# PROTECT
## a **P**latform for **Ro**bust **T**hr**e**shold **C**ryp**t**ography

Jason Resch, Christian Cachin, Hugo Krawczyk, Tal Rabin, Chrysa Stathakopoulou

# Outline

## Project
- Overview
- Functionality and Goals

## Architecture
- Model and Assumptions
- Tunable Security
- System Architecture

## Performance
- Single System
- Distributed Locally
- Distributed Geographically

## Demo
- Installation and Configuration
- Secrets: Generation, Refresh, Recovery
- Cryptographic Operations

## Discussion
- General Discussion
- Questions and Answers

# Project Overview

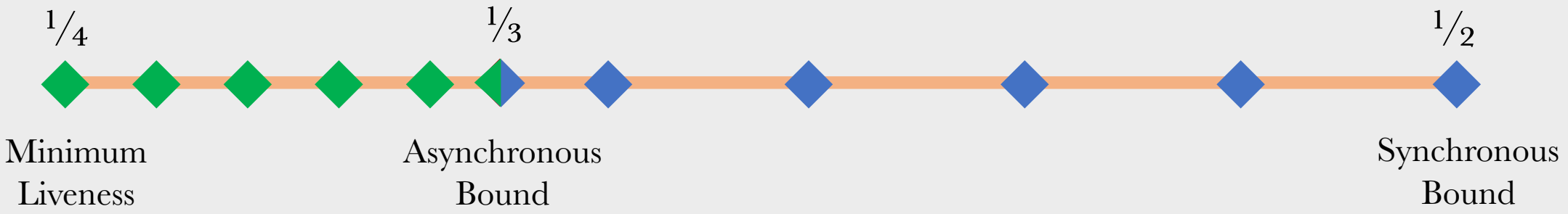
- PROTECT is an open source (MIT Licensed) platform for threshold-secure operations
  - Tolerates **(n/3)** – (**n/2**) Byzantine faults
  - Operates over *eventually synchronous* networks
  - Self-heals and self-secures after faults and breaches
- Supported operations:
  - (O)PRF, ECIES, BLS, RSA (Blind)Sign/Decrypt
  - Generate, Store, Read, Delete, Enable, Disable
- Future enhancement goals:
  - Share conversion, Schnorr signatures, ECDSA
  - Share multiplication, MPC, Threshold AES
  - RSA (DKG/Refresh/Recover)
  - Post-Quantum Cryptography

# System Model

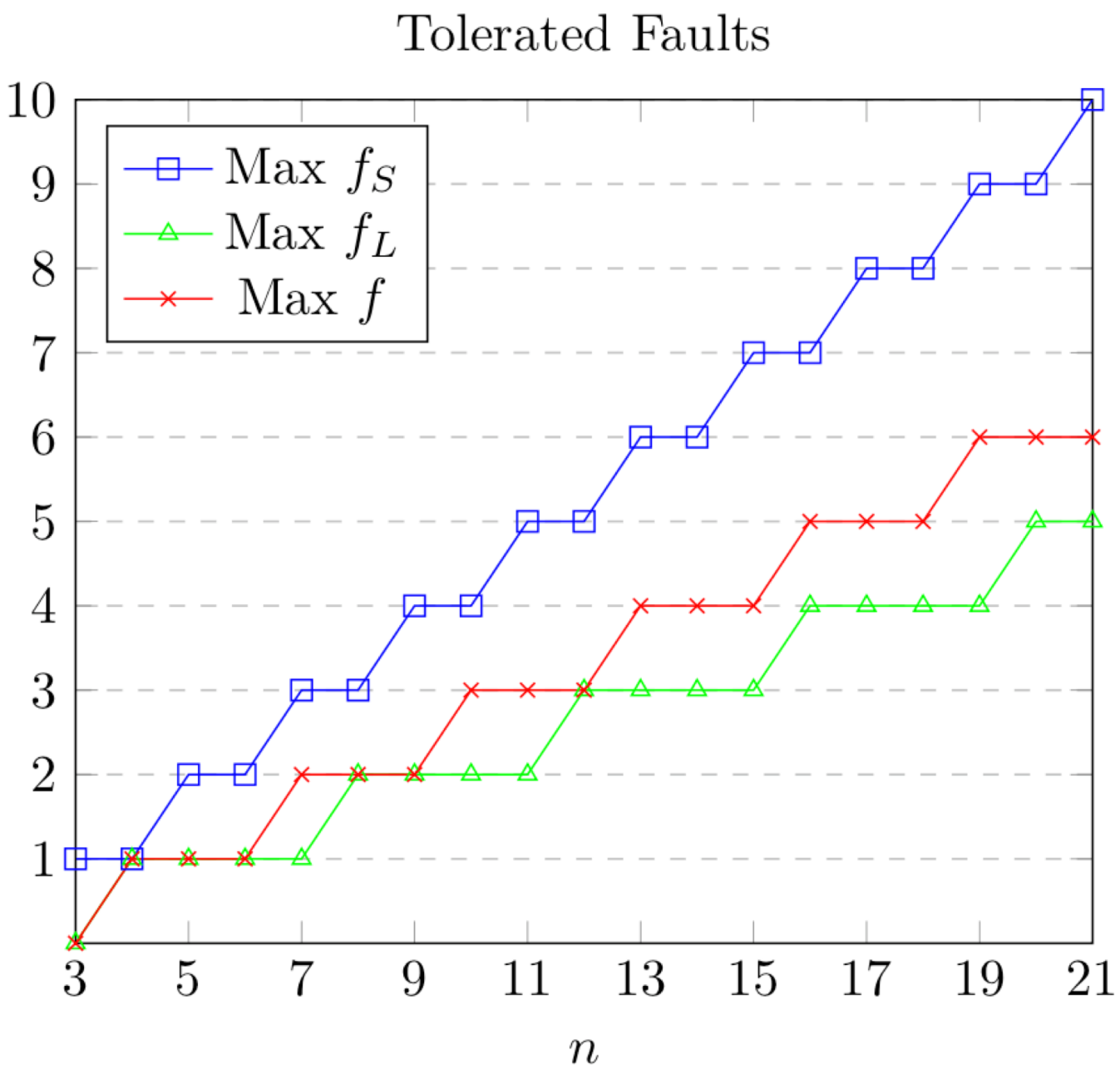
$$f = 1/3$$

$$f_S = n - 2 f_L - 1$$

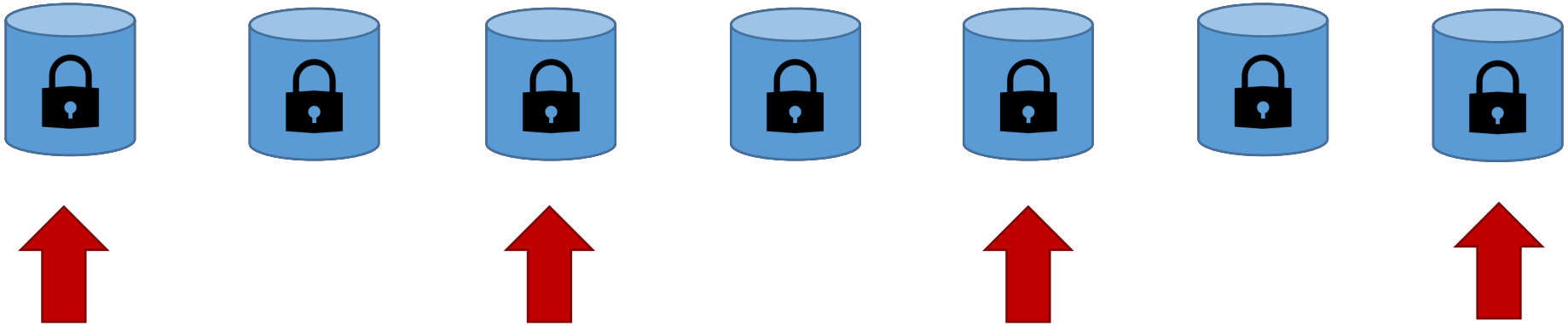
$$f_L = (n - f_S - 1) / 2$$



# Tunable Security

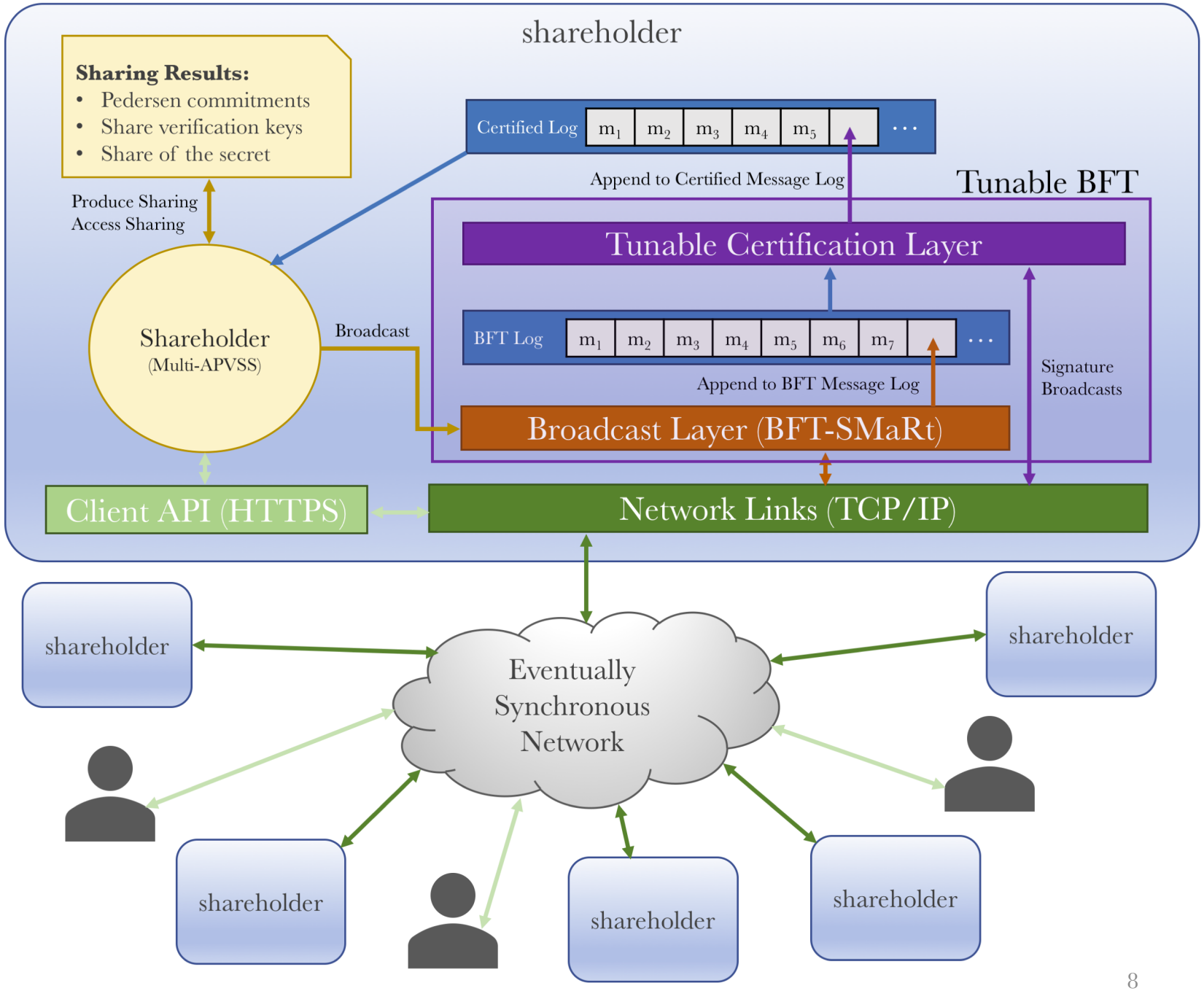


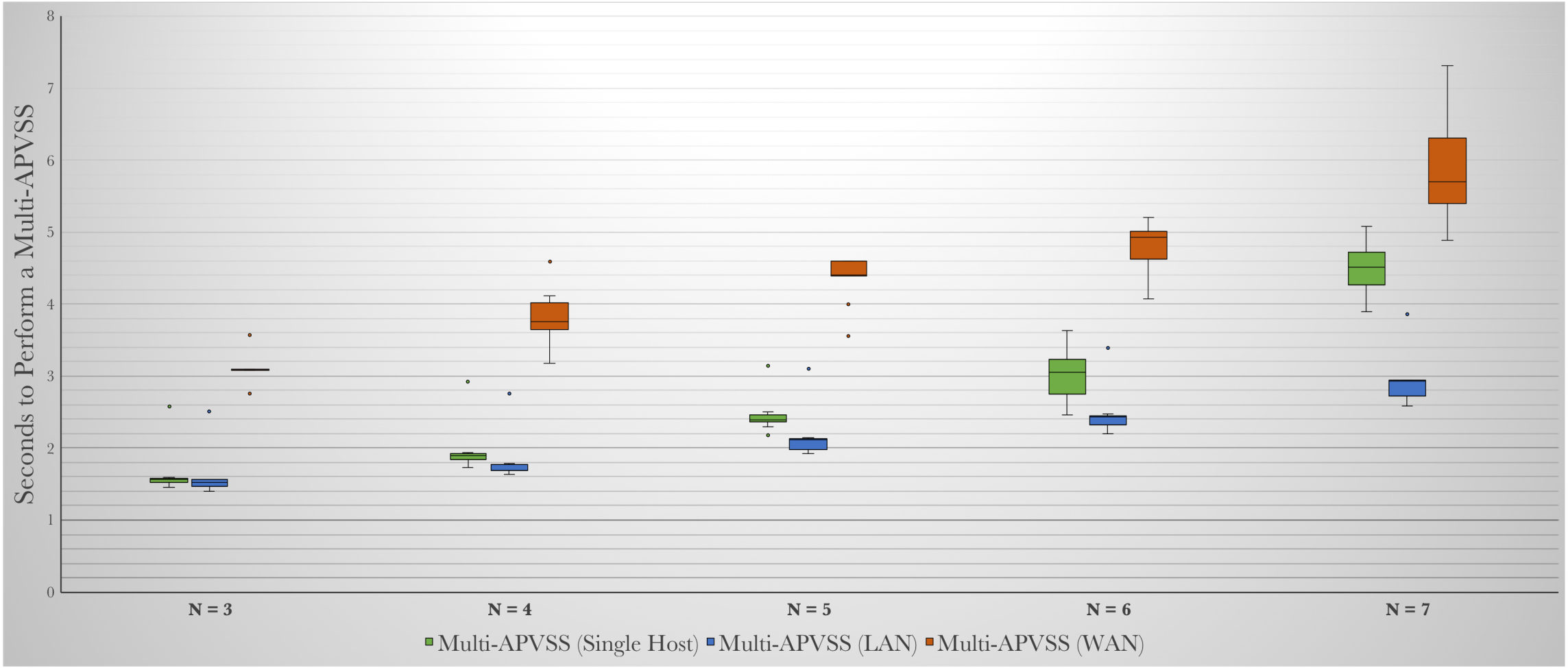
| | Safety | Liveness |
|---|---|---|
| Conventional Setting ($\mathbf{f} = 2$) | ✗ | ✗ |
| Tunable Security ($\mathbf{f_L} = 1$, $\mathbf{f_S} = 3$) | ✗ | ✗⚠ |



# Tunability in Practice

# System Architecture

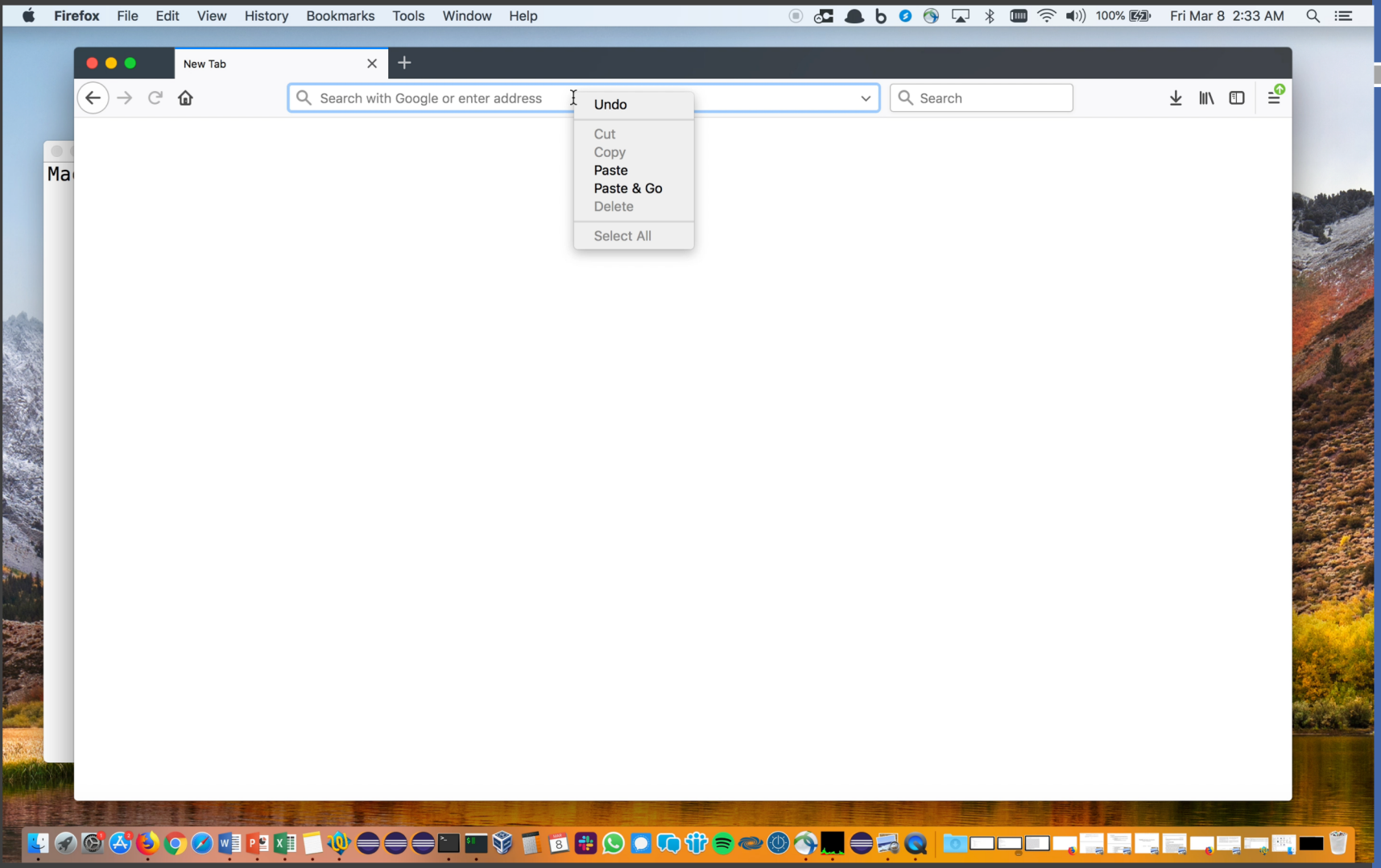





# Performance Results

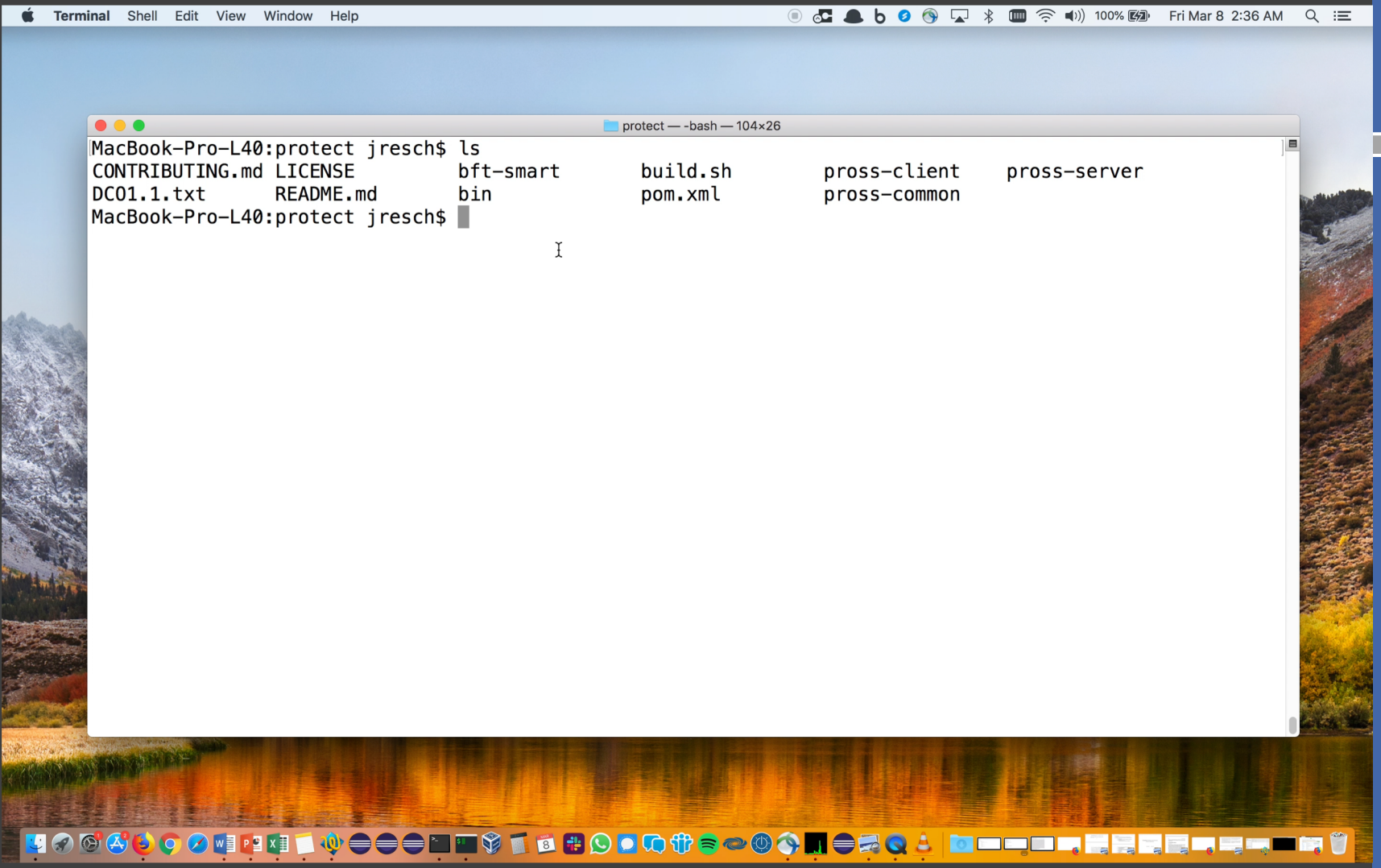
# Demo

1. Downloading
2. Installing
3. Configuration
4. Launching Servers
5. Client Setup
6. Secret Management:
   a. Distributed Generation
   b. Proactive Refresh
   c. Share Recovery
7. Cryptographic Operations:
   a. Decryption
   b. Signing



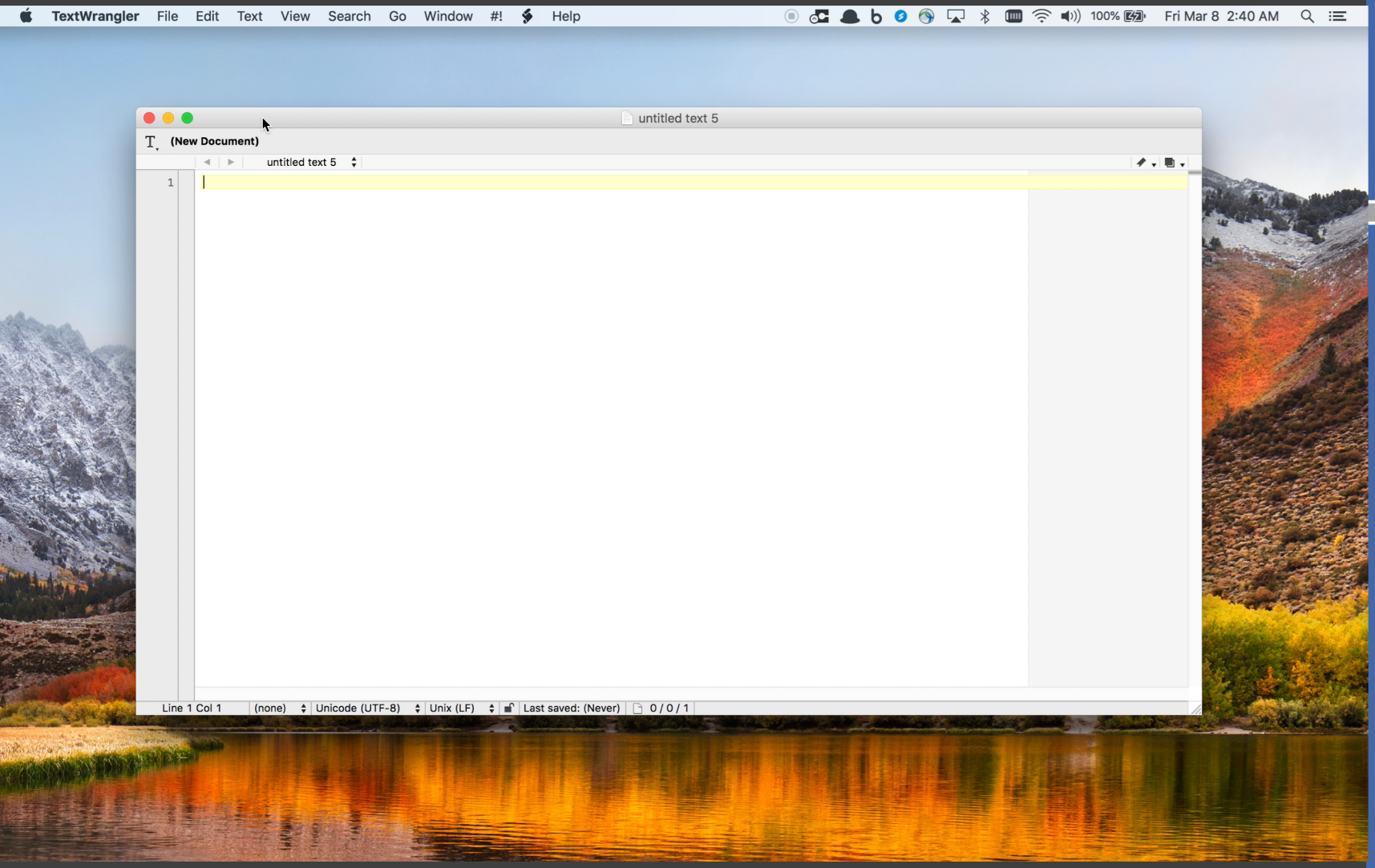

https://youtu.be/9sDgPOUpADw

1. Downloading
2. Installing
3. Configuration
4. Launching Servers
5. Client Setup
6. Secret Management:
   a. Distributed Generation
   b. Proactive Refresh
   c. Share Recovery
7. Cryptographic Operations:
   a. Decryption
   b. Signing



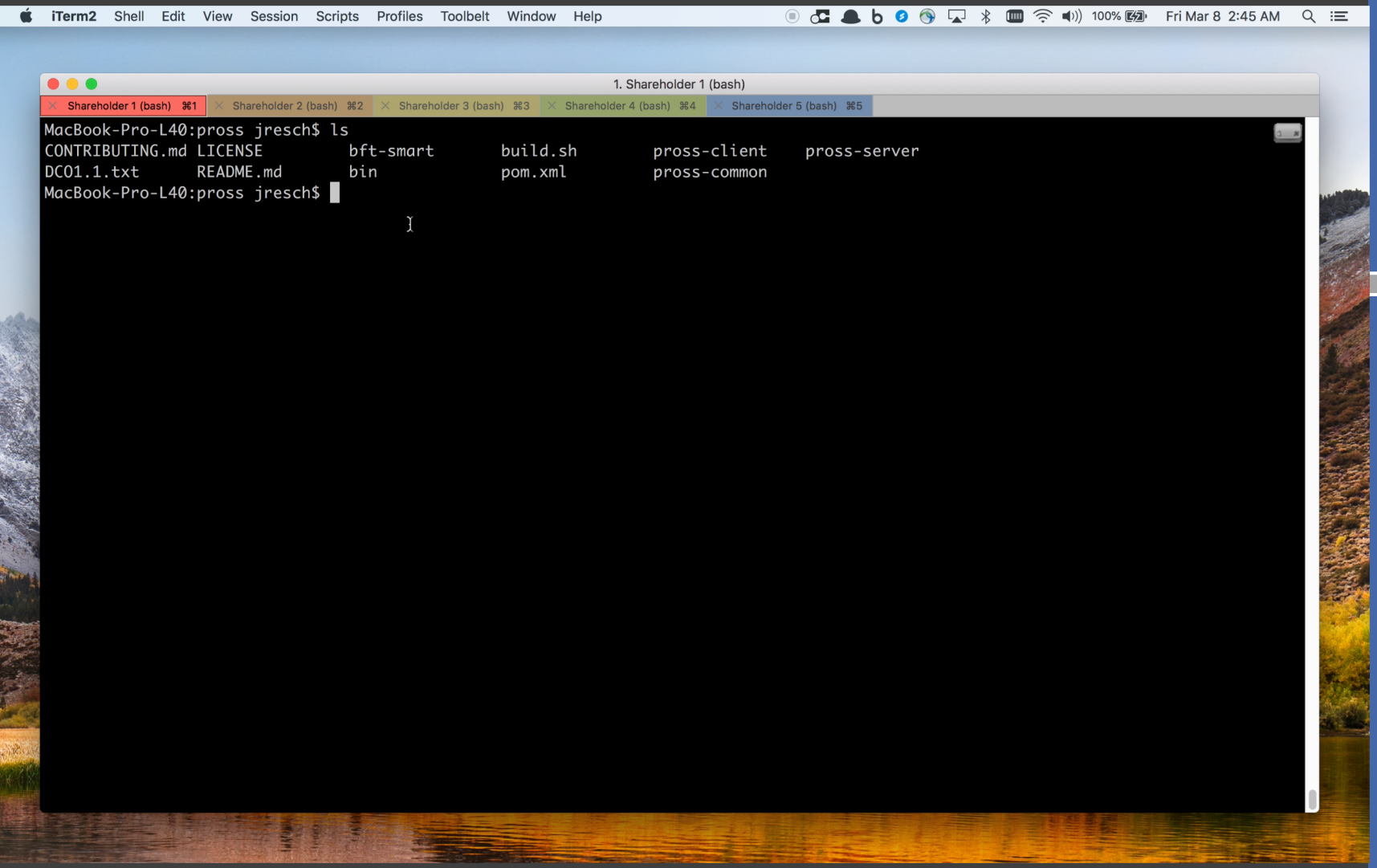

https://youtu.be/Cz9VV0FzW10

1. Downloading
2. Installing
3. Configuration
4. Launching Servers
5. Client Setup
6. Secret Management:
   a. Distributed Generation
   b. Proactive Refresh
   c. Share Recovery
7. Cryptographic Operations:
   a. Decryption
   b. Signing



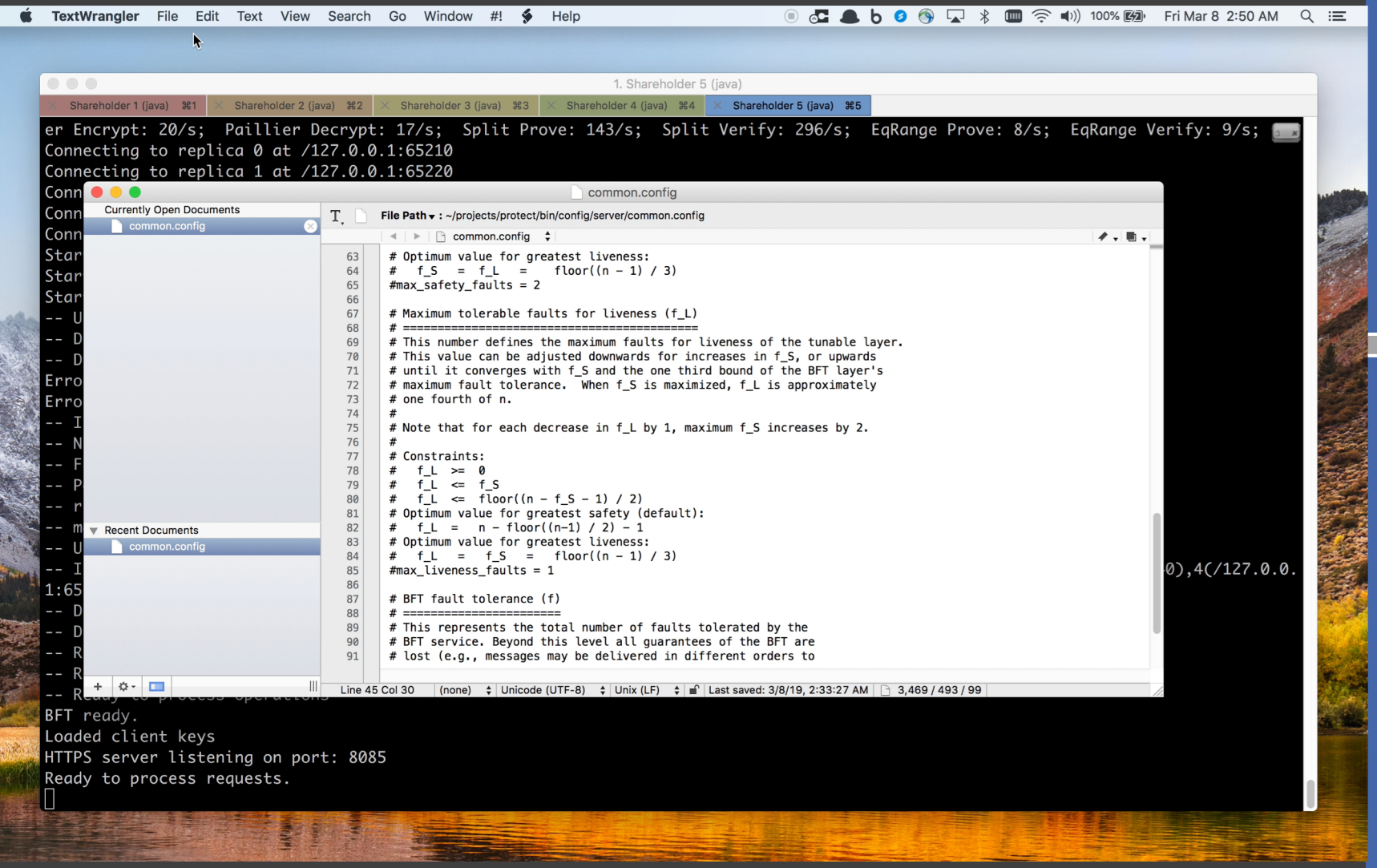

1. Downloading
2. Installing
3. Configuration
4. Launching Servers
5. Client Setup
6. Secret Management:
   a. Distributed Generation
   b. Proactive Refresh
   c. Share Recovery
7. Cryptographic Operations:
   a. Decryption
   b. Signing

https://youtu.be/BHM17XE6ZhQ



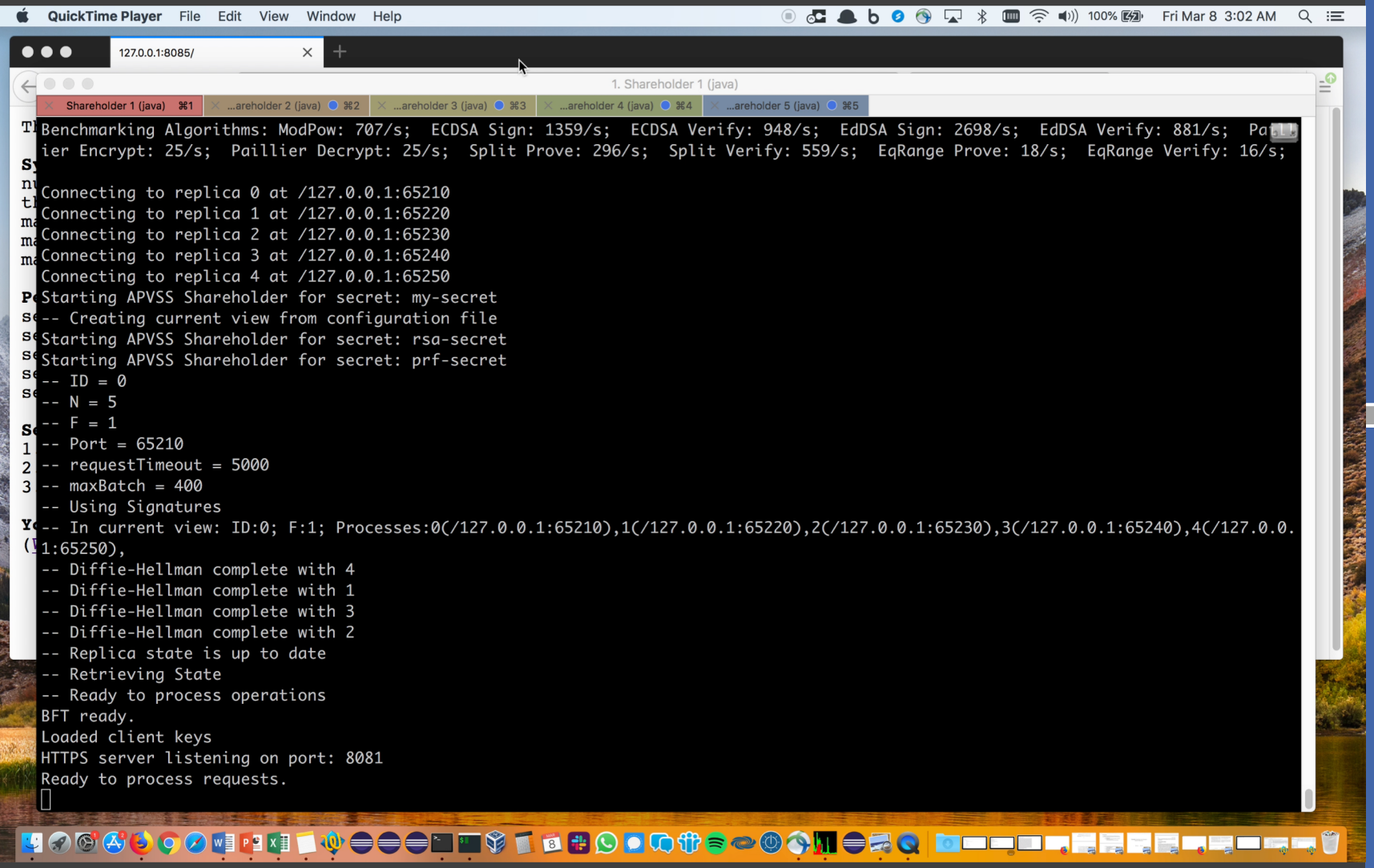

https://youtu.be/H4rX8gtqjrI

1. Downloading
2. Installing
3. Configuration
4. Launching Servers
5. Client Setup
6. Secret Management:
   a. Distributed Generation
   b. Proactive Refresh
   c. Share Recovery
7. Cryptographic Operations:
   a. Decryption
   b. Signing



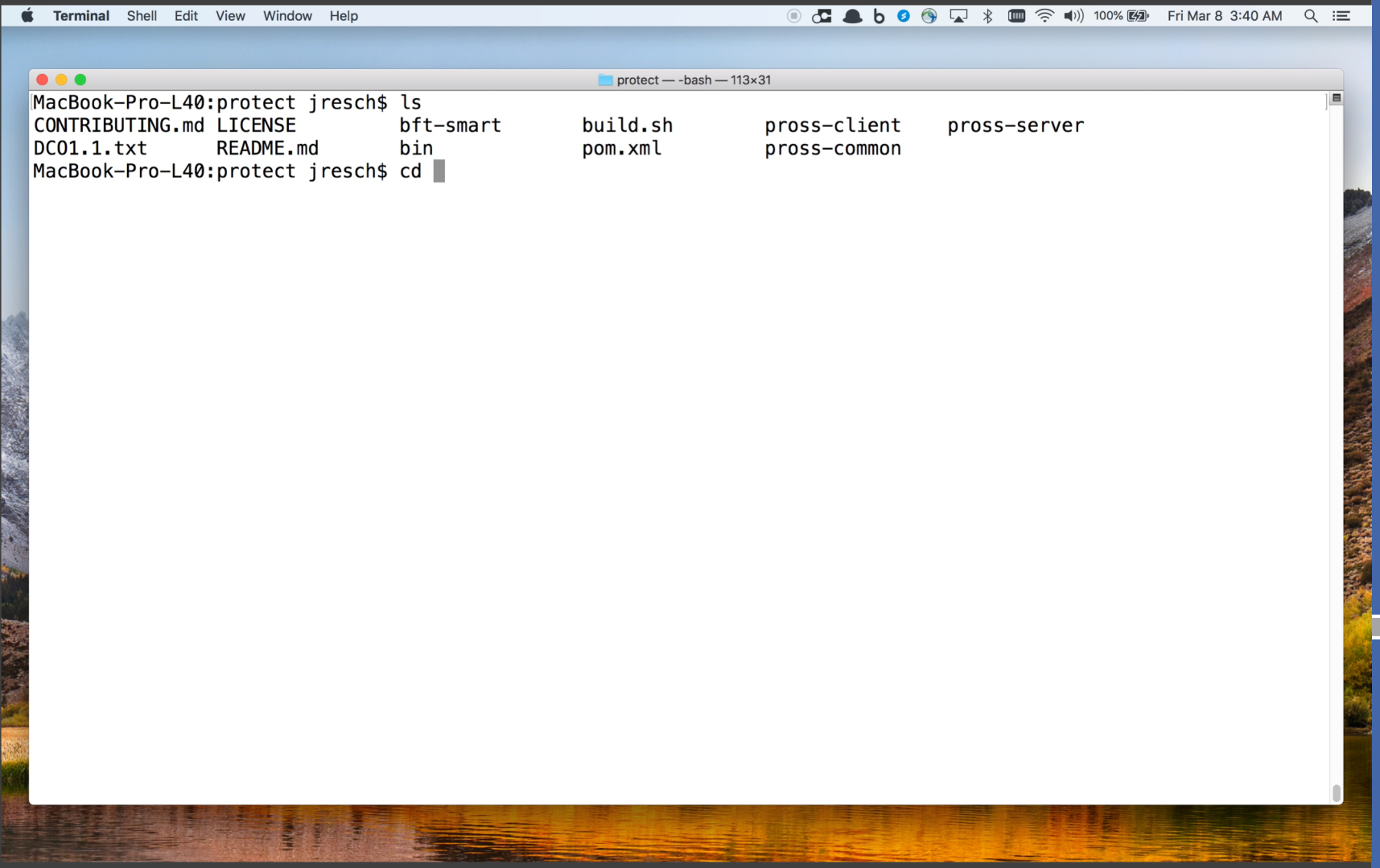

https://youtu.be/DXvrh1b8GH4

1. Downloading
2. Installing
3. Configuration
4. Launching Servers
5. Client Setup
6. Secret Management:
   a. Distributed Generation
   b. Proactive Refresh
   c. Share Recovery
7. Cryptographic Operations:
   a. Decryption
   b. Signing




https://youtu.be/ZMjMlC52MJc

1. Downloading
2. Installing
3. Configuration
4. Launching Servers
5. Client Setup
6. Secret Management:
   a. Distributed Generation
   b. Proactive Refresh
   c. Share Recovery
7. Cryptographic Operations:
   a. Decryption
   b. Signing




https://youtu.be/hVjxZmUPwlU

1. Downloading
2. Installing
3. Configuration
4. Launching Servers
5. Client Setup
6. Secret Management:
   a. Distributed Generation
   b. Proactive Refresh
   c. Share Recovery
7. Cryptographic Operations:
   a. Decryption
   b. Signing

# Q&A + Discussion

- Further Reading:
  - The "PROTECT" open source project:
    - http://bit.ly/protect123 (Github Repository)
    - New contributors, testers, implementers welcome!
  - Our eprint "Tunable Protocols for Threshold and Proactive Cryptography"
    - Provides details of the underlying protocols
    - Coming soon!