

Plan: Professional README for tor-go

Contents

Problem	2
Research Findings	3
Slices	4
Not Doing	6

Instructions for implementing agents: - Work through slices in order - Use `/build [slice description]` to implement each slice — this includes code review and security assessment per slice - After EACH slice: check it off (-) in this file and run `~/claude/md2pdf.sh` on this file to update the PDF - If a slice is larger than expected, split it further before building - If you discover new slices needed, add them with checkboxes and regenerate the PDF

Scope: Feature

Problem

tor-go has no README.md, no LICENSE, no CI workflow, and no SECURITY.md. Without these, the project looks abandoned on GitHub and developers cannot evaluate or trust it. The README is the single most important file for open-source adoption — it needs to follow established Go library patterns: short, factual, confident.

Research Findings

Full research report: `research/readme-best-practices.md`

Key findings from analyzing 15+ Go library READMEs:

- **Under 100 lines.** Library READMEs are landing pages, not documentation (quic-go: ~100 lines, wireguard-go: ~50 lines)
- **3 badges exactly:** Go Reference, CI Status, Go Report Card. No license badge (0/13 top Go projects use one)
- **"Pure Go" goes in the H1 title** (quic-go pattern), mentioned once only
- **Lead with CLI quick start** (2-3 shell lines), defer Go API to godoc
- **No disclaimers, no apologies, no Tor explainers, no comparison tables, no roadmaps** — all destroy trust in security projects
- **Include a security reporting channel** — every credible security project has one

— Confidence level: High (based on primary evidence from actual README files, not secondary advice)

Slices

- **F-1: Create LICENSE file**

- **What:** Add an MIT LICENSE file to the repository root. This is a prerequisite for the README (which references the license) and for Go Report Card (which checks for a license).
- **Files/areas:** LICENSE
- **Acceptance criteria:**
 - MIT license file exists at repository root
 - Copyright holder is set to the repository owner
 - Year is 2026
- **Dependencies:** None
- **Risks:** Need to confirm with user which license they want. Default to MIT (most common for Go libraries in this space — wireguard-go, bine, age all use MIT or similar permissive licenses). Ask the user before creating.

- **F-2: Create GitHub Actions CI workflow**

- **What:** Add a basic CI workflow that runs `go test ./...` on push and PR. This is a prerequisite for the CI badge in the README to be functional.
- **Files/areas:** .github/workflows/test.yml
- **Acceptance criteria:**
 - Workflow triggers on push to main and on pull requests
 - Runs `go test ./...` with Go 1.26
 - Workflow name matches what the README badge references
- **Dependencies:** None
- **Risks:** Minimal. Standard Go CI workflow. Keep it simple — just `go test`, no linting or extra steps.

- **F-3: Create SECURITY.md**

- **What:** Add a brief security policy file with a reporting channel. Research shows every credible security project has one, and the README will link to it.
- **Files/areas:** SECURITY.md
- **Acceptance criteria:**
 - File exists with instructions for reporting security vulnerabilities
 - Mentions GitHub Security Advisories as the reporting mechanism
 - Brief (under 20 lines)
- **Dependencies:** None
- **Risks:** None. Standard file.

- **F-4: Write README.md**

- **What:** Create the README following the research blueprint. Under 100 lines, factual tone, quic-go/wireguard-go pattern. Title with “pure Go”, 3 badges, feature bullets, CLI quick start, links to godoc, security section, license line.
- **Files/areas:** README.md
- **Acceptance criteria:**

- Under 100 lines of markdown
 - H1 title includes “pure Go” (quic-go pattern)
 - Exactly 3 badges: Go Reference, CI Status, Go Report Card
 - Feature bullet list with 5-8 specific protocol capabilities
 - CLI quick start section showing `go install` + `tor-client` + `curl` verification with `{"IsTor":true}` output
 - Library section linking to `cmd/tor-client` as a complete example and to `pkg.go.dev`
 - Installation section with `go get`
 - Security section linking to `SECURITY.md`
 - License section (one line)
 - NO disclaimers, apologies, Tor explainers, comparison tables, roadmaps, TODO lists, or Go API code examples
 - NO more than 2 paragraphs of prose in any section
 - Factual, confident tone throughout
- **Dependencies:** F-1, F-2, F-3 (needs LICENSE, CI workflow, and `SECURITY.md` to exist for links/badges to be valid)
 - **Risks:** The `go install` command needs the module to be published on a Go module proxy for it to work. If the repo isn’t public yet, use `go run ./cmd/tor-client` instead. Check what works.
 - **Not doing:** Logo/branding (optional enhancement, not needed for initial `README`). Convenience API layer (separate feature).

Not Doing

- **Logo/branding** — research shows 10/13 projects have one, but it's not essential for launch. Can be added later without changing README structure.
- **Convenience API (`torgo.Dial()`)** — research identified this as the highest-impact improvement for developer adoption, but it's a library design change, not a README task. Track separately.
- **CONTRIBUTING.md** — only 3/9 surveyed projects include one. Not needed for initial launch.
- **examples/ directory** — the `cmd/tor-client` binary serves as the complete example for now.