Wireguard: A Modern VPN Protocol

Jinank Jain, Rayhaan Jaufeerally

July 15, 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

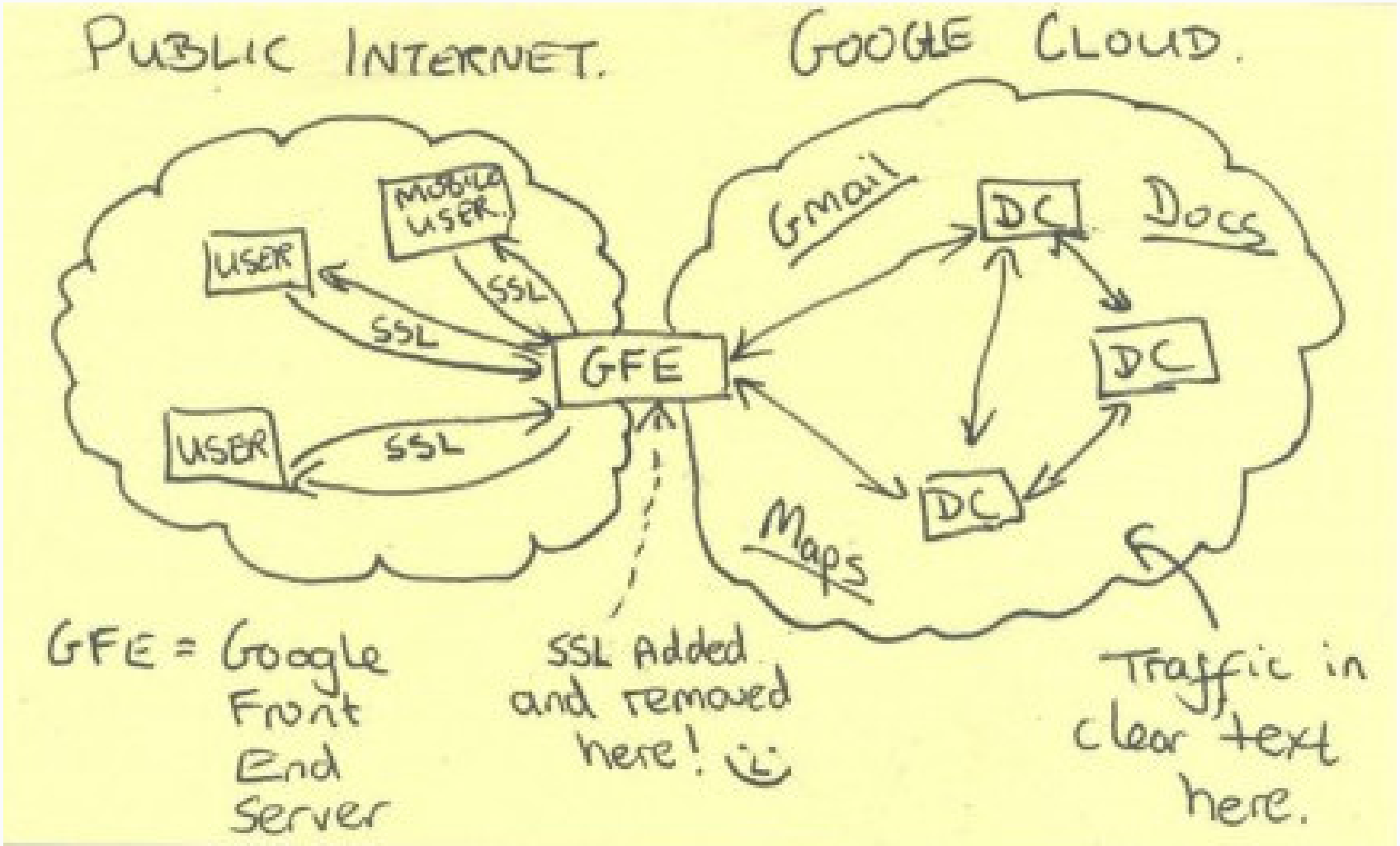

Figure 1: Government attempts to intercept user's content

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,

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