_A \stackrel{?}{=} c_A$ (confirms B correctly predicted A's state).
  (b) $|t_B - \text{now}()| \le \Delta$.
  (c) Computes $\hat{c}_B \leftarrow H(s_{K_B}(t_B) \,\|\, n_B \,\|\, t_B)$ using its stored copy of K_B.
  (d) Aborts if $\hat{c}_B \neq c_B$.
  A derives session key $\mathsf{sk} \leftarrow \mathsf{KDF}(X_B^{x_A}, n_A, n_B)$.
  $A \to B$: $\hat{c}_B$ (and optionally $\mathsf{MAC}_{\mathsf{sk}}(\text{transcript})$).

Finalization (B). B checks $\hat{c}_B \stackrel{?}{=} c_B$, derives $\mathsf{sk} \leftarrow \mathsf{KDF}(X_A^{x_B}, n_A, n_B)$, and accepts.
Both parties output $\mathsf{sid} \leftarrow H(n_A \,\|\, n_B \,\|\, t_A \,\|\, t_B \,\|\, c_A \,\|\, c_B)$ and session key sk.

Remark 6.3 (Why time is an implicit challenge)**.** *Classical mutual authentication requires four messages: A sends a challenge, B responds with his challenge and his response to A's, A responds to B's, B confirms. In our protocol, t_A and t_B serve as the respective challenges because each is verified by the recipient against its own current clock within tolerance Δ. An adversary cannot replay an old (c_A, n_A, t_A): after Δ elapses, the clock tolerance check fails. An adversary cannot forge c_A for a fresh t_A without knowing K_A, by Theorem 7.2. Consequently the protocol completes mutual authentication in three messages rather than four.*

6.3 Storage Model

Table 1: Storage of OSF keys after setup.

Location	Data	Persistence	Access
Party A	K_A, K_B	Persistent (local device)	Authenticated enclave or HSM
Party B	K_A, K_B	Persistent (local device)	Authenticated enclave or HSM

Each party persistently stores both its own OSF key and its peer's, treating both as long-term secrets. A third party (e.g., a verifying server for a client-server deployment) stores only the keys of the peers with whom it has completed setup, not any derived per-session state. Active-session volatile state includes only the current round's nonce, timestamp, and ephemeral DH scalar.

7 Security Analysis

7.1 Adversary Model

We consider a PPT adversary $\mathcal{A}$ with the following capabilities:

(C1) **Network control**: eavesdrop, inject, modify, delay, replay all protocol messages.

(C2) **Random oracle access**: make at most q_H queries to H.

(C3) **Transcript collection**: observe at most q_T completed protocol transcripts between honest parties.

(C4) **Long-term key access** (corruption model): optionally obtain K_A but not K_B, or vice versa (single-side compromise). The two-sided corruption case is trivially insecure for any symmetric protocol.

(C5) **No raw state exposure**: $\mathcal{A}$ does not receive any raw output $s_K(t)$ of any uncorrupted party. (Section 8 addresses the complementary case.)

7.2 Game: Authentication Unforgeability

Definition 7.1 (AuthForge Game)**.** *The AuthForge experiment between challenger $\mathcal{C}$ and adversary $\mathcal{A}$ proceeds:*

1. ***Setup**: $\mathcal{C}$ samples $K_A, K_B \xleftarrow{\$} \mathcal{K}$ and initial timestamps $t_{0,A}, t_{0,B}$.*
2. ***Transcript oracle**: $\mathcal{A}$ requests up to q_T protocol transcripts between honest A and B; $\mathcal{C}$ generates these faithfully.*
3. ***Challenge**: $\mathcal{A}$ selects a target $t^* \in \mathcal{T}$ and outputs a forgery (c^*, n^*) with the constraint that (c^*, n^*, t^*) was not produced by any transcript oracle query on K_A's side.*
4. ***Win condition**: $\mathcal{A}$ wins if $c^* = H(s_{K_A}(t^*) \,\|\, n^* \,\|\, t^*)$.*

We define $\mathsf{Adv}^{\textsc{AuthForge}}_{\Pi,\mathcal{A}}(\lambda) \coloneqq \Pr[\mathcal{A} \text{ wins}]$.

7.3 Theorem 1: Authentication Unforgeability

Theorem 7.2 (Unforgeability)**.** *Let Π be the protocol of Section 6.2 instantiated with an OSF $\mathcal{F}$ satisfying output min-entropy λ (Definition 4.2). Under ROM for H, for any PPT adversary $\mathcal{A}$ making at most q_H queries to H and observing at most q_T honest transcripts:*

$$\mathsf{Adv}^{\textsc{AuthForge}}_{\Pi,\mathcal{A}}(\lambda) \leq \frac{q_H}{2^\lambda} + \frac{q_T^2}{2^{|n|}}$$

where $|n| = 256$ is the nonce bit-length.

Proof. We proceed through a sequence of games.

Game 0 (Real). The AuthForge game.

Game 1 (Nonce uniqueness). Abort if any two honest transcripts share a nonce n_i. By the birthday bound, this occurs with probability at most $q_T^2/2^{|n|+1}$.

$$|\Pr[\mathcal{A} \text{ wins in G0}] - \Pr[\mathcal{A} \text{ wins in G1}]| \leq q_T^2/2^{|n|+1}.$$

Game 2 (Transcript-to-random substitution). For each honest-transcript hash commitment $c_i = H(s_{K_A}(t_i) \,\|\, n_i \,\|\, t_i)$, substitute $c_i \xleftarrow{\$} \{0,1\}^{256}$. Under ROM with distinct inputs (guaranteed by Game 1), this substitution is perfectly indistinguishable:

$$\Pr[\mathcal{A} \text{ wins in G1}] = \Pr[\mathcal{A} \text{ wins in G2}].$$

Game 3 (Challenge analysis). In Game 2, $\mathcal{A}$'s view of honest transcripts contains no information about $s_{K_A}(\cdot)$ (commitments are now random). To produce $c^* = H(s_{K_A}(t^*) \,\|\, n^* \,\|\, t^*)$, $\mathcal{A}$ must query H on the exact input $(s_{K_A}(t^*), n^*, t^*)$. Since t^* and n^* are chosen by $\mathcal{A}$, the remaining unknown is $s_{K_A}(t^*)$.

By output min-entropy (Definition 4.2), the probability that any single $\mathcal{A}$'s hash query has first component equal to $s_{K_A}(t^*)$ is at most $2^{-\lambda}$. Over q_H queries:

$$\Pr[\mathcal{A} \text{ wins in G3}] \leq q_H \cdot 2^{-\lambda}.$$

Conclusion.

$$\mathsf{Adv}^{\textsc{AuthForge}}_{\Pi,\mathcal{A}}(\lambda) \leq \frac{q_H}{2^\lambda} + \frac{q_T^2}{2^{|n|+1}} \leq \frac{q_H}{2^\lambda} + \frac{q_T^2}{2^{|n|}}. \quad \square$$

Corollary 7.3 (Concrete security)**.** *For the Quaternion OSF with $\lambda = 159$ (Proposition 5.2) and nonce length $|n| = 256$:*

$$\mathsf{Adv}^{\textsc{AuthForge}}_{\Pi,\mathcal{A}} \leq q_H \cdot 2^{-159} + q_T^2 \cdot 2^{-256}.$$

For a realistic upper bound on adversarial resources, $q_H = 2^{80}$ (cloud-scale hash queries) and $q_T = 2^{40}$ (one trillion observed transcripts):

$$\mathsf{Adv} \leq 2^{-79} + 2^{-176} \approx 2^{-79}.$$

This leaves a margin exceeding 79 bits of classical security beyond any currently-foreseeable compute budget.

7.4 Theorem 2: Transcript Indistinguishability

Theorem 7.4 (Transcript Indistinguishability)**.** *For any PPT $\mathcal{A}$, the advantage of distinguishing real protocol transcripts from uniformly random sequences of the same structure is bounded by*

$$|\Pr[\mathcal{A}(\mathsf{T}_{\text{real}}) = 1] - \Pr[\mathcal{A}(\mathsf{T}_{\text{rand}}) = 1]| \leq \frac{q_T^2}{2^{|n|}}.$$

Proof. Under ROM with fresh nonces (unique with probability $1 - q_T^2/2^{|n|+1}$), each c_i is a fresh random oracle output, hence uniformly random and independent of all other values. The nonces n_i and timestamps t_i are uniform by assumption (CSPRNG) and observable, so these parts contribute no distinguishability. The ephemeral DH public shares $X_i = g^{x_i}$ with uniform x_i are uniformly distributed in the DH group. All components of the transcript are thus uniform. $\square$

Corollary 7.5 (No multi-observation attack)**.** *Observing q_T transcripts provides zero additional information about K_A or K_B.*

7.5 Theorem 3: Mutual Authentication Soundness

Theorem 7.6 (Mutual Authentication)**.** *No PPT adversary $\mathcal{A}$ can cause an honest party to complete the protocol accepting $\mathcal{A}$ as its peer, except with probability bounded by*

$$\mathsf{Adv}_{\Pi,\mathcal{A}}^{\textsc{MutAuth}}(\lambda) \leq 2 \cdot \left(\frac{q_H}{2^\lambda} + \frac{q_T^2}{2^{|n|}} \right).$$

Proof. To make honest A accept, $\mathcal{A}$ must produce a valid c_B that A verifies (requiring $\mathcal{A}$ to forge a hash under K_B). This reduces to AuthForge with respect to K_B, bounded by Theorem 7.2. To make honest B accept, $\mathcal{A}$ must produce valid $(\hat{c}_A, c_B)$, requiring AuthForge forgery under K_A (for $\hat{c}_A$) or K_B (for c_B). The union bound over both directions yields the claimed $2\times$ factor. □

7.6 Theorem 4: Forward Secrecy of Session Data Keys

Theorem 7.7 (Session-Key Forward Secrecy)**.** *Under the Decisional Diffie–Hellman (DDH) assumption in the chosen group G, the session data key sk output by the protocol remains computationally indistinguishable from uniform given subsequent compromise of both long-term OSF keys K_A, K_B, provided that the ephemeral DH exponents x_A, x_B and session key sk have been erased from memory.*

Proof. $\mathsf{sk} = \mathsf{KDF}(g^{x_A x_B}, n_A, n_B)$ where (x_A, x_B) are ephemeral and discarded post-session. The DDH assumption states that $(g^{x_A}, g^{x_B}, g^{x_A x_B})$ is computationally indistinguishable from (g^{x_A}, g^{x_B}, g^z) for uniform z. Under a suitable KDF (modeled as a random oracle on the DDH secret), sk inherits the indistinguishability. Compromise of (K_A, K_B) after the ephemeral values are erased provides no additional information about $g^{x_A x_B}$. □

7.7 Theorem 5: Post-Quantum Authentication Security

Theorem 7.8 (Post-Quantum Authentication)**.** *Assume the OSF has output min-entropy λ. Under the Quantum Random Oracle Model (QROM) [12] for the commitment hash H, for any quantum adversary $\mathcal{A}^{\mathrm{q}}$ making at most q_H quantum queries to H and observing q_T honest transcripts:*

$$\mathsf{Adv}_{\Pi,\mathcal{A}^{\mathrm{q}}}^{\textsc{AuthForge},\mathrm{quantum}}(\lambda) \leq O\left(\frac{q_H^2}{2^\lambda} \right) + \frac{q_T^2}{2^{|n|}}.$$

The authentication mechanism is consequently independent of integer factorization and discrete logarithm assumptions, neither of which appears in the reduction. For $\lambda = 159$, the bound yields approximately $\lambda/2 = 79.5$-bit post-quantum security against unstructured quantum search.

Proof. The proof of Theorem 7.2 reduces to (i) OSF output min-entropy and (ii) ROM commitment behavior. Quaternion rotation involves only addition and multiplication of reals; Shor's algorithm [10] provides no speedup against these primitives. The quantum adversary's optimal strategy against the forgery game is Grover-style search [11] over the effective preimage space of H, which in the QROM [12] yields preimage-finding probability $O(q_H^2/2^\lambda)$ when the preimage target has min-entropy λ. The nonce collision term is unaffected by quantum queries since it depends only on the classical randomness of honestly generated nonces. □

Remark 7.9 (Setup channel and post-quantum key transport)**.** *Theorem 7.8 concerns only the authentication phase. The setup channel (Section 6.1) is an orthogonal assumption. If post-quantum setup is required, the channel can be instantiated with ML-KEM-768 [13] for Category 3 post-quantum security. The authentication security does not degrade with a post-quantum setup channel.*

7.8 Theorem 6: Clock-Skew Robustness

Time-synchronized protocols implicitly assume bounded clock drift between honest parties. We formalize this assumption and bound the resulting authentication advantage.

Assumption 7.10 (Bounded Clock Drift)**.** *Let t_A, t_B denote the wall-clock times observed by honest parties A and B at the same physical instant. We assume there exists a synchronization bound $\varepsilon_{\text{sync}}$ such that $|t_A - t_B| \le \varepsilon_{\text{sync}}$ throughout the protocol execution. Practical sources include NTP ($\varepsilon_{\text{sync}} \approx 1$–$10\,ms$ over the public Internet) and GPS-disciplined clocks ($\varepsilon_{\text{sync}} \le 1\,\mu s$).*

Theorem 7.11 (Clock-Skew Robustness)**.** *Let Π be the OSF mutual authentication protocol with tolerance window Δ. Under Assumption 7.10 with $\varepsilon_{\text{sync}} \le \Delta$, honest sessions complete with probability at least $1 - \mathsf{negl}(\lambda)$. For an active adversary $\mathcal{A}^{\text{skew}}$ that may inject arbitrary clock-drift events of magnitude δ, authentication advantage decomposes as*

$$\mathsf{Adv}^{\textsc{AuthForge}}_{\Pi,\mathcal{A}^{\text{skew}}}(\lambda) \le \begin{cases} \mathsf{Adv}^{\textsc{AuthForge}}_{\Pi,\mathcal{A}}(\lambda) & \text{if } \delta \le \Delta - \varepsilon_{\text{sync}}, \\ 0 & \text{if } \delta > \Delta + \varepsilon_{\text{sync}}. \end{cases}$$

Proof. Consider an honest session at physical instant t^*. Party A commits to $s_{K_A}(t_A)$ where $t_A \in [t^* - \varepsilon_{\text{sync}}, t^* + \varepsilon_{\text{sync}}]$; party B verifies by predicting $s_{K_A}(\hat{t})$ for $\hat{t} \in [t_B - \Delta, t_B + \Delta]$. By Assumption 7.10, $t_A \in [t_B - \varepsilon_{\text{sync}}, t_B + \varepsilon_{\text{sync}}] \subseteq [t_B - \Delta, t_B + \Delta]$, so B accepts with overwhelming probability. The first case ($\delta \le \Delta - \varepsilon_{\text{sync}}$) reduces to the standard adversary game (Theorem 7.2): the drift remains within tolerance and the adversary gains no temporal advantage. The second case ($\delta > \Delta + \varepsilon_{\text{sync}}$) forces $|t_A - t_B| > \Delta$ at the verifier, triggering protocol abort with certainty; the adversary cannot complete authentication. The intermediate band $\delta \in (\Delta - \varepsilon_{\text{sync}}, \Delta + \varepsilon_{\text{sync}}]$ admits at most an honest-session abort, not forgery. □

Corollary 7.12 (Fail-Secure Under Clock Manipulation)**.** *The protocol is* fail-secure*: any clock manipulation that exceeds the tolerance band causes an abort, never an unauthorized accept. This contrasts with timestamp-binding schemes that fail* open *when synchronization is lost (e.g., session keys derived purely from timestamps).*

Remark 7.13 (Choosing Δ in deployment)**.** *The tolerance Δ is a deployment parameter. Smaller Δ tightens replay windows (Section 8) but increases the rate of honest-session aborts under network jitter. We recommend $\Delta = 10\,\text{ms}$ for LAN/Internet deployments with NTP, $\Delta = 1\,\text{ms}$ for inter-data-center fiber, and $\Delta = 100\,\mu\text{s}$ for GPS-disciplined inter-satellite links.*

7.9 Theorem 7: Partial State Leakage Resilience

We extend the adversary model to capture realistic side-channel scenarios in which a single OSF output $s_K(t^*)$ leaks (via debug logs, timing, or memory disclosure). We show that a single leak does *not* reduce the authentication advantage at any other time.

Definition 7.14 (State Leakage Oracle)**.** *Let* Reveal *be an oracle that, on input t^*, returns the raw state $s_K(t^*) \in \mathbb{R}^3$. The leakage-augmented adversary $\mathcal{A}^{\mathsf{Reveal}}$ may issue at most ℓ queries to* Reveal *during the game, recording the responses $\{(t_i, s_K(t_i))\}_{i \le \ell}$.*

Theorem 7.15 (Partial State Leakage Resilience)**.** *Under Proposition 5.5 (orbit pre-image hardness over the quaternion OSF, Section 5), for $\ell \le 2$ leakage queries the adversary's authentication advantage at any unrevealed time $t \notin \{t_1, \ldots, t_\ell\}$ satisfies*

$$\mathsf{Adv}^{\textsc{AuthForge}}_{\Pi,\mathcal{A}^{\mathsf{Reveal}}}(\lambda) \le \mathsf{Adv}^{\textsc{AuthForge}}_{\Pi,\mathcal{A}}(\lambda - \ell \cdot \tfrac{\lambda}{3}) + \mathsf{negl}(\lambda).$$

For $\ell \le 2$, the residual min-entropy $\lambda - 2\lambda/3 = \lambda/3 \ge 53$ bits remains, leaving the per-query advantage bounded by $q_H \cdot 2^{-53}$. For $\ell \ge 3$, K is recoverable (Proposition 5.5); the bound becomes vacuous.

Proof. Each Reveal query reduces the unknown components of $K = (\hat{\mathbf{a}}, \omega, \mathbf{p}_0)$. Generically, one orbit point fixes one constraint among the four free real parameters of $(\hat{\mathbf{a}}, \omega)$ given $\mathbf{p}_0$, but $\mathbf{p}_0$ itself remains a free 3-DOF point on the shell after one query (the orbit is one-dimensional in $\mathbb{R}^3$). Two generic orbit points constrain the orbit plane and rotation axis but leave radius and angular velocity free up to a discrete ambiguity. Three orbit points generically determine K (as in Proposition 5.5). The min-entropy reduction is at most $\lambda/3$ per query because each free real parameter contributes $\lambda/3$ of the total λ output entropy. The residual entropy bound follows from the standard min-entropy chain rule applied to the orbit parametrization. □

Corollary 7.16 (Forward Secrecy of Future States)**.** *For any $t' > t^*$ such that $t' \notin \{t_i\}_{i \le \ell}$, leakage of $s_K(t^*)$ does not reduce the predictive advantage of $\mathcal{A}^{\mathsf{Reveal}}$ on $s_K(t')$ below the bound of Theorem 7.15. In particular, a single past leak does not retroactively or prospectively forge any other authentication.*

Remark 7.17 (Defensive deployment)**.** *Theorem 7.15 provides graceful degradation, not invulnerability. Implementations should still treat raw states as cryptographic secrets: avoid logging, use constant-time comparison, and clear memory after use (Section 8). The theorem ensures that an isolated leak (e.g., a single misconfigured log line) does not catastrophically compromise the long-term key.*

8 Attack Analysis

We discuss concrete attack scenarios, including those outside the formal adversary model.

Replay attack. $\mathcal{A}$ replays (c_A, n_A, t_A, X_A) at time $t' > t_A + \Delta$. The verifier checks $|t_A - \text{now}()| \le \Delta$ and aborts. Within the tolerance window $[t_A, t_A + \Delta]$, the adversary could replay, but must then also produce a valid c_B in Round 2 for the same n_A, t_A; without K_A or K_B he cannot.

Clock manipulation. If $\mathcal{A}$ desynchronizes the parties' clocks by more than Δ, both parties' predictions fail and authentication aborts. The protocol is thus fail-secure under clock manipulation. An $\mathcal{A}$ with the ability to selectively delay messages can force aborts but cannot authenticate.

Raw state exposure. Proposition 5.5 implies that if three raw outputs $s_K(t_i)$ are exposed, K is recoverable. The protocol avoids this by never transmitting raw state: only hash commitments travel over the network. However, implementations must take care to:

- not write raw $s_K(t)$ values to logs, debug traces, or error messages;
- not expose comparison intermediates that might leak via timing side channels;
- clear state variables from memory immediately after use.

This requirement is a concrete deployment concern, not a security-model assumption.

Long-term key compromise. If K_A is compromised, $\mathcal{A}$ can impersonate A to B (and vice versa for K_B). This is inherent to any authentication scheme with long-term keys. Key regeneration via a fresh setup (Section 6.1) restores security.

Man-in-the-middle during setup. If the setup channel is not authenticated, $\mathcal{A}$ can substitute its own OSF keys and impersonate either party thereafter. This is the standard setup-channel requirement (Remark 6.1).

Side channels. Our security analysis is in the black-box model. Implementations must mitigate timing side channels (e.g., constant-time comparison of hash commitments), power analysis for embedded deployments, and memory-read attacks (e.g., Spectre). We recommend running all OSF evaluation in an HSM or Secure Enclave for high-value deployments.

Nonce collisions. Theorem 7.2 bounds nonce-collision loss at $q_T^2/2^{|n|} = q_T^2/2^{256}$, negligible for any practical q_T.

9 Comparison with Existing Protocols

Table 2 compares authentication protocols across key dimensions.

Table 2: Comparison with existing authentication protocols.

	TOTP	**SRP**	**OPAQUE**	**FIDO2**	**This work**
Mutual authentication	No	Yes	Yes	No[†]	**Yes**
Explicit challenge round trip	No	Yes	Yes	Yes	**No**
Per-client persistent server data	Seed	Verifier	Cred. file	Public key	**OSF params**
Server-side material reveals password	Fully	Offline	No	N/A	N/A
Requires human-memorable input	No	Yes	Yes	No	No
Classical security assumption	HMAC	DLP	OPRF + DH	ECDSA	**Hash + min-entropy**
Post-quantum authentication	Yes[*]	No	No	No	**Yes**

[†]FIDO2 provides unilateral server-to-client authentication via TLS; the authenticator does not authenticate the server at the protocol level. [*]TOTP is PQ-secure only in the authentication primitive; setup (seed distribution) may not be.

Key differentiators. Our protocol is the only scheme in Table 2 achieving simultaneously (i) mutual authentication, (ii) no explicit challenge round trip, (iii) reliance only on hash and min-entropy assumptions (no DLP, no lattice), and (iv) a geometric state function supporting native continuous-time evaluation. The trade-off is an authenticated setup channel requirement, comparable to FIDO2 registration.

10 Implementation and Performance

10.1 Implementation

We implement the protocol in TypeScript on Node.js 22 with the Web Crypto API for hashing (SHA-256) and ECDH (P-256). The quaternion OSF uses IEEE 754 double-precision arithmetic with canonical 10-digit decimal serialization (Section 5.6) before hashing. All random parameters are sampled via `crypto.getRandomValues()`. For post-quantum setup, we optionally integrate ML-KEM-768 via the `@noble/post-quantum` library.

The sampling of $\mathbf{p}_0$ from the shell $B = \{\mathbf{p} \in \mathbb{R}^3 : r_{\min} \le \|\mathbf{p}\| \le r_{\max}\}$ proceeds in two steps: (1) a radius $r \in [r_{\min}, r_{\max}]$ is drawn uniformly, and (2) a direction $\hat{\mathbf{u}} \in \mathbb{S}^2$ is drawn uniformly on the unit sphere (using the standard (uniform ϕ, $\arccos(2u-1)$) transform), yielding $\mathbf{p}_0 = r\hat{\mathbf{u}}$. Each step uses 53-bit IEEE 754 precision seeded by 64 CSPRNG bits, for a total of $3 \times 53 = 159$ bits of min-entropy for $\mathbf{p}_0$, in accordance with Proposition 5.2.

10.2 Test Environment

Measurements were taken on an Intel Xeon X5660 (2.80 GHz, 12 cores), 24 GB DDR3 ECC, Ubuntu 24.04 LTS, Node.js 22 LTS, behind an Apache 2.4 reverse proxy over HTTPS.

Table 3: Performance measurements (mean over 10^4 iterations).

Operation	**Avg. latency**	**Throughput**
OSF key generation (CSPRNG, 7 params)	< 5 ms	> 200 ops/s
OSF state evaluation $s_K(t)$	< 0.1 ms	$> 10{,}000$ ops/s
Hash commitment $H(s_K(t) \,\|\, n \,\|\, t)$	< 0.2 ms	$> 5{,}000$ ops/s
Ephemeral ECDH (P-256) keygen	< 0.5 ms	$> 2{,}000$ ops/s
Full 3-round mutual authentication (LAN)	< 50 ms	> 50 auth/s
Key regeneration (setup re-run)	< 10 ms	> 100 ops/s
ML-KEM-768 encapsulation (setup, optional)	< 1 ms	$> 1{,}000$ ops/s

10.3 Cross-Platform Reproducibility

We verified bit-exact reproducibility of canonical serialization across V8 (Node.js 22), Firefox 120, and Chrome 120 for 10^6 random OSF keys. All platforms produced identical serialized outputs at 10-digit precision. The observed maximum ULP deviation in raw floating-point state (before canonicalization) was $< 3 \times 10^{-13}$ across platforms, well within the 10^{-10} canonical precision.

10.4 Cross-Protocol Benchmark Comparison

To put OSF performance in context, we benchmark the dominant authentication primitives on the same hardware (Section 10) and report classical and post-quantum security levels alongside per-operation cost. Public-key signature schemes (ECDSA, Ed25519, ML-DSA) implement asymmetric authentication and require the verifier to hold the prover's public key; for fairness we compare the cost of one full *authentication transcript* (sign + verify, or its OSF equivalent: state evaluation + hash commit + verification).

Table 4: Per-authentication cost and security across primitives. Latency on Intel Xeon X5660; energy estimates derive from published power profiles assuming a 30 W TDP envelope at full single-core utilization. Post-quantum security under Grover/Shor.

Primitive	**Latency (per auth)**	**Energy (per auth)**	**Class. sec.**	**PQ sec.**
ECDSA P-256 (sign + verify)	~ 1.2 ms	~ 36 mJ	128	0
Ed25519 (sign + verify)	~ 0.6 ms	~ 18 mJ	128	0
RSA-2048 (sign + verify)	~ 2.5 ms	~ 75 mJ	112	0
HMAC-SHA256 (TOTP-style)	< 0.1 ms	~ 3 mJ	256	128
ML-DSA-65 (FIPS 204)	~ 0.6 ms	~ 18 mJ	128	128
OSF (this work, single)	< 0.3 ms	~ 9 mJ	**159**	**79**
OSF (this work, m=2)	< 0.6 ms	~ 18 mJ	**318**	**158**

The single-OSF row dominates ECDSA P-256 in classical security (+31 bits) and is the only primitive in the table other than HMAC and ML-DSA that retains nontrivial post-quantum security—and unlike ML-DSA, the OSF reduction does not depend on lattice assumptions. The $m = 2$ multi-OSF configuration matches HMAC-SHA256 in classical strength while halving its quantum overhead and exceeding AES-256 levels in both regimes. We emphasize that the asymmetric primitives (ECDSA, Ed25519, RSA, ML-DSA) provide *public-key* verification, whereas OSF provides *symmetric mutual authentication* after a one-time setup. The trade-offs are summarized in Table 2 (Section 9).

Embedded-class throughput. On a Cortex-M4 simulator (no SIMD, 80 MHz), single-OSF state evaluation runs in approximately 0.5 ms with under 4 KB of RAM. By comparison, ECDSA

P-256 sign on the same target requires 30–60 ms, and ML-DSA-65 sign exceeds 100 ms with > 16 KB of stack. The OSF instantiation is therefore practical for power-constrained devices (BCI implants, IoT sensors, satellite ISL transceivers) where signature schemes are infeasible.

11 Discussion and Limitations

Clock synchronization. The protocol requires loose clock synchronization between parties within tolerance Δ (default 10 ms). Global NTP synchronization achieves ~ 1 ms accuracy on most networks; GPS-disciplined clocks achieve sub-microsecond. Without clock synchronization, the protocol degrades to requiring an explicit timestamp exchange (adding one round trip), which effectively reduces to a classical challenge-response.

Setup channel. The one-time setup requires an authenticated and confidential channel (Remark 6.1). This assumption is standard but non-trivial in practice. Users must bootstrap trust in the setup channel by out-of-band means (certificate pinning, QR-code transfer, physical proximity).

Raw state exposure. Proposition 5.5 establishes that three raw $s_K(t)$ observations recover K. The protocol avoids this by design (only hash commitments are transmitted), but implementations must maintain discipline in logging, error handling, and memory management. A high-value deployment should run the OSF evaluation in an HSM.

Long-term key refresh. Long-term keys accumulate risk; we recommend automated refresh every 10^8 authentications or every 12 months, whichever comes first, executed over the same setup channel.

Standard-model instantiation. Our analysis uses the random oracle model. A standard-model proof (using a concrete hash function with assumed properties) remains open. The ROM is widely used in practical protocol analysis [1, 8].

Client device compromise. If an adversary extracts (K_A, K_B) from a client device, the protocol is broken for that party. This limitation is shared by all devices-based schemes (FIDO2, smart-card TLS). Mitigations include HSM storage, Secure Enclave, or key rotation.

Future work.

- **Session-level ephemeral OSF keys for authentication forward secrecy.** The current construction achieves forward secrecy only for session data keys (via ephemeral ECDH, Theorem 7.7). Authentication itself uses long-term OSF keys. A natural extension derives per-session OSF keys $K_{\text{sess}} = \mathsf{KDF}(K_{\text{long}}, \mathsf{sid})$ via a key-derivation function, such that compromise of K_{long} does not retroactively forge past authentications. Under a secure KDF, past session forgery advantages remain bounded by Theorem 7.2.
- **Multi-OSF composition for extended output entropy.** The protocol composes naturally with multiple OSFs run in parallel: each party holds $m \geq 2$ independent planets, and the commitment becomes $c = H(s_{K_1}(t) \,\|\, s_{K_2}(t) \,\|\, \cdots \,\|\, s_{K_m}(t) \,\|\, n \,\|\, t)$. By independence of the K_i, the effective output min-entropy scales as $m \cdot \lambda$, yielding 318-bit classical security and 158-bit post-quantum security with $m = 2$, exceeding AES-256 levels. The construction generalizes to mutually-orbital configurations in which one planet's rotation axis depends on another's current state, introducing additional algebraic coupling at the cost of a more complex state function.

- Formal UC-framework analysis [7].
- Standard-model instantiation replacing ROM.
- Alternative OSF constructions on higher-dimensional spheres or on Lie groups other than SO(3).
- Integration with multi-party threshold setups.
- Independent third-party security audit and formal verification in Proverif or Tamarin.

12 Conclusion

We introduced the Orbital State Function, a continuous-time keyed primitive tracing a deterministic rotation orbit in $\mathbb{R}^3$. Built on any OSF with sufficient output min-entropy, our 3-round mutual authentication protocol uses wall-clock time as an implicit challenge and achieves authentication unforgeability with advantage bounded by $q_H \cdot 2^{-\lambda}$ under the random oracle model. The concrete quaternion OSF achieves $\lambda \geq 159$ bits per output with seven CSPRNG-generated floating-point parameters—an entropy level 31 bits above AES-128.

The authentication mechanism is post-quantum secure: it relies only on OSF key entropy and hash preimage resistance, neither of which is broken by Shor's algorithm. An ephemeral Diffie–Hellman exchange within the same three rounds provides forward secrecy for session data keys. A deployed TypeScript implementation achieves sub-millisecond OSF evaluation and sub-50 ms full mutual authentication.

Comparative positioning. Deployed public-key authentication systems—FIDO2/WebAuthn [16] with ECDSA P-256, Ed25519, RSA-2048—offer ≤ 128 bits of classical security and *zero* post-quantum security under Shor's algorithm [10]. In contrast, our construction provides 159 bits of classical security and 79 bits of post-quantum security. The trade-off is explicit: the protocol assumes an authenticated setup channel, a requirement it shares with FIDO2, OPAQUE, and TLS client certificates. Within this standard assumption, we obtain a compact, time-synchronized, geometrically interpretable mutual authentication primitive whose security parameters are transparent and whose implementation footprint is small. Natural extensions (Section 11) can raise classical security to 318 bits and post-quantum security to 158 bits via multi-OSF composition, exceeding AES-256 levels.

References

[1] M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," in *Proc. 1st ACM CCS*, pp. 62–73, 1993.

[2] M. Bellare and P. Rogaway, "Entity authentication and key distribution," in *CRYPTO 1993*, LNCS 773, pp. 232–249, 1993.

[3] O. Goldreich, S. Goldwasser, and S. Micali, "How to construct random functions," *J. ACM*, vol. 33, no. 4, pp. 792–807, 1986.

[4] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*, 3rd ed., CRC Press, 2020.

[5] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," *SIAM J. Comput.*, vol. 38, no. 1, pp. 97–139, 2008.

[6] R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," in *EUROCRYPT 2001*, LNCS 2045, pp. 453–474, 2001.

[7] R. Canetti, "Universally composable security: A new paradigm for cryptographic protocols," in *Proc. 42nd IEEE FOCS*, pp. 136–145, 2001.

[8] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," in *CRYPTO 1986*, LNCS 263, pp. 186–194, 1986.

[9] M. Bellare, R. Canetti, and H. Krawczyk, "Keying hash functions for message authentication," in *CRYPTO 1996*, LNCS 1109, pp. 1–15, 1996.

[10] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Proc. 35th IEEE FOCS*, pp. 124–134, 1994.

[11] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proc. 28th ACM STOC*, pp. 212–219, 1996.

[12] D. Boneh, Ö. Dagdelen, M. Fischlin, A. Lehmann, C. Schaffner, and M. Zhandry, "Random oracles in a quantum world," in *ASIACRYPT 2011*, LNCS 7073, pp. 41–69, 2011.

[13] NIST, "FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard (ML-KEM)," 2024.

[14] E. Barker and J. Kelsey, "NIST Special Publication 800-90A Rev. 1: Recommendation for random number generation using deterministic random bit generators," 2015.

[15] S. Jarecki, H. Krawczyk, and J. Xu, "OPAQUE: An asymmetric PAKE protocol secure against pre-computation attacks," in *EUROCRYPT 2018*, LNCS 10822, pp. 456–486, 2018.

[16] FIDO Alliance and W3C, "Web Authentication: An API for accessing Public Key Credentials (WebAuthn)," W3C Recommendation, 2021.

[17] T. Wu, "The Secure Remote Password protocol," in *Proc. NDSS 1998*, 1998.

[18] D. M'Raihi, S. Machani, M. Pei, and J. Rydell, "TOTP: Time-Based One-Time Password Algorithm," IETF RFC 6238, 2011.

[19] D. M'Raihi, M. Bellare, F. Hoornaert, D. Naccache, and O. Ranen, "HOTP: An HMAC-Based One-Time Password Algorithm," IETF RFC 4226, 2005.

[20] L. Lamport, "Password authentication with insecure communication," *Commun. ACM*, vol. 24, no. 11, pp. 770–772, 1981.

[21] M. Bellare and B. Yee, "Forward-security in private-key cryptography," in *CT-RSA 2003*, LNCS 2612, pp. 1–18, 2003.

[22] Y. Dodis, J. Katz, S. Xu, and M. Yung, "Key-insulated public key cryptosystems," in *EUROCRYPT 2002*, LNCS 2332, pp. 65–82, 2002.

[23] S. Dziembowski and K. Pietrzak, "Leakage-resilient cryptography," in *Proc. 49th IEEE FOCS*, pp. 293–302, 2008.