It's been two years since Network Extension, Part 1 - Introduction. You've been asking for Part 2. Here it finally is.

- Let's Build a VPN Protocol (You are here)
- Prologue: How Does VPN Work?
- VPN, Part 1: VPN Profiles
- VPN, Part 2: Packet Tunnel Provider

VPN

What is VPN? Originally, a VPN (Virtual Private Network) was designed to establish secure connections between a computer and a protected or private corporate network, using an otherwise insecure public network (such as the Internet). In recent years, VPNs also grew in popularity in the consumer market as a way to stay secure and anonymous when going online, and also enjoying unrestricted network access by hiding the source of traffic.

If that sounds a bit too abstract, then you are in the right place. In this series, I'm going to cover most of the bits and pieces that make VPN work. And what is a better way to do that than implementing a custom VPN protocol?

⚠️ **Requirements**.

To run the sample code in this series, you are going to need a physical device and an Apple Developer Account to get proper entitlements.

Network Extension Framework

You are going to learn how the iOS manages VPN configuration, how to create custom VPN configurations programmatically, how to implement both the client and the server sides of the VPN.

The primary topic of this series is going to be Network Extension framework. Among many of its features, it has extensive support for virtual private networks (VPN). The client side of the custom protocol implementation is implemented as a Packet Tunnel Provider extension. The Packet Tunnel Provider extension's containing app uses `NETunnelProviderManager` to create and manage VPN configurations that use the custom protocol, and to control the VPN connections specified by the configurations.

> The entire app source code is available at kean/VPN.

Continue Reading »