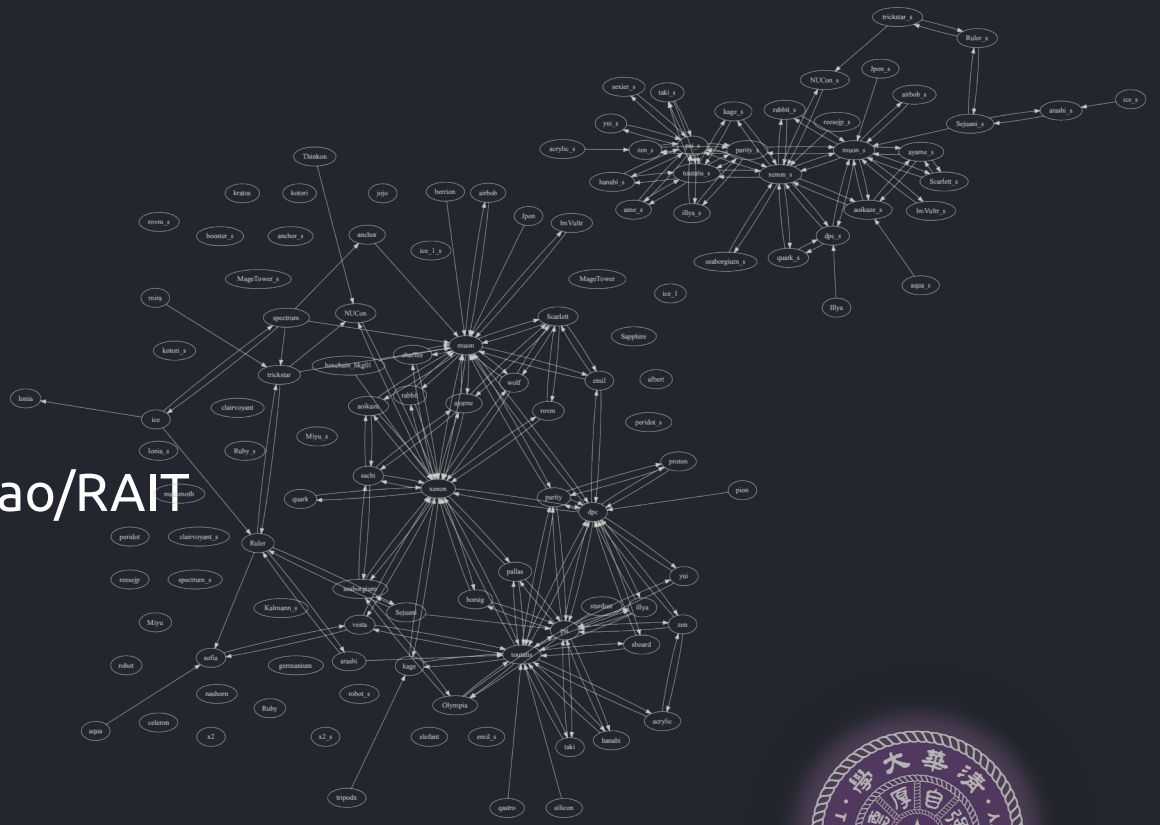
Full mesh overlay network with **IPsec**: myths, experiments and beyond.

Ao Shen, Yipeng Liu, Yuchen Wei, Yi Xu

Background

- Entropy
  - Full mesh overlay network with **TINC** tunnels

- Mesh 拓扑结构：不依赖中心节点持续在线
- 从注册表自动生成隧道配置：gitlab.com/NickCao/RAIT
- 路由协议：babel，基于 RTT 选择路径
- 承载 NFS 等科学计算日常负载



Background

- Gravity
  - Full mesh overlay network with **WireGuard** tunnels

- Mesh 拓扑结构：不依赖中心节点持续在线
- 从注册表自动生成隧道配置：gitlab.com/NickCao/RAIT
- 路由协议：babel，基于 RTT 选择路径
- 承载 NFS 等科学计算日常负载

Background

- WireGuard 建立隧道依赖 UDP 开放端口
- 使用点对点隧道完成 VPN server

- 后果
  - 加入新节点需要所有节点新开放 UDP 端口
  - 大量端口不利于网络管理

Prior Work

- Full-Mesh IPsec Network: 10 Dos and 500 Don'ts
- Fran Garcia, *Hosted Graphite*, USENIX SECCon '16

- WORST MIGIRATION EVER?
- 面临的问题类似，提供了一些部署建议
- 使用的 racoon 较为陈旧
  - 维护困难
  - 讲者的第一条 Don't: Don't use ipsec-tools/racoon! (like we did)

TL;DW

“IPsec is awful”

(trust me on this one)

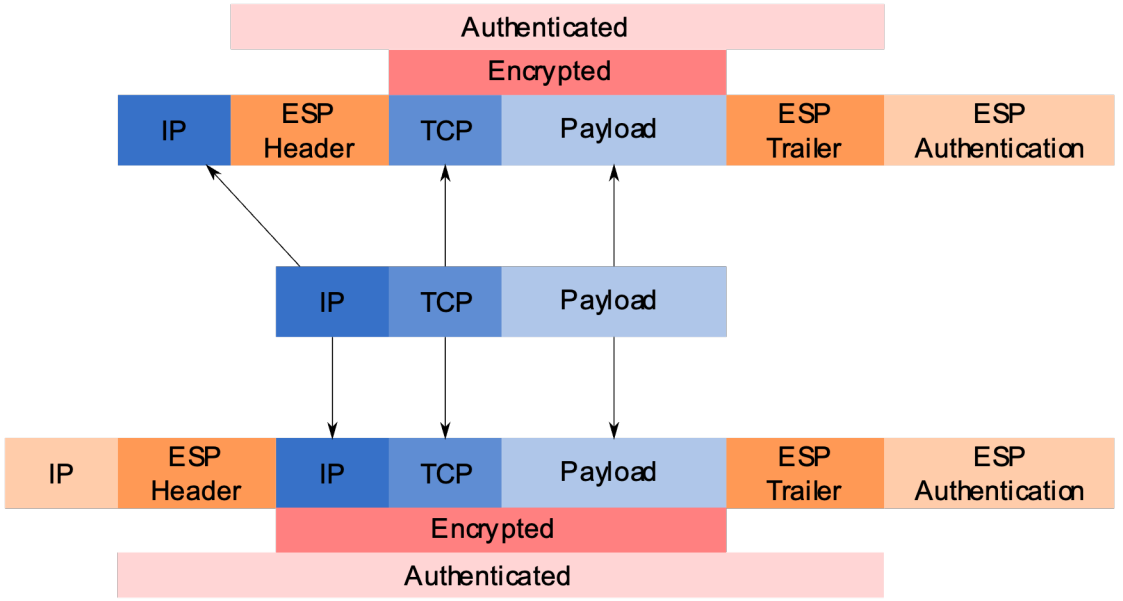
IPsec as RFCs

- 比起“协议”，更像是“协议套装”
- Wikipedia 提到了 60+ RFCs
- StrongSwan 实现文档中提到了接近 100 RFCs

- Wireguard：一组通用的密码学参数（Chacha20 etc）
- IPsec：各种各样可以修改的参数

IPsec as RFCs

- 对任意三层（IP 协议）通信的认证和加密协议
- Authenticated Header (AH, RFC 4302) 认证而不加密
  - 听起来似乎被 ESP 严格覆盖，但是
- Encapsulating Security Payload (ESP, RFC 4303) 加密 payload
  - Tunnel = VPN
    - 加入额外 IP header
    - 可以连接两个子网
  - Transparent = P2P
    - 仅加密 payload
    - 使用原来的 header
    - 对 MTU 影响较小



IPSec and IKE

AH vs. ESP

International Engineering Task Force meeting held just before AH and ESP were finalized:

Microsoft rep: "AH is useless given the existence of ESP, cluttered up the spec, and couldn't be implemented efficiently because the HMAC is in front of the data it authenticates."

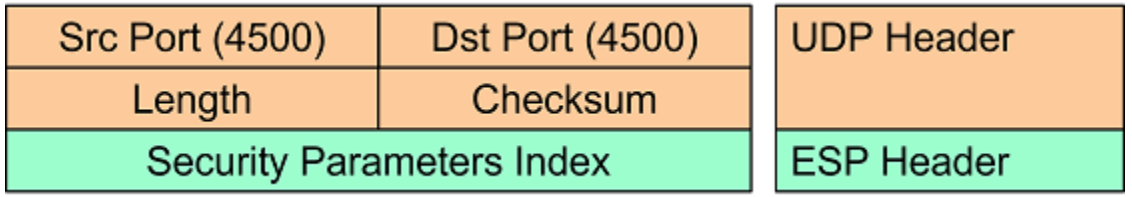
Then everyone in the room looked around at each other and thought:

Hmmm...he's right, and we hate AH, but if it annoys Microsoft let's leave it in since we hate Microsoft more than we hate AH

IPsec as RFCs

- 防火墙会如何处理这些包
  - 三层协议

- 在 IP 头部加入一层 UDP 头部传输 (RFC 3948)
  - 本意为解决 NAT 等 middlebox 问题 (RFC 3947)
  - 网络节点仅需处理一个统一端口，解决端口核心问题
- 也有 TCP 传输的方案 (RFC 8229)，支持尚不完善



IPsec in Linux Kernel

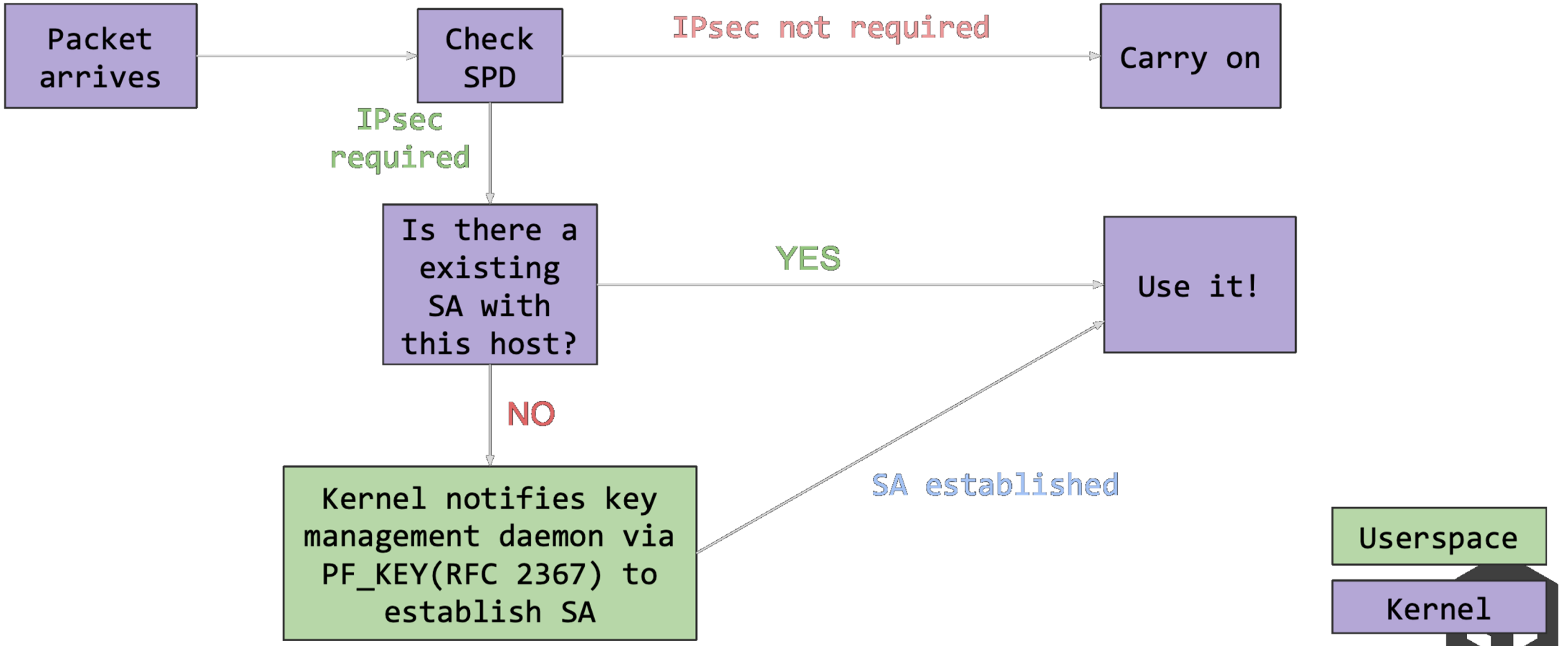
- Security Policy Database
  - *ip xrfm policy*
  - *src - dst / if_id*
  - 判断是否需要 IPsec

```
root@ipsec01:~# ip xfrm policy
src ::/0 dst 3000:4::2/128
        dir out priority 334463 ptype main
        tmpl src 172.28.196.168 dst 114.253.36.234
                proto esp spi 0xc84ea865 reqid 6 mode tunnel
src 3000:4::2/128 dst ::/0
        dir fwd priority 334463 ptype main
        tmpl src 114.253.36.234 dst 172.28.196.168
                proto esp reqid 6 mode tunnel
src 3000:4::2/128 dst ::/0
        dir in priority 334463 ptype main
        tmpl src 114.253.36.234 dst 172.28.196.168
                proto esp reqid 6 mode tunnel
src ::/0 dst ::/0
        dir out priority 399999 ptype main
        tmpl src 172.28.196.168 dst 101.201.70.121
                proto esp spi 0xcad14f73 reqid 3 mode tunnel
        if_id 0x3
```

- Security Association Database
  - ip xrfm state
  - 加密使用的密钥
  - 如何维护？
  - 理论上来说你可以 ip xfrm policy add

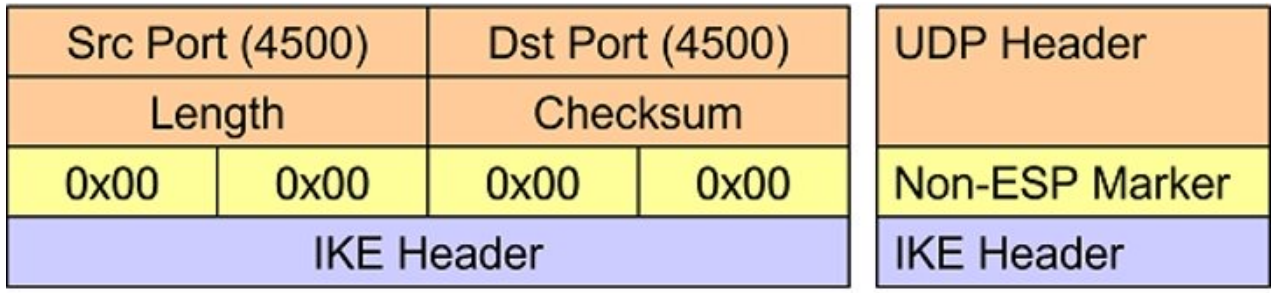
```
> sudo ip xfrm state
src 192.168.1.16 dst 47.92.28.59
        proto esp spi 0xce363a30 reqid 1 mode tunnel
        replay-window 0 flag af-unspec
        auth-trunc hmac(sha256) 0xced550b5d6f438b89c18f1ff8a5ecfa54a5b10b537c7c9caeb28398f4cf765c6 128
        enc cbc(aes) 0xf8bb1a3cc68cb48b3afc88200c1c8124
        encap type espinudp sport 4500 dport 5600 addr 0.0.0.0
        anti-replay context: seq 0x0, oseq 0x18, bitmap 0x00000000
        if_id 0x5
src 47.92.28.59 dst 192.168.1.16
        proto esp spi 0xc84ea865 reqid 1 mode tunnel
        replay-window 32 flag af-unspec
        auth-trunc hmac(sha256) 0x7088ae3329eba313360888815a855d83328ffdfc69c578a5900e1872c75c9b0a 128
        enc cbc(aes) 0x115dae1f7bd2fcc0464a6d6b2c6e3072
        encap type espinudp sport 5600 dport 4500 addr 0.0.0.0
        anti-replay context: seq 0x0, oseq 0x0, bitmap 0x00000000
        if_id 0x5
```

IPsec in Linux Kernel



IPsec – Internet Key Exchange

- Internet Key Exchange (IKE, RFC 4109(v1), RFC 7296(v2)) 协议
- 用户态 daemon，进行认证和密钥交换操作
  - StrongSwan 最常用也最灵活的实现
  - LibreSwan 是 StrongSwan 前身
  - Racoon 是 BSD 最早的 IKE daemon 实现
- IKE daemon 响应内核请求，负责往 *ip xfrm* 接口插入相应数据
- 网络上的教程（多数来源于 Cisco 等网络设备厂商）：需要使用证书进行认证
  - CA 的管理和签发复杂
  - 适应Windows等客户端的实现
- 直接使用 pubkey 认证
  - 已有中心化的节点列表（git）



IPsec – Internet Key Exchange

- Do you really need all this?

```
$ ipsec pki --gen --type rsa --size 2048 --outform pem > private/ClientKey.pem
$ chmod 600 private/ClientKey.pem
$ ipsec pki --pub --in private/ClientKey.pem --type rsa | \
      ipsec pki --issue --lifetime 730 --outform pem \
                --cacert cacerts/strongswanCert.pem \
                --cakey private/strongswanKey.pem \
                --dn "C=CH, O=strongSwan, CN=myself@example.com" \
                --san myself@example.com \
           > certs/ClientCert.pem
```

Fortunately, you don't

IPsec – Internet Key Exchange

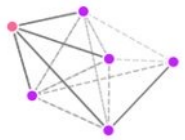
openssl genpkey -algorithm ed25519 -outform PEM

openssl pkey -pubout

- Directly code public key is fine!

IPsec - testbed

Routers



Configuration

Current node: ipsec-pek-alfa

Show uninstalled routes

1:1 Zoom in Zoom out

State: Connected.

Legend Current Neighbours Others

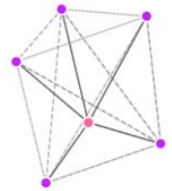
Neighbours

address	if	reach	rxcost	txcost	cost	rtt
fe80::9ba8:f7e1:9a92:ef33	xfirm-szx	ffff	96	96	96	
fe80::2e58:c939:8f60:b9c8	xfirm-zjk1	001f	65535	96	65535	
fe80::b84f:76e9:13af:14da	xfirm-pek	ffff	96	96	96	
fe80::13e7:72fb:dc0b:ae11	xfirm-hgh	ffff	96	96	96	
fe80::642d:3751:a09b:5245	xfirm-sha	ffff	96	96	96	
fe80::2701:e9a4:348c:503e	xfirm-zjk2	ffff	96	96	96	

Legend Wired link (96) Lossless wireless link (256) Unreachable (65535)

(based on *rxcost*; colors are interpolated logarithmically)

Routers



Configuration

Current node: ipsec-pek-alfa

Show uninstalled routes

1:1 Zoom in Zoom out

State: Connected.

Legend Current Neighbours Others

Neighbours

address	if	reach	rxcost	txcost	cost	rtt
fe80::9ba8:f7e1:9a92:ef33	xfirm-szx	ffff	96	96	127	45.762
fe80::642d:3751:a09b:5245	xfirm-sha	ffff	96	96	118	35.798
fe80::13e7:72fb:dc0b:ae11	xfirm-hgh	ffff	96	96	112	28.827
fe80::2701:e9a4:348c:503e	xfirm-zjk2	ffff	96	96	96	8.829
fe80::6fd4:5d7e:e8ac:943	xfirm-zjk1	ffff	96	96	96	8.822
fe80::b84f:76e9:13af:14da	xfirm-pek	ffff	96	96	96	7.164

Legend Wired link (96) Lossless wireless link (256) Unreachable (65535)

(based on *rxcost*; colors are interpolated logarithmically)

Routes