

PostQuantum DualUSB Token Library v0.1.1

Release Classification: Documentation Enhancement and Structural Reorganization

This release represents a comprehensive documentation revision initiative aimed at improving the technical clarity, structural coherence, and academic rigor of the project's supporting materials. The primary objective is to establish a unified, authoritative information repository that enhances both developer comprehension and implementation fidelity.

Release Summary

Version 0.1.1 constitutes a documentation-focused maintenance release with no modifications to the core cryptographic implementation or API surface. This release addresses critical documentation fragmentation issues identified in version 0.1.0 and introduces enhanced visual documentation through standardized architectural diagrams.

Technical Documentation Enhancements

Documentation Consolidation and Standardization

Primary Documentation Unification: The main `README.md` has been comprehensively restructured to serve as the canonical reference document for the library. This reorganization eliminates information redundancy, resolves documentation inconsistencies, and establishes a single source of truth for technical specifications, usage patterns, and security considerations.

Architectural Visualization: Formal architectural diagrams have been introduced using Mermaid.js notation to provide precise visual representations of:

- System component interactions and dependencies
- Cryptographic data flow pathways and transformations
- Security boundary definitions and trust domains
- Key derivation hierarchies and lifecycle management
- Authentication and authorization sequence protocols

Threat Model Formalization: A rigorous threat modeling section has been incorporated, providing formal definitions of:

- Attack surface analysis and threat actor capabilities
- Security properties and cryptographic guarantees
- Defense mechanisms and mitigation strategies
- Assumptions regarding adversary models (Dolev-Yao, post-quantum adversaries)
- Compliance mapping to established security frameworks (NIST, ISO 27001)

Project Roadmap Integration: Future development objectives and research directions have been systematically documented within the primary `README`, including:

- Planned cryptographic algorithm upgrades

- Hardware security module (HSM) integration pathways
- Formal verification initiatives
- Performance optimization strategies
- Extended platform support objectives

PyPI Distribution Synchronization: The `PYPI_README.md` has been revised to maintain semantic equivalence with the primary documentation while adhering to Python Package Index formatting constraints and best practices.

Documentation Maintenance and Hygiene

Changelog Maintenance: The `CHANGELOG.md` has been updated following the Keep a Changelog specification (version 1.1.0), documenting all documentation modifications in a structured "Unreleased" section prior to formal version tagging.

Redundancy Elimination: The standalone `ROADMAP.md` file has been deprecated and removed to prevent documentation drift and maintenance burden. Its content has been migrated and integrated into the appropriate sections of the primary `README`.

Rationale and Impact

Comprehensive, accurate, and formally structured documentation is a foundational requirement for cryptographic security libraries. This documentation enhancement provides several critical benefits:

- **Enhanced Comprehension:** Developers can systematically understand the library's architectural design principles, implementation decisions, and operational characteristics through unified, non-contradictory documentation.
- **Implementation Correctness:** Clear integration guidelines, explicit security assumptions, and documented best practices reduce the likelihood of implementation errors that could compromise security properties.
- **Trust Establishment:** Transparent documentation of threat models, security guarantees, and cryptographic design decisions enables security researchers and practitioners to perform informed risk assessments and trust evaluations.
- **Academic and Professional Adoption:** Formal documentation standards facilitate citation, peer review, and integration into security-critical production environments where documentation quality directly impacts adoption decisions.

Installation and Distribution

This release contains no modifications to the library's functional implementation, API interfaces, or cryptographic primitives. The codebase remains semantically identical to version 0.1.0. Installation procedures are unchanged:

```
pip install pqcdualusb==0.1.1
```

For development installations from source:

```
git clone https://github.com/Johnsonajibi/PostQuantum-DualUSB-Token-Library.git
cd PostQuantum-DualUSB-Token-Library
pip install -e .
```

Version Compatibility

- **Python Compatibility:** Requires Python 3.8 or higher
- **Dependency Stability:** All cryptographic dependencies (pqcrypto, cryptography) remain at previously specified versions
- **API Stability:** Full backward compatibility maintained with version 0.1.0
- **Breaking Changes:** None

References and Resources

- **PyPI Distribution:** <https://pypi.org/project/pqcdualusb/>
- **Source Repository:** <https://github.com/Johnsonajibi/PostQuantum-DualUSB-Token-Library>
- **Issue Tracking:** <https://github.com/Johnsonajibi/PostQuantum-DualUSB-Token-Library/issues>
- **Security Policy:** SECURITY.md - Includes responsible disclosure procedures
- **Technical White Paper:** WHITEPAPER_COMPLETE.md - Comprehensive technical specification
- **Contributing Guidelines:** CONTRIBUTING.md
- **License:** MIT License - See LICENSE

Verification and Integrity

Release integrity can be verified through the following mechanisms:

- **Git Tag Signature:** Release v0.1.1 is cryptographically signed
- **PyPI Hash Verification:** SHA-256 checksums available via PyPI
- **Source Code Audit:** Complete source available for inspection

Citation

For academic citations, please use the following format:

```
@software{pqcdualusb2025,
  author = {Johnson, Ajibi},
  title = {PostQuantum DualUSB Token Library},
  version = {0.1.1},
  year = {2025},
  url = {https://github.com/Johnsonajibi/PostQuantum-DualUSB-Token-Library},
  note = {Post-quantum cryptographic library for dual-device authentication}
}
```

Acknowledgments

We acknowledge the broader cryptographic research community for establishing the theoretical foundations and security analysis methodologies that inform this implementation. Special recognition to NIST for the post-quantum cryptography standardization effort and the open-source cryptographic library maintainers whose work forms the foundation of this project.

Release Metadata

- **Release Date:** October 18, 2025
- **Release Type:** Maintenance (Documentation Enhancement)
- **Semantic Version:** 0.1.1
- **Git Commit:** [To be determined upon release]
- **Release Manager:** Johnson Ajibi
- **Documentation Standard:** IEEE Software Documentation Standard (IEEE 1063-1987)