# MULTIVERS KeyGen

Whitepaper v1.0

## The First Human-Entropy Cryptographic Key Generation Protocol

*Published: June 14, 2026 · Author: Quentin / spaceq01 · CONFIDENTIAL v1.0*

**ABSTRACT**

We introduce **MULTIVERS KeyGen**, a cryptographic key generation system that derives entropy from real-time human gameplay interactions. The system implements a novel **Proof of Play (PoP)** protocol that captures player actions, high-resolution timestamps, and hardware-level random values, then feeds them into a SHA-256 pipeline to produce a unique 256-bit key. We argue that human-in-the-loop entropy provides meaningful security advantages over pure-software CSPRNGs in specific use cases, particularly for collectible credentials, shared secret exchange, and narrative-linked authentication. The system is implemented as a standalone HTML5 application with zero external dependencies, fully auditable by end users.

# 1. Introduction

The security of any cryptographic system depends fundamentally on the quality of its random number generation. Modern systems rely on CSPRNGs seeded by hardware entropy sources (CPU jitter, thermal noise, interrupt timing). While mathematically robust, these systems share a critical property: they are entirely opaque to the user. The user has no visibility into, participation in, or emotional connection to the key generation process.

MULTIVERS KeyGen proposes a complementary approach: **human entropy**. By embedding key generation within a strategy game, we create a system where the user's own decisions, reactions, and timing contribute directly to the cryptographic output. The result is a key that is not only mathematically strong, but experientially unique — tied to a specific moment, a specific game, and specific human choices.

# 2. The Proof of Play Protocol

## 2.1 Design Principles

- **Non-determinism**: The key cannot be reproduced without exact knowledge of every player action and its timestamp
- **Transparency**: The entire pipeline is visible and auditable in a single HTML file
- **Independence**: No external servers, no network transmission during generation
- **Composability**: The entropy log can be exported and fed into downstream KDFs
- **Human verifiability**: A player can intuitively understand they created the key

## 2.2 Entropy Pipeline

The pipeline operates in three phases:

1. **Collection** — Every game event triggers addEntropy(), recording event type, player, high-res timestamp (performance.now()), and a CSPRNG value (crypto.getRandomValues()).
2. **Accumulation** — Events are serialized as JSON and accumulated in an array (entropyLog) throughout the game session. A typical game generates 80-300 entropy events.
3. **Derivation** — At game end, all events are joined into a single string, appended with a final timestamp and 128-bit CSPRNG seed, UTF-8 encoded, and passed to crypto.subtle.digest('SHA-256') — the browser's native cryptographic implementation.

# 3. Security Analysis

## 3.1 Entropy Estimation

A typical 30-turn game generates approximately:

- 30 × dice roll events (6 possible values + CSPRNG + timestamp)
- 30 × move events (66 possible cells × 2 galaxies × high-res timestamp)
- 5-10 × combat events (6 × 6 die combinations + timestamp)
- 3-8 × conquest events (15 planet names × galaxy × timestamp)
- 0-4 × wormhole events
- 1 × game_end event
- 4 × 32-bit CSPRNG values injected at finalization

The combined entropy, before hashing, exceeds 512 bits from game events alone, plus 128 bits from finalization CSPRNG. SHA-256 condenses this into a uniformly distributed 256-bit output.

## 3.2 Attack Vectors

| Attack | Risk | Mitigation |
|---|---|---|
| Replay attack | Infeasible | Attacker would need exact timestamps (sub-ms precision) + CSPRNG values |
| Brute force | Infeasible | 2^256 search space — computationally impossible |
| Side-channel | Low risk | Local execution, no network transmission during generation |
| Social engineering | Reduced | Emotional attachment to game makes phishing less effective |
| Implementation bug | Mitigated | Uses browser's native crypto.subtle — battle-tested |

## 4. Competitive Landscape

| System | Entropy Source | User Visibility | Emotional Bond | Audit |
|---|---|---|---|---|
| **MULTIVERS KeyGen** | **Human gameplay + CSPRNG** | **Full (open source)** | **High** | **Complete** |
| Bitwarden | OS CSPRNG | None | None | Partial |
| 1Password | OS CSPRNG | None | None | None |
| KeePass | OS CSPRNG | None | None | Full (OSS) |
| HSM (hardware) | Hardware noise | None | None | Certified |
| Dice (physical) | Physical dice | Full | Medium | Manual |
| MetaMask | OS CSPRNG + BIP-39 | Partial | Low | Partial |

## 5. Business Model & Market Opportunity

MULTIVERS KeyGen targets three distinct market segments with a freemium SaaS model:

| Tier | Price | Features | Target |
|---|---|---|---|
| Free | 0€/mo | 5 keys/month · TTL max 24h · No API · QR export | Individual users |
| Pro | 9€/mo | Unlimited keys · All TTL · API 10k calls/mo · File export | Developers / Creators |
| Business | 49€/mo | Unlimited API · Verification endpoint · SLA · White label | Companies |
| Enterprise | Custom | Private deployment · Audit · FIPS compliance · Custom integration | Finance / Gov |

### Revenue milestones:

- **100 Pro subscribers** → 900€/month MRR — First viability threshold
- **500 Pro + 20 Business** → 5,480€/month MRR — Seed funding repayment pace
- **2,000 Pro + 100 Business + 5 Enterprise** → ~27k€/month MRR — Series A readiness

## 6. Development Roadmap

| Phase | Timeline | Goal | Actions |
|---|---|---|---|
| Phase 1 | Weeks 1-2 | IP Protection | INPI Soleau · GitHub private · Timestamp registration |
| Phase 2 | Weeks 2-6 | Technical Foundation | Whitepaper · Landing page · Backend API (Supabase + Netlify Functions) |
| Phase 3 | Weeks 6-10 | Market Validation | Product Hunt launch · Reddit community · Email list (target 500+) |
| Phase 4 | Weeks 10-16 | Fundraising | Bpifrance application · Angel outreach · Accelerator applications |
| Phase 5 | Month 4-12 | Product Growth | Mobile app · SDK · Enterprise features · Partnerships |

## 7. Conclusion

MULTIVERS KeyGen introduces a genuinely novel approach to cryptographic key generation: one where the human user is not a passive recipient of a random string, but an active participant whose choices, timing, and gameplay shape the cryptographic output. The Proof of Play protocol is, to our knowledge, the first published implementation of human-gameplay-derived cryptographic entropy.

The system's unique properties — emotional traceability, full auditability, zero-dependency implementation, and dual-player shared secret capability — position it in a market segment that no existing solution addresses. Combined with a scalable SaaS business model and clear IP protection, MULTIVERS KeyGen represents a compelling investment opportunity at the intersection of gaming, cryptography, and digital identity.

| | |
|---|---|
| **Contact** | spaceq01 — spaceq01.itch.io |
| **Bitcoin** | bc1qddluvgymexjf0j38hvjlnze5fs8jmxluwksruc |
| **Document** | MULTIVERS KeyGen Whitepaper v1.0 — June 14, 2026 |
| **Status** | Seeking seed funding · Open to strategic partnerships |