Wireguard: A Modern VPN Protocol

Jinank Jain, Rayhaan Jaufeerally

July 17, 2018

ETH Zürich

Introduction

Why VPN?

- Necessary for point to point security between campuses (e.g. DC's, corporate offices, . . .)
  - As early as the 1970's governments have tapped undersea cables for intelligence,
  - Operation Ivy Bells in 1971 tapped Russian communicatioons to millitary bases[1],
- Necessary for end users to get a clean connection:
  - ISP's doing DNS hijacking to serve inappropriate content,
  - Open WiFi networks when travelling,
  - Geoblocking,

[1] https://en.wikipedia.org/wiki/Operation_Ivy_Bells

Necessity in real life

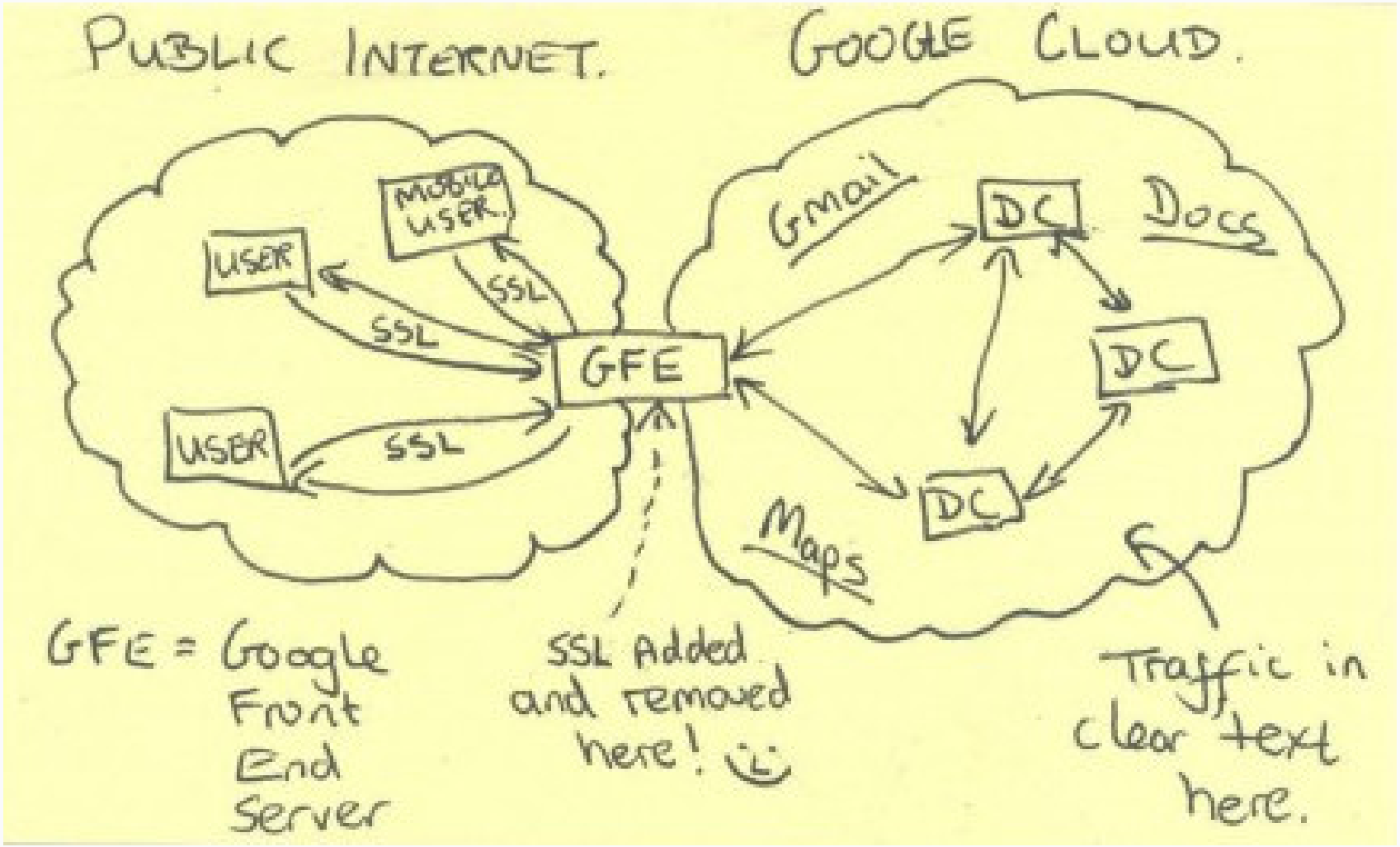


Figure 1: Government attempts to intercept user's content

History of VPN protocols

- IPSEC
  - Popular for site to site connections with dedicated router hardware,
  - Tedious to set up and high degree of complexity,
  - Large attack surface between IKE (v2), SA mechanisms, XFRM in Linux,
  - Legacy protocol support,
  - IP in IP,
- OpenVPN
  - Implemented in userspace with TUN/TAP (slow),
  - Complex confoguration vulnerable to leaks,
  - Stateful protocol which is brittle in real networks,
  - Large codebase / attack surface,

What is Wireguard?

- Opinionated Layer 3 secure network tunnel for IPv4 and IPv6.
- Lives in the Linux kernel, but cross platform userspace implementations are available.
- UDP based. Punches through firewall.
- Modern conservative cryptographic principles.
- Emphasis on simplicity and auditability.
- Authentication model similar to SSH's `authorized_keys`.

Easily Auditable

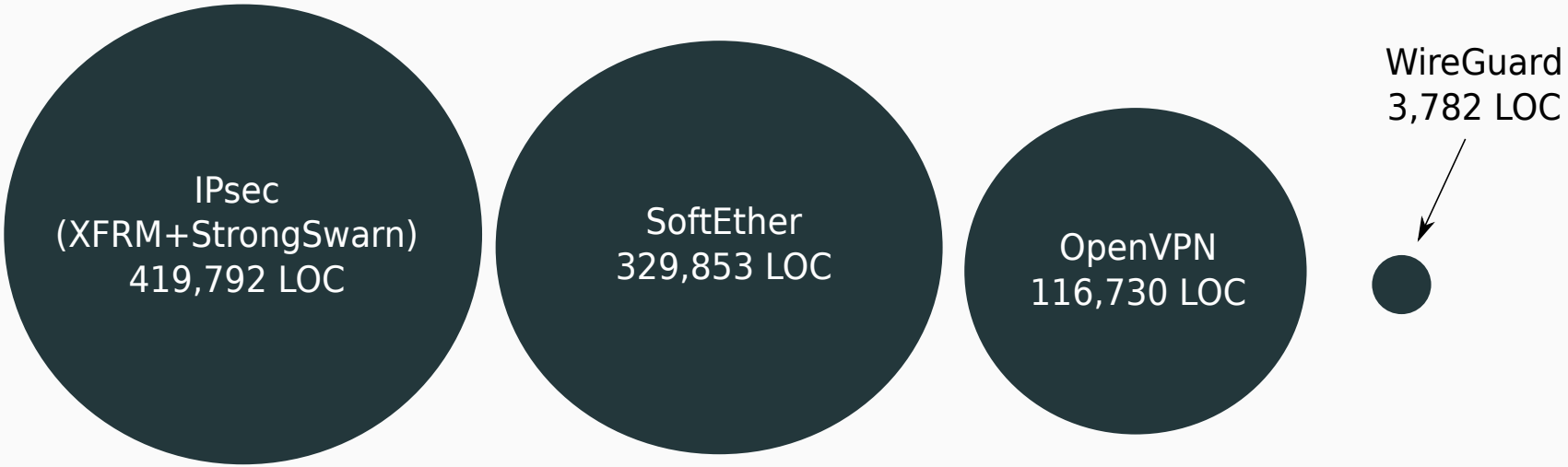


Figure 2: Comparing different VPN protocols in terms of LOC

Minimialistic Interface

"Developers should write programs that can communicate easily with other programs"

— Unix Philosophy

- Wireguard presents a normal network interface

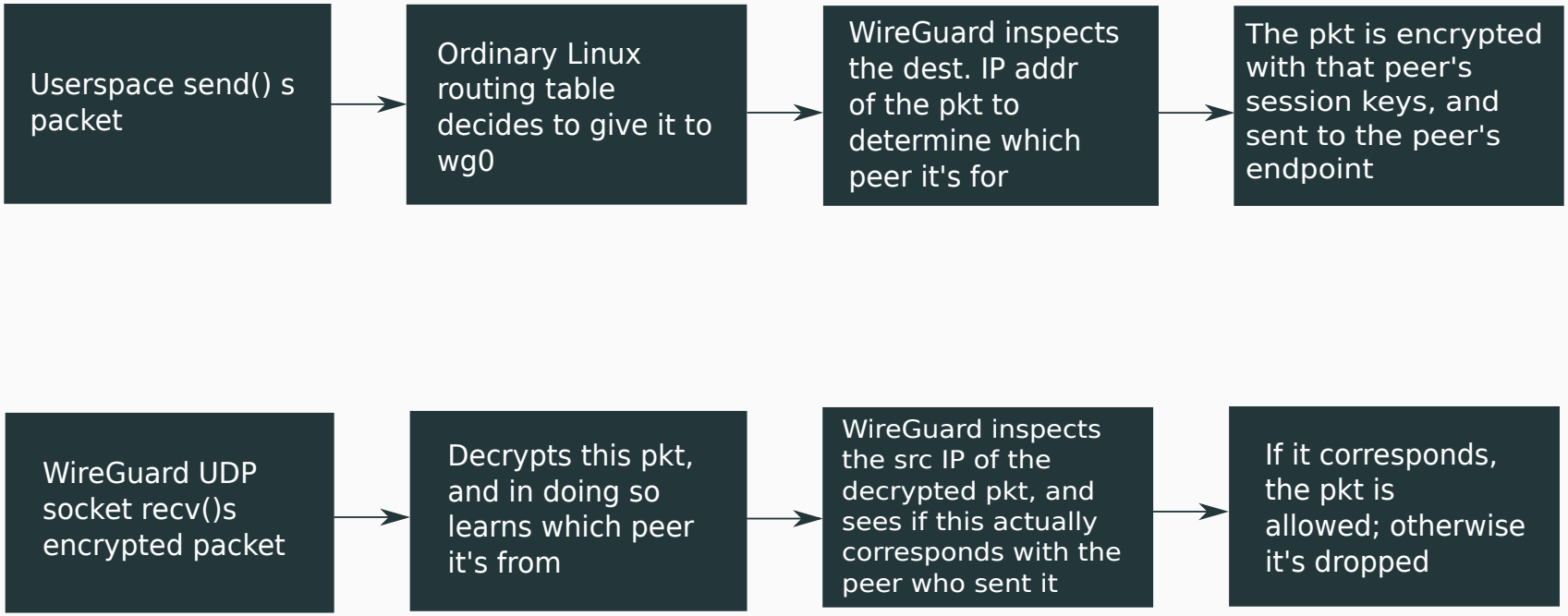
```
# ip link add wg0 type wireguard
# ip address add 10.0.32.1/24 dev wg0
# ip route add default via wg0
```

- By using a standard interface it becomes easier to administer using the existing iproute2 utilities for example

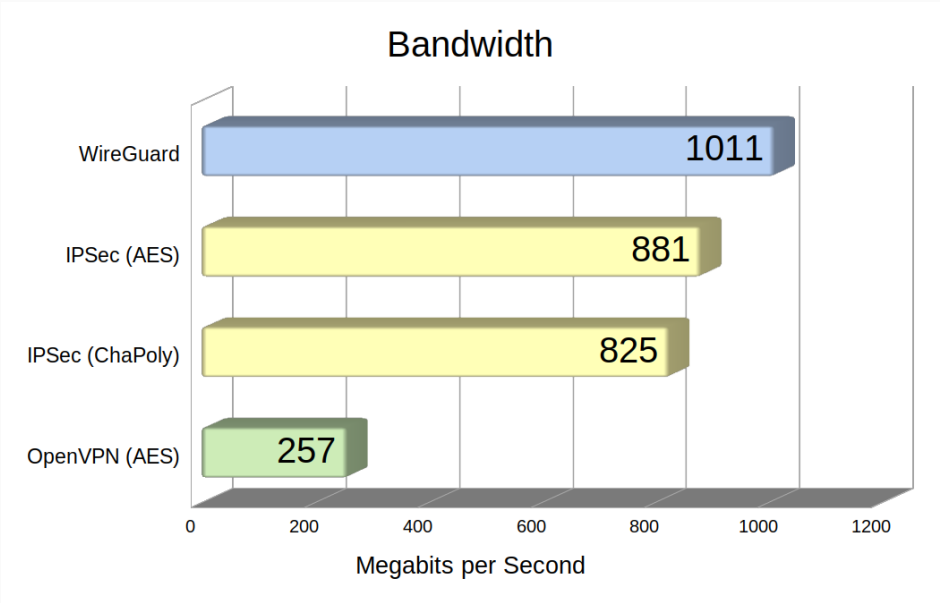
Cryptokey Routing

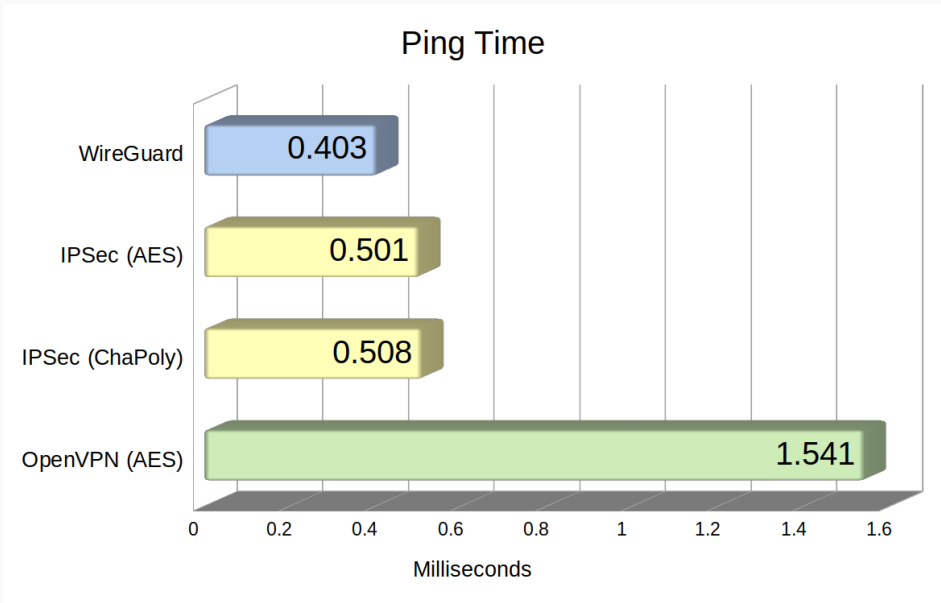
- Fundamental concept of any VPN service
  - Create **mapping** between **public keys of peers** and their **IPs**.
- WireGuard interface has:
  - A private key
  - A listening UDP port
  - A list of peers
- Peer has
  - A public key
  - A list of associated tunnel IPs
  - Optionally has an endpoint IP and port

Cryptokey Routing



Performance

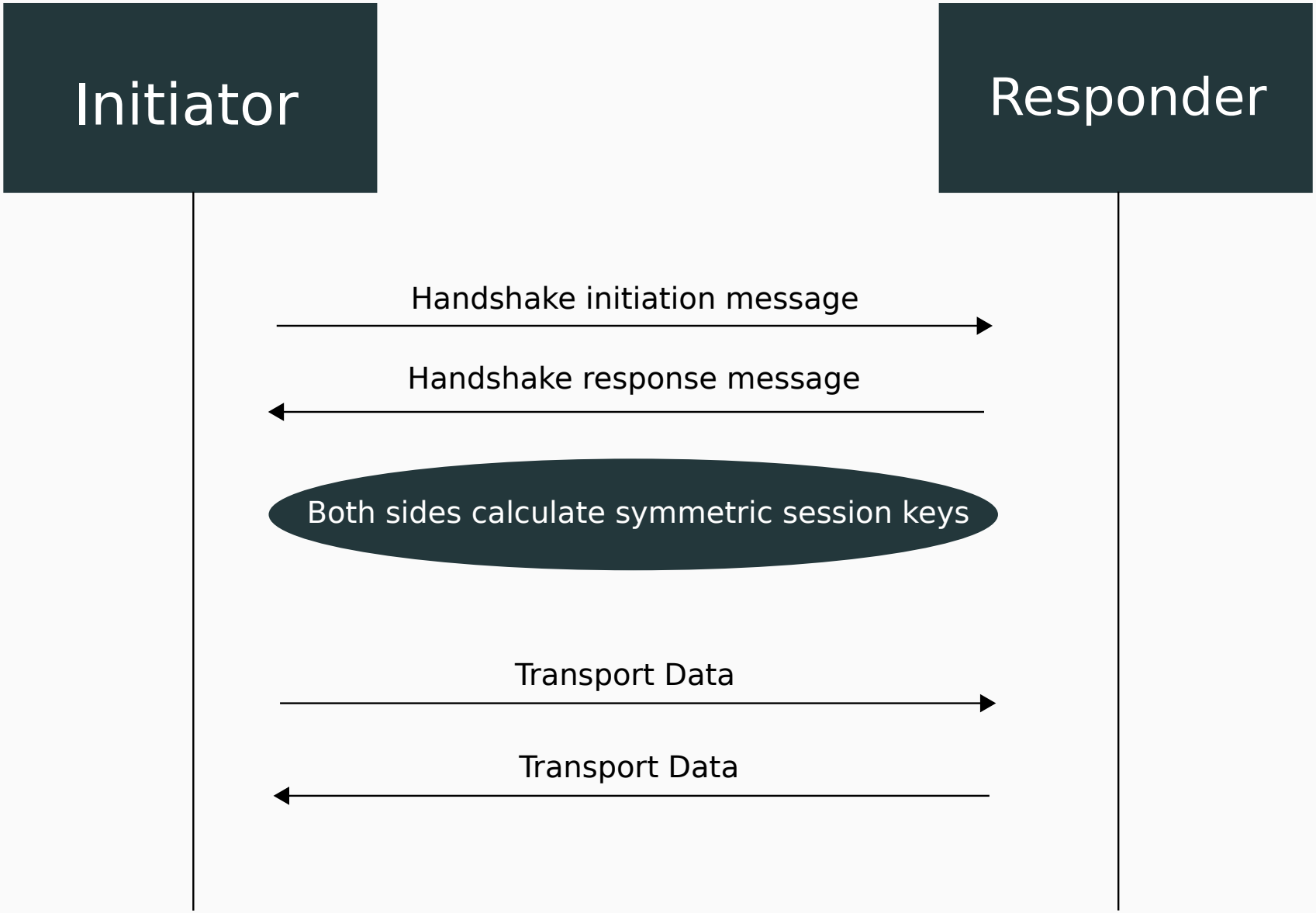




Stealth

- Behaves like a rootkit in some sense.
- Should not respond to any unauthenticated packets.
- Hinder scanners and service discovery.
- Service only responds to packets with correct crypto.
- Not chatty at all.

The Key Exchange



Session key derivation - NoiseIK

- One peer is the initiator; the other is responder
- Each peer has their static identity - their long term *static keypair*
- For each new handshake, each peer generates an *ephemeral keypair*