The Journal of Open Source Software

secp256k1-frost: Minimal-Dependency RFC-9591 FROST Threshold Schnorr Signatures for libsecp256k1

Matteo Nardelli [1] **and Antonio Muci** [1]

1 Bank of Italy

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor:

Submitted: 26 February 2026
Published: unpublished

Summary

Flexible Round-Optimized Schnorr Threshold Signatures (FROST), standardized in IETF RFC 9591 (Connolly et al., 2024), enable a set of participants to collaboratively produce Schnorr signatures while distributing signing authority across multiple parties. The FROST protocol was originally introduced by Komlo and Goldberg (Komlo & Goldberg, 2021). As threshold signatures gain adoption in distributed infrastructures and blockchain systems, there is a growing demand for implementations that integrate with existing high-assurance cryptographic libraries.

secp256k1-frost is an open-source implementation of FROST written in portable C (C89) and designed as an extension module for the widely used `libsecp256k1` library (Wuille & contributors, 2021). The project implements the protocol logic defined in RFC-9591 while reusing the elliptic-curve arithmetic and engineering practices of `libsecp256k1`. The implementation emphasizes minimal dependencies, auditability, and interoperability with secp256k1-based software, making it suitable for experimentation, research, and integration into performance-critical systems.

Statement of Need

Threshold signatures are increasingly used in distributed systems, digital asset infrastructures, and secure multiparty applications. Despite the standardization of FROST, developers integrating threshold signing into existing secp256k1-based environments face practical challenges, including dependency-heavy implementations and limited availability of low-level C libraries aligned with established cryptographic toolchains.

Existing FROST implementations are often written in higher-level languages or rely on broader cryptographic frameworks. In contrast, `secp256k1-frost` focuses on a minimal C implementation aligned with the engineering philosophy of `libsecp256k1`, enabling standards-compliant threshold signing without introducing additional cryptographic dependencies.

The library addresses the needs of developers and researchers requiring:

- portable and minimal-footprint cryptographic components,
- compatibility with secp256k1-based infrastructures,
- deterministic builds and auditability,
- experimentation with RFC-9591-compliant threshold signing workflows.

State of the Field

Several open-source implementations of FROST have emerged alongside the standardization process, spanning multiple programming languages and elliptic-curve ciphersuites. The CFRG

draft repository maintains a list of existing implementations, including reference prototypes in Sage, Rust-based libraries supporting multiple curves (e.g., ristretto255, ed25519, and secp256k1), and implementations in languages such as Go (Connolly et al., 2024). Many of these projects emphasize modularity and generic protocol abstractions, enabling experimentation across different curves and deployment environments.

Notably, Rust implementations such as the Zcash Foundation FROST library provide flexible multi-ciphersuite support and have contributed significantly to the maturation of the ecosystem (Zcash Foundation, 2023). However, these implementations typically rely on higher-level cryptographic frameworks and introduce additional dependency layers that may complicate integration into minimal or embedded environments.

In contrast, `secp256k1-frost` explores a complementary design space by extending the existing libsecp256k1 ecosystem (Wuille & contributors, 2021) rather than introducing a standalone framework. This decision reflects a different engineering goal: evaluating how standardized threshold Schnorr signatures can be integrated directly into a widely deployed, high-assurance elliptic-curve library. By prioritizing minimal dependencies, C89 portability, and reuse of audited primitives, the project provides a distinct systems-oriented contribution that complements existing multi-language FROST implementations.

Software Design

The implementation is developed as an extension module within the `libsecp256k1` codebase (Wuille & contributors, 2021). Rather than re-implementing elliptic-curve primitives, `secp256k1-frost` builds upon the existing, well-reviewed group and scalar operations provided by the library. The project implements the protocol logic required by RFC-9591, including commitment handling, signing share computation, and signature aggregation.

The repository follows a structure consistent with the secp256k1 ecosystem, facilitating integration into existing projects and enabling familiar build workflows. The codebase is written in portable C89 to maximize platform compatibility. Testing and validation are performed through RFC test vectors, unit tests, and continuous integration workflows to ensure interoperability and correctness.

Design Goals and Engineering Decisions

The development of `secp256k1-frost` emphasizes engineering decisions aligned with high-assurance cryptographic software:

- **Minimal dependency footprint:** no external cryptographic libraries beyond `libsecp256k1`.
- **Portability:** adherence to the C89 standard supports diverse platforms and toolchains.
- **Auditability:** reuse of existing cryptographic primitives reduces complexity and leverages prior security reviews.
- **Interoperability:** seamless integration with secp256k1-based infrastructures, including blockchain and distributed systems applications.
- **Deterministic builds:** suitable for environments requiring reproducibility and controlled deployment pipelines.

These design goals differentiate the library from higher-level implementations by prioritizing integration into established low-level cryptographic ecosystems.

Use Cases

`secp256k1-frost` enables experimentation and development of threshold Schnorr signing workflows in contexts where a small binary footprint and minimal external dependencies are required. Example scenarios include:

- blockchain clients and wallet infrastructures,
- hardware-backed or embedded signing environments,
- research prototypes exploring threshold cryptographic protocols,
- performance-sensitive distributed systems.

Security Considerations

FROST requires careful handling of nonce generation, participant coordination, and commitment verification. The implementation follows the security guidance provided in RFC 9591 and documents assumptions and safe usage patterns in the project documentation.

The library is designed primarily for experimentation and research use while adhering to secure engineering practices consistent with libsecp256k1.

Research Impact Statement

The project contributes to ongoing research and engineering efforts around standardized threshold cryptography by providing a reproducible and inspectable implementation aligned with RFC-9591. Its integration within the secp256k1 ecosystem enables researchers to evaluate threshold signing workflows in environments representative of real-world deployments.

The repository includes examples, build tooling, and testing infrastructure that support reproducibility and experimentation. The software is intended to serve as a reference point for future studies on threshold signatures, distributed signing protocols, and performance-oriented cryptographic engineering.

AI usage disclosure

Generative AI tools were used exclusively to assist in drafting and refining portions of the manuscript text. All AI-assisted text was carefully reviewed, edited, and validated by the authors to ensure accuracy, clarity, and consistency with the software and its research objectives. The software implementation, design decisions, testing infrastructure, documentation, and all technical artifacts were developed solely by the authors.

Availability

- Source code: https://github.com/bancaditalia/secp256k1-frost/
- License: MIT License

Acknowledgements

This project builds upon the design of the FROST protocol standardized by the IETF Crypto Forum Research Group (CFRG) (Connolly et al., 2024) and the engineering work behind the libsecp256k1 library (Wuille & contributors, 2021).

References

Connolly, D., Komlo, C., Goldberg, I., & Wood, C. A. (2024). *The flexible round-optimized schnorr threshold (FROST) protocol for two-round schnorr signatures*. RFC 9591, IETF. https://doi.org/10.17487/RFC9591

Komlo, C., & Goldberg, I. (2021). FROST: Flexible round-optimized schnorr threshold signatures. In O. Dunkelman, M. J. Jacobson Jr., & C. O'Flynn (Eds.), *Selected areas in cryptography* (pp. 34–65). Springer International Publishing. ISBN: 978-3-030-81652-0

Wuille, P., & contributors. (2021). *libsecp256k1: Optimized c library for elliptic curve operations on curve secp256k1*. https://github.com/bitcoin-core/secp256k1.

Zcash Foundation. (2023). *ZF FROST (flexible round-optimised schnorr threshold signatures)*. https://github.com/ZcashFoundation/frost.