black hat®
USA 2024

AUGUST 7-8, 2024
BRIEFINGS

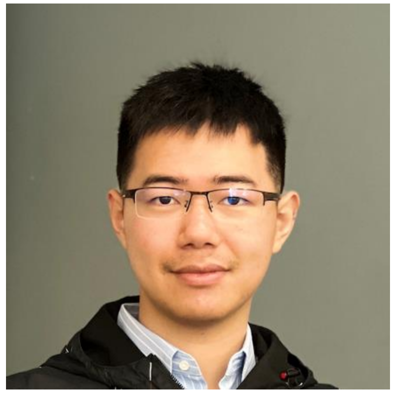
Fallen Tower of Babel: Rooting Wireless Mesh Networks by Abusing Heterogeneous Control Protocols

Speakers: Xin’an Zhou and Zhiyun Qian

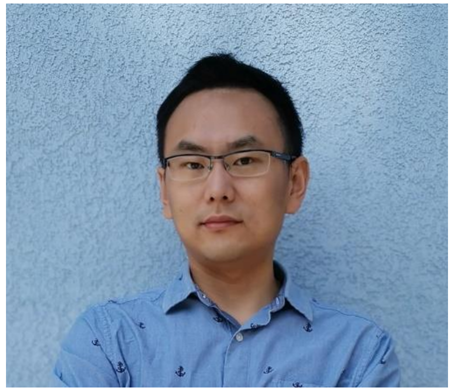
Contributors: Juefei Pu, Qing Deng, Srikanth Krishnamurthy, Keyu Man

8/7/2024

Team/Contributors at UC RIVERSIDE

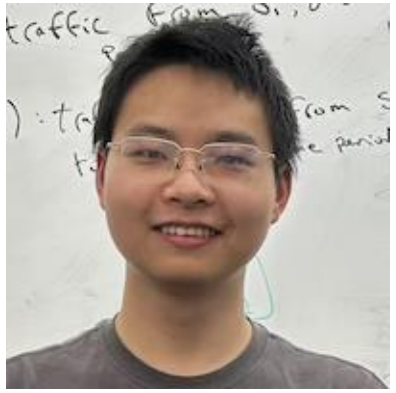


Xin’an Zhou



Zhiyun Qian

Qing Deng



Juefei Pu

Keyu Man

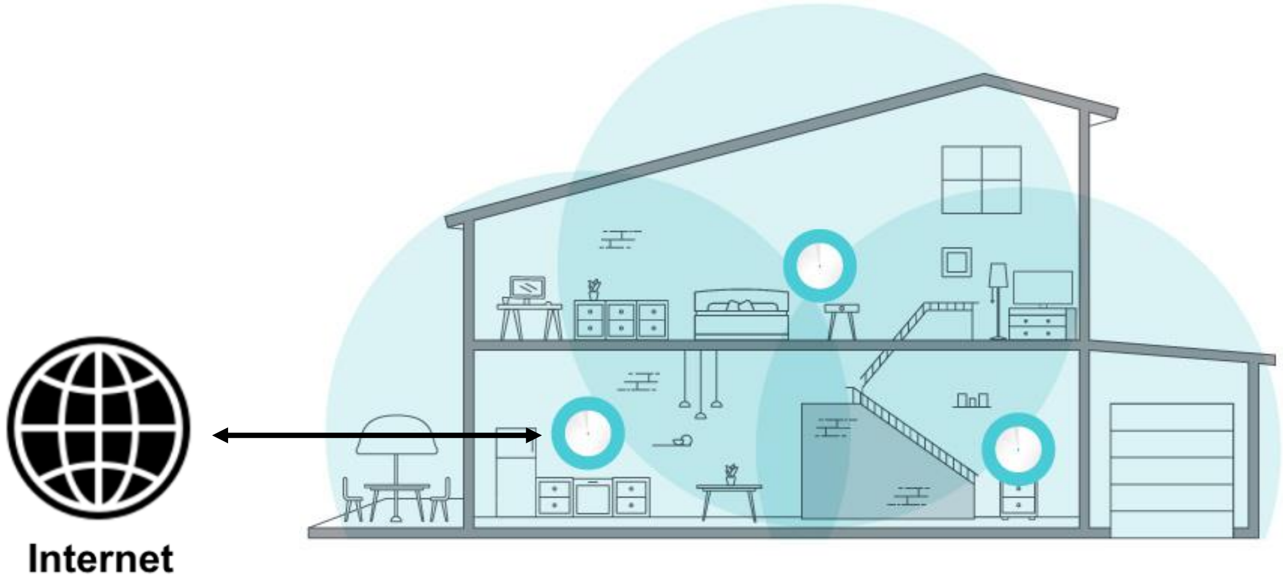
Srikanth Krishnamurthy

Agenda

- Background on home wireless mesh networks
- Two types of security flaws
- Exploitation
- Defenses

Background: Home Wireless Mesh Networks

1. An emerging type of Wi-Fi network. Wi Fi ALLIANCE
2. Single gateway node + multiple extender nodes



Images: TP-Link

Wireless Mesh Networks are increasingly popular!

Netgear Orbi

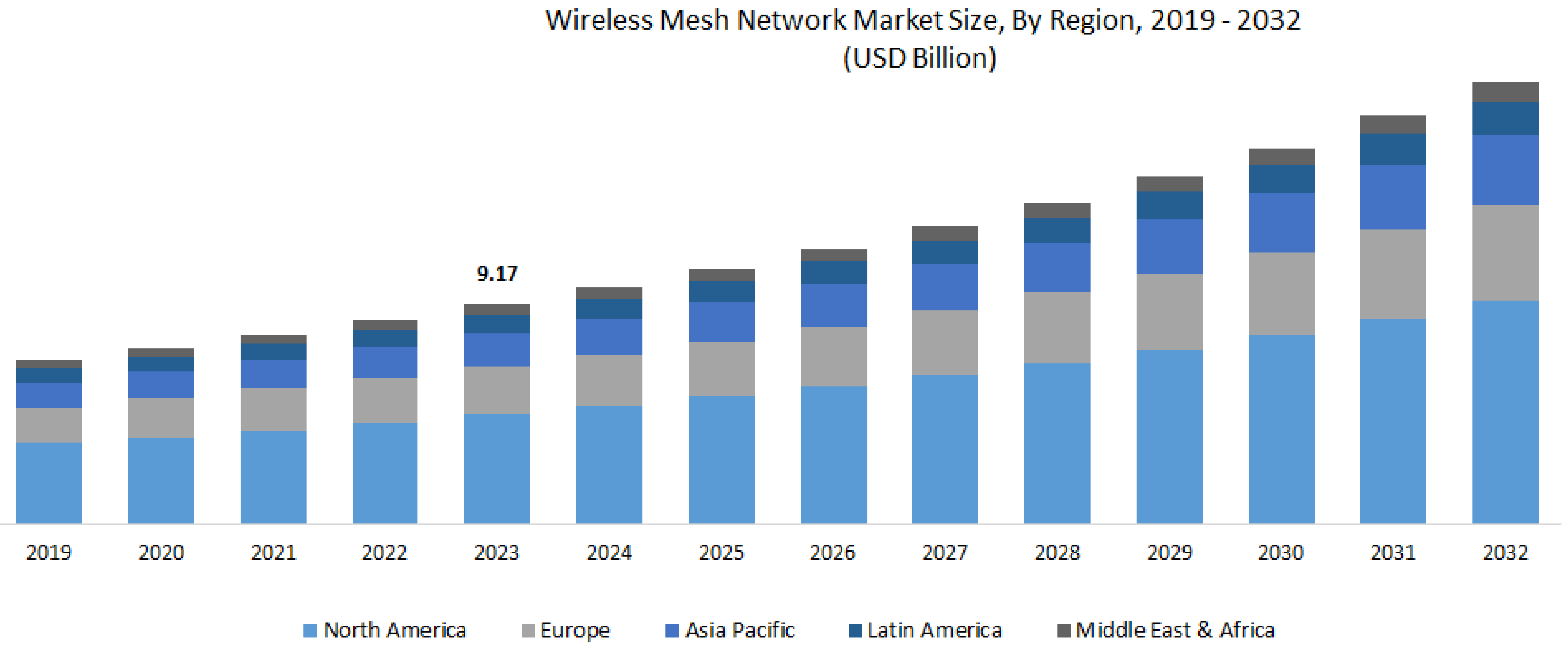
TP-Link Deco

Linksys

ASUS

Images: Netgear, TP-Link, Linksys, ASUS

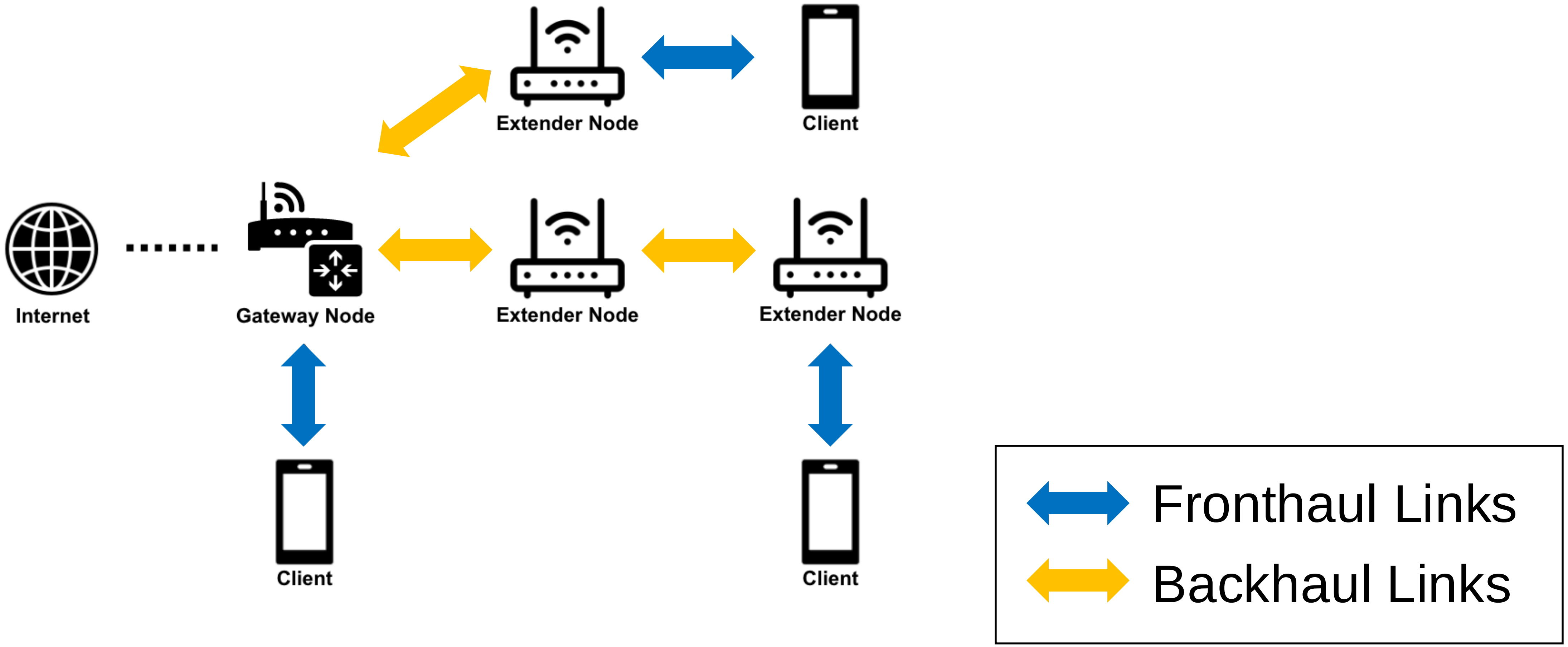
Wireless Mesh Networks are increasingly popular!


Wireless Mesh Network Market Size, By Region, 2019 - 2032 (USD Billion)

Source: Polaris Market Research Analysis

Extending Connectivity in Home Networks with WMNs

- Inter-access-point backhaul links carry both user traffic and configurations.



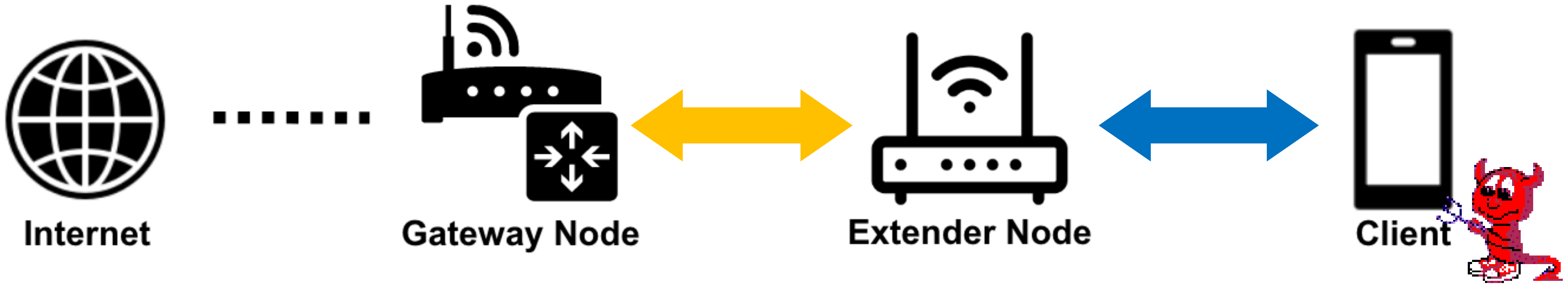
A Motivating Question: How to Change Wi-Fi Passwords?

- Network Access Policy Synchronization (NAPS) helps access points

  Synchronize the Wi-Fi password

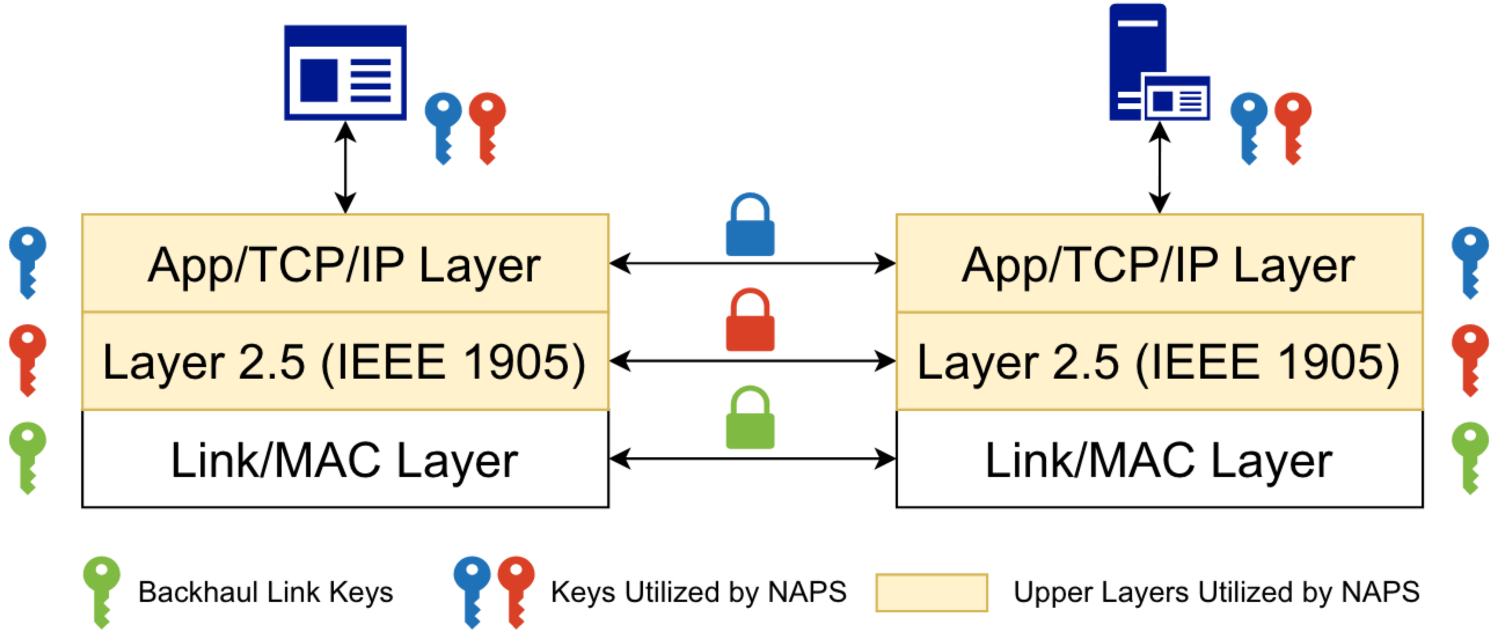
  Switch the SSID

  Update firewall rules, DNS settings, Web UI password…
- A novel attack surface!



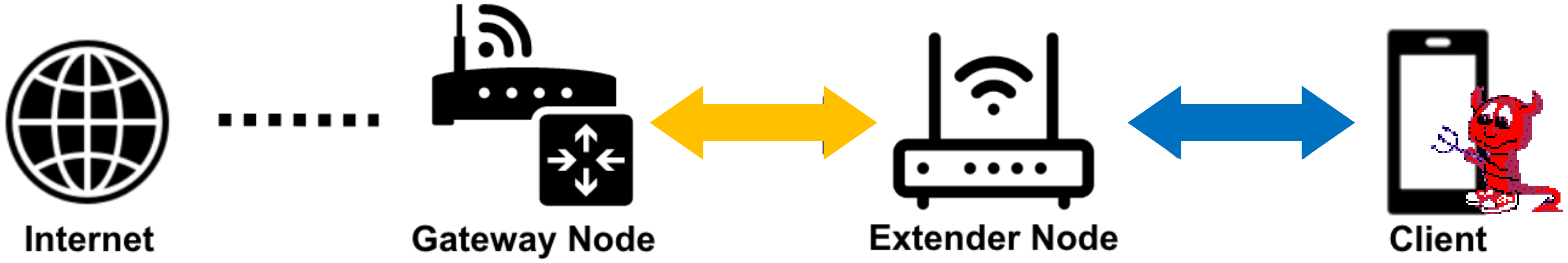
How is NAPS implemented?

- Channels: over backhaul links
- Protocols: ad-hoc crypto protocols and Wi-Fi EasyMesh
- We call them Network Access Policy Synchronization (NAPS) protocols



Threat Model

- A wireless client (attacker) has a fronthaul link credential.
- Can use ARP poisoning to perform MITM attacks.
- Goal 1: To obtain root shell to access points
- Goal 2: To steal WPA2/3 passphrases of backhaul/fronthaul links



Images: Dan Boneh

Overall Results

Vendor	NAPS Protocol	Attack Results
NETGEAR	SOAP over TLS	Root shell
ASUS	AiMesh protocol	Root shell
tp-link	TCP over Dropbear SSH	Root shell
LINKSYS	TLS-SRP	Root shell
WYZE	MQTT with TLS	Wi-Fi password leakage
AMPLIFI	WebSocket with TLS	Wi-Fi password leakage
WiFi ALLIANCE	EasyMesh	Wi-Fi password leakage

Logos are from vendor websites

Security Flaws

1. Type I: Missing cross-layer trust (among mesh nodes)

2. Type II: Cross-layer trust compromise

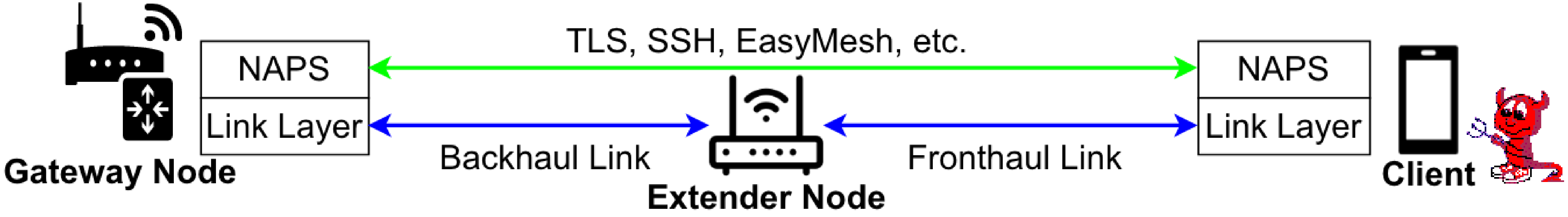
Security Flaws

1. **Type I: Missing cross-layer trust (among mesh nodes)**

2. Type II: Cross-layer trust compromise

Flaw Type I: Missing Cross-layer Trust

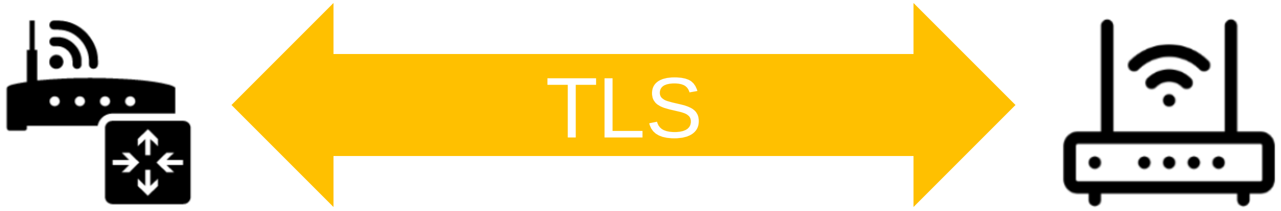
1. Trust at link layer is well-established.
2. No trust anchors for NAPS layer (not bootstrapped properly)
3. Thus, attackers can manipulate NAPS protocols.



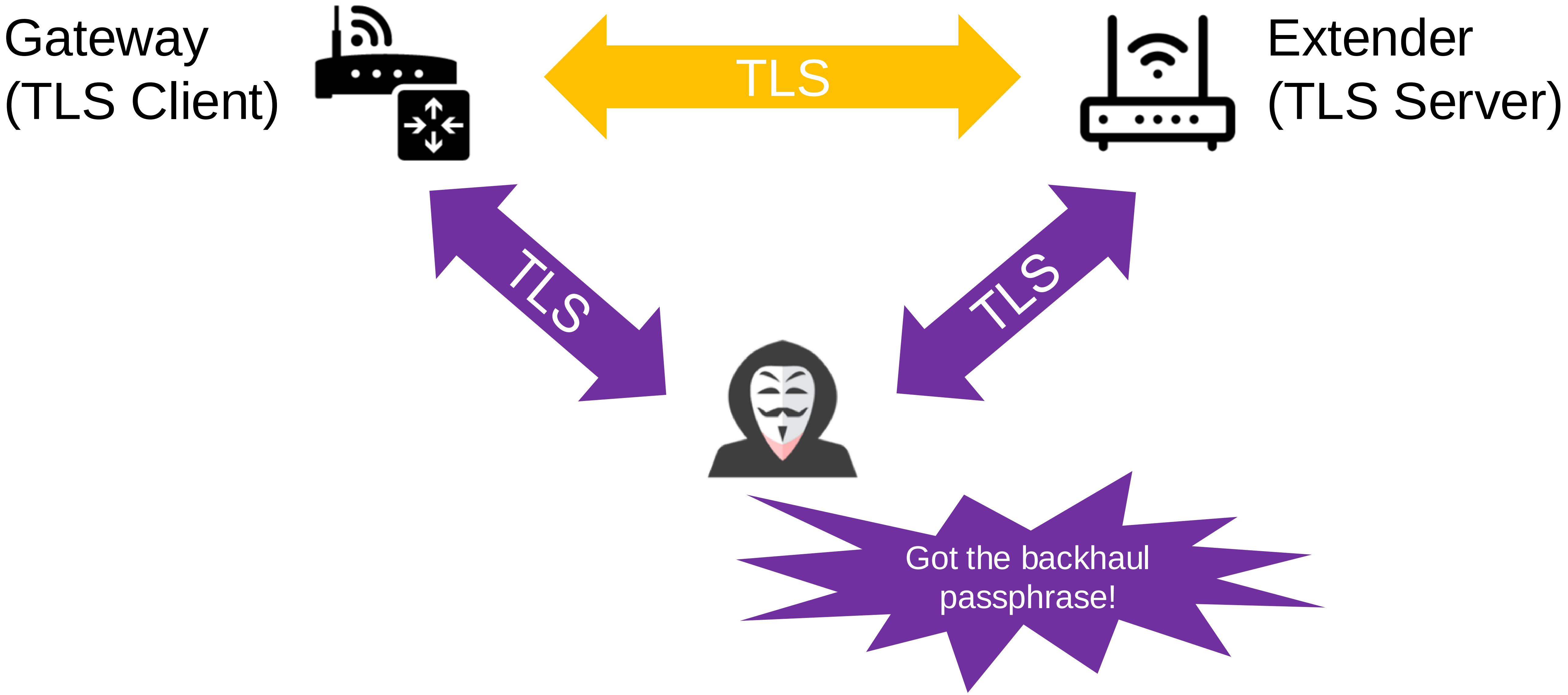
Case Study: Netgear Orbi’s SOAP-over-TLS

Vulnerability:

TLS but self-signed certificates



Attack #1: MITM against SOAP-over-TLS



Gateway
(TLS Client)

Extender
(TLS Server)

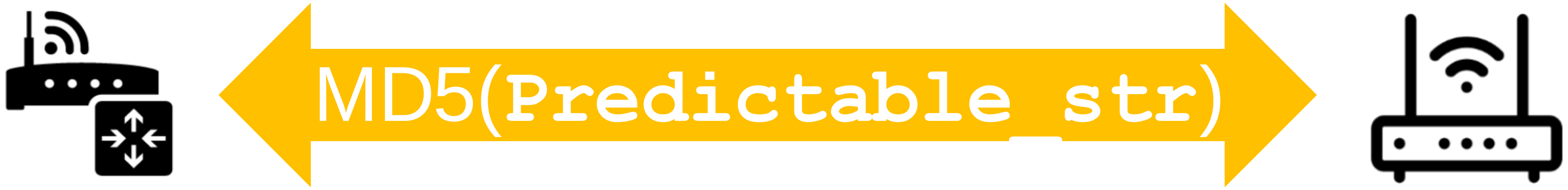
Got the backhaul passphrase!

Case Study: Netgear Orbi’s SOAP-over-TLS

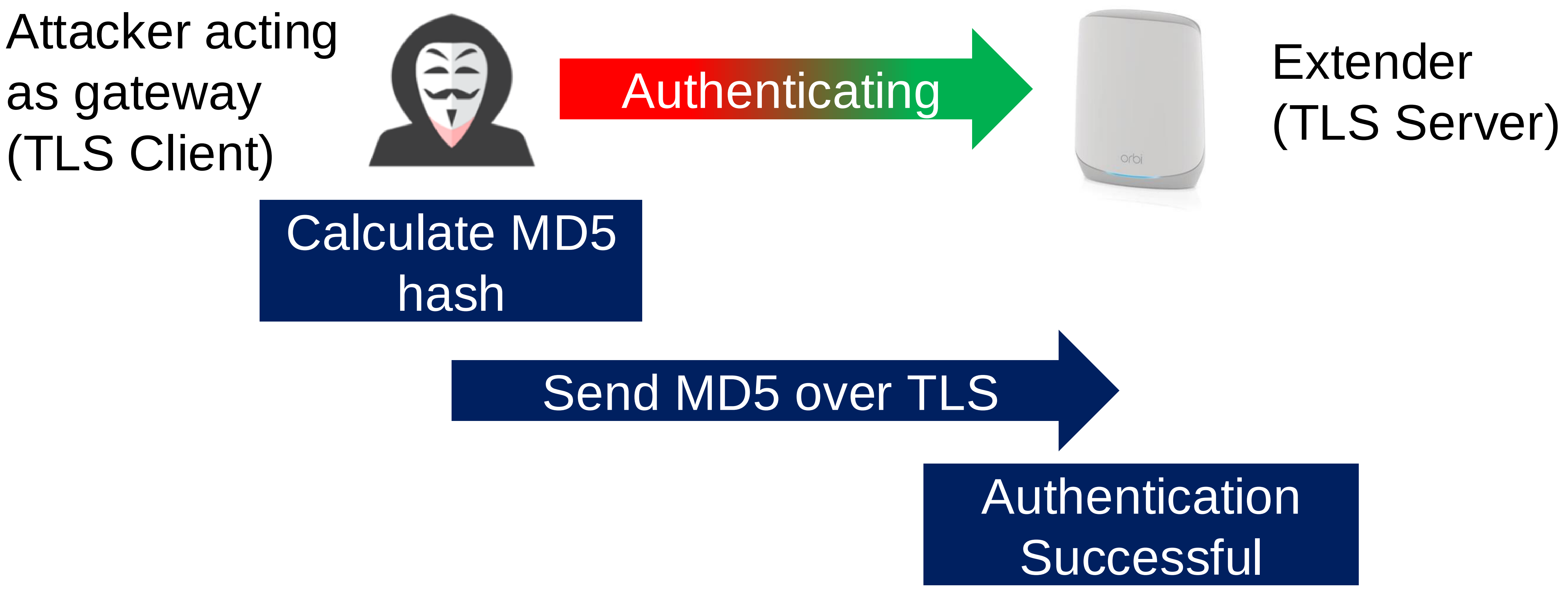
Vulnerability:

Password required for invoking SOAP commands, but fully predictable

Predictable_str =
“NETGEAR_Orbi_<$\text{MAC}_{\text{Gateway}}$>_<$\text{MAC}_{\text{Extender}}$>_password”



Attack #2: Exploiting SOAP-over-TLS (Step 1)



Attacker acting as gateway (TLS Client)

Authenticating

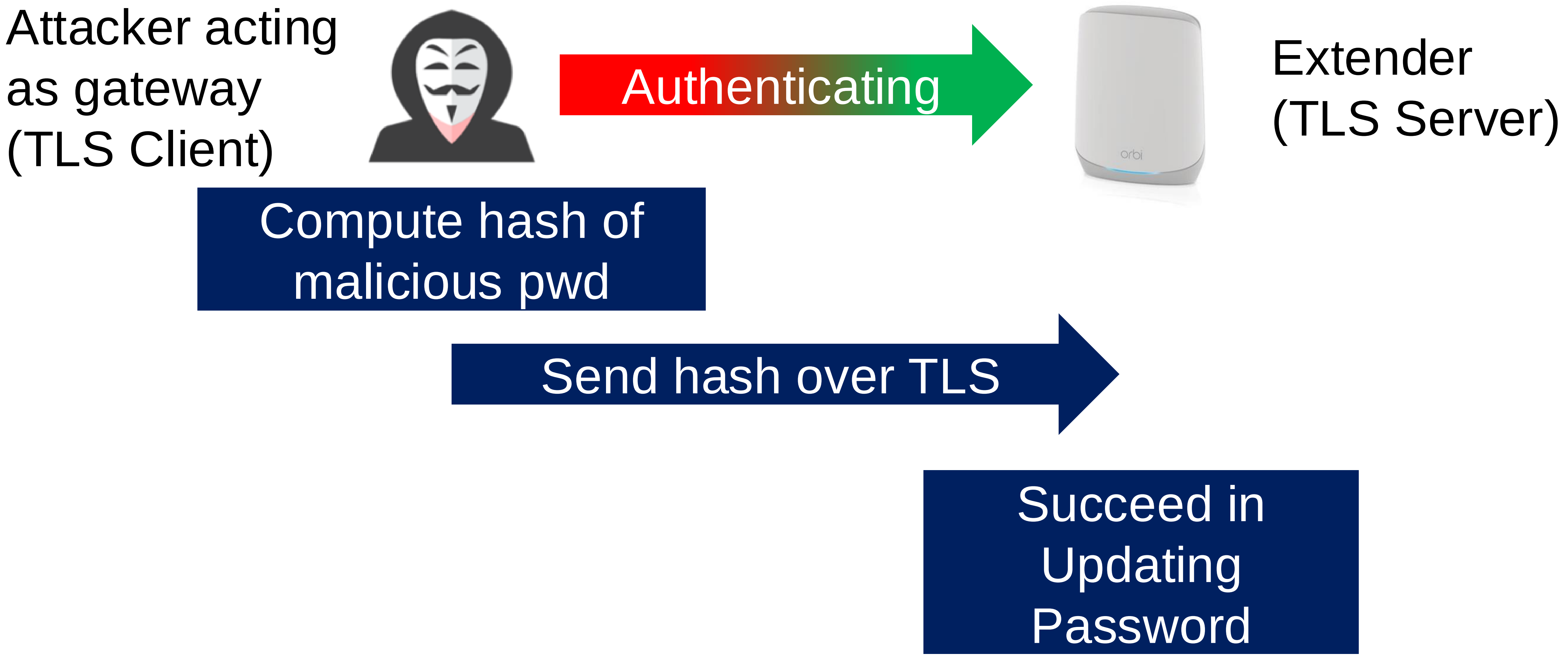
Extender (TLS Server)

Calculate MD5 hash

Send MD5 over TLS

Authentication Successful

Attack #2: Exploiting SOAP-over-TLS (Step 2)



Attacker acting as gateway (TLS Client)

Authenticating

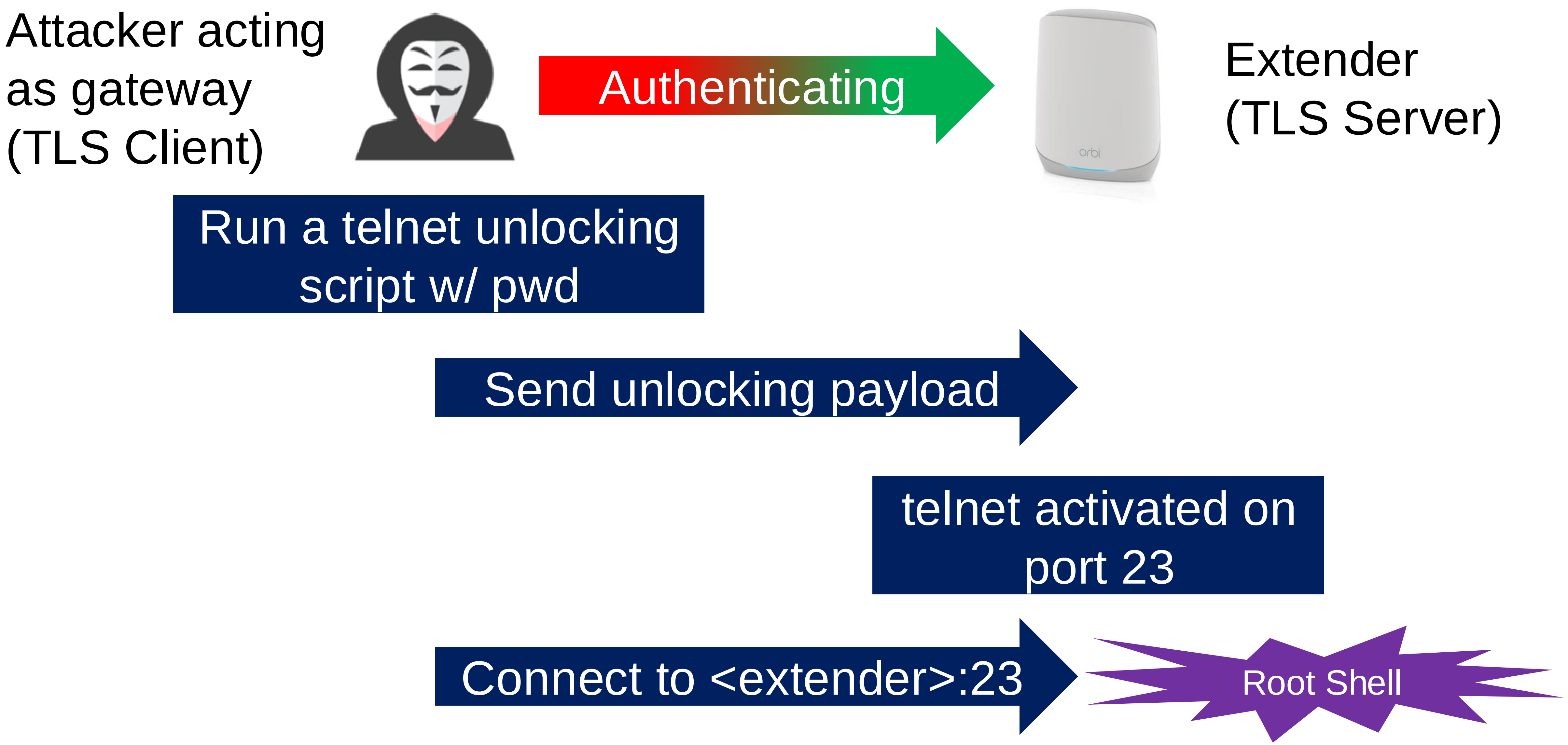
Extender (TLS Server)

Compute hash of malicious pwd

Send hash over TLS

Succeed in Updating Password

Attack #2: Exploiting SOAP-over-TLS (Step 3)



Attacker acting as gateway (TLS Client)

Authenticating

Extender (TLS Server)

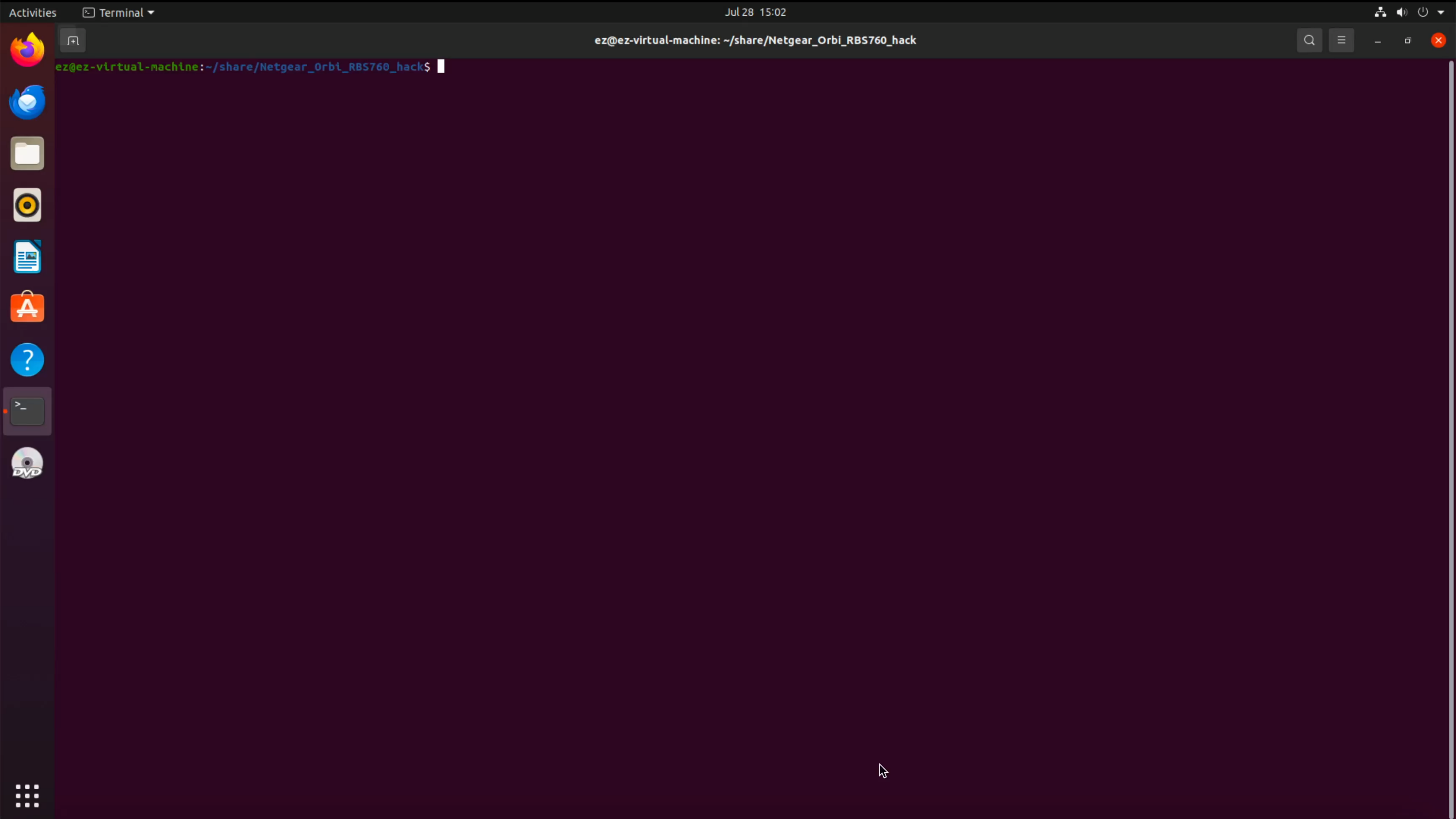
Run a telnet unlocking script w/ pwd

Send unlocking payload

telnet activated on port 23

Connect to <extender>:23

Root Shell



ez@ez-virtual-machine: ~/share/Netgear_Orbi_RBS760_hack

```
ez@ez-virtual-machine:~/share/Netgear_Orbi_RBS760_hack$ 
```

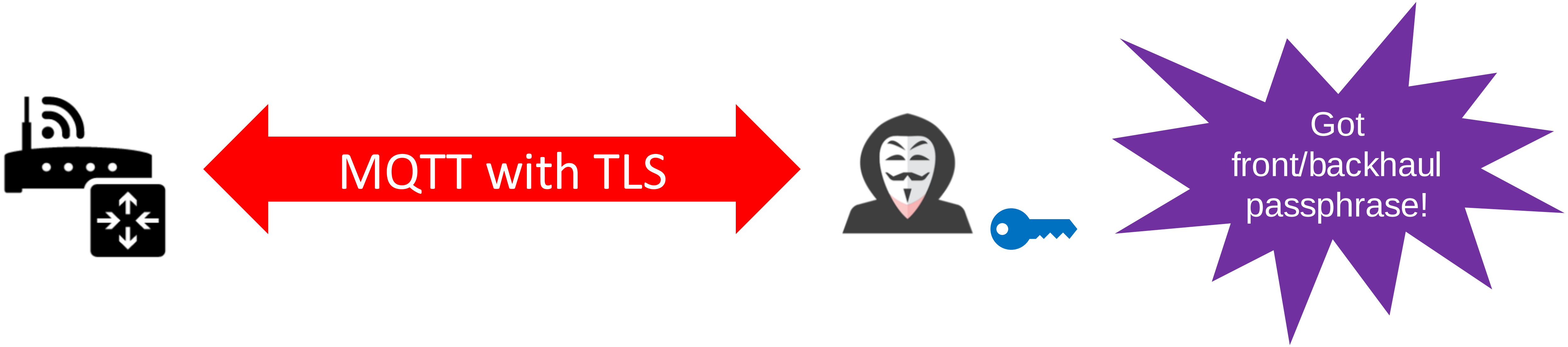
Case Study: Wyze's MQTT with TLS

Vulnerability:

- The key 🔑 for MQTT(S) is shared among ALL Wyze devices

Attack:

- Unpack the firmware, jackpot!
- Attacker wiretaps control data



MQTT with TLS

Got front/backhaul passphrase!

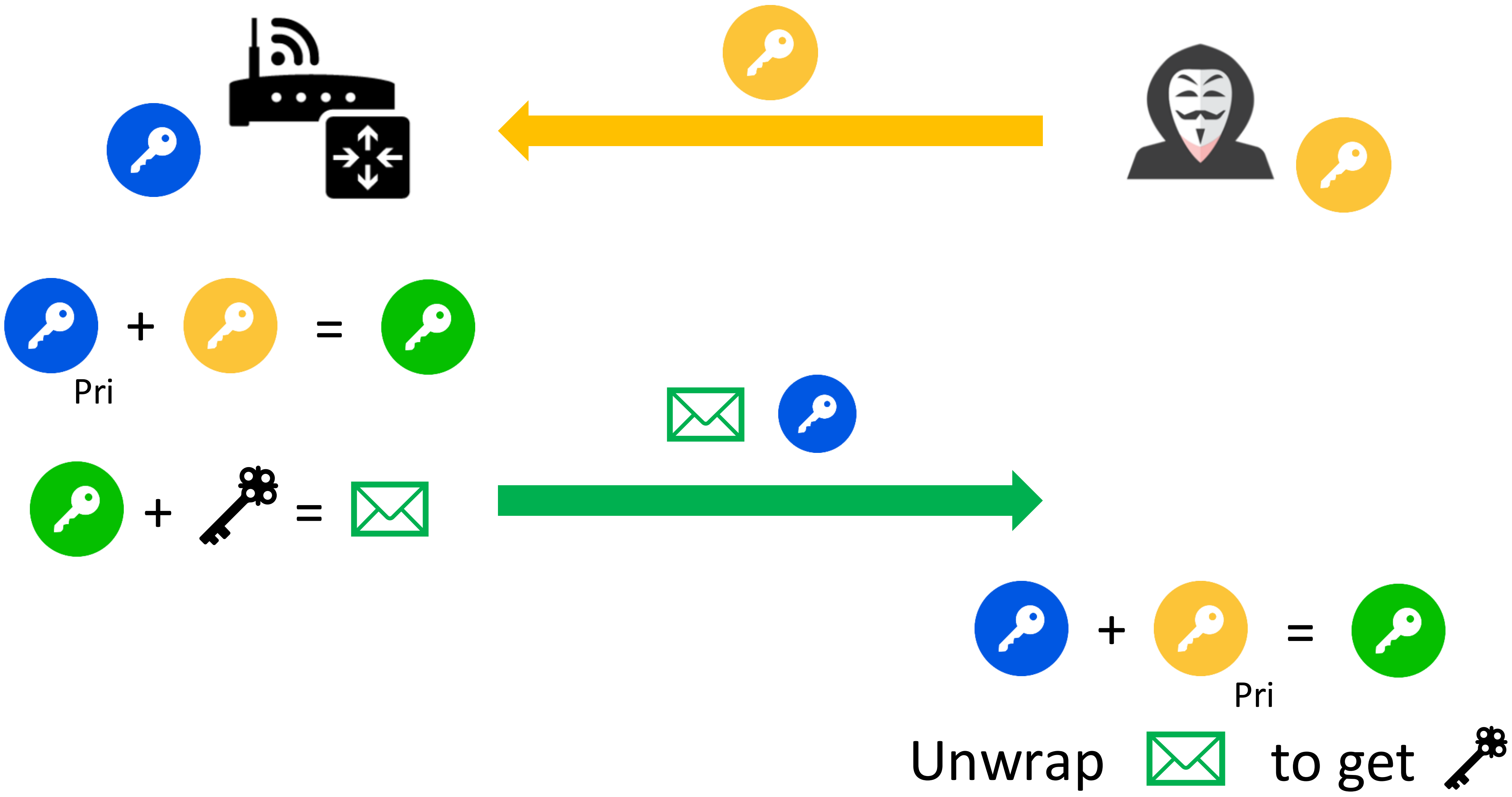
Case Study: AmpliFi's WebSocket with TLS

1. Self-signed certificates for inter-AP TLS connections (again)
2. Fronthaul/backhaul passphrases were wrapped in (unencrypted) MessagePack formats

Example: Wi-Fi EasyMesh standard

- The opt-in standard for NAPS
- No authentication at all
- Uses 2 messages to perform opportunistic encryption in one round-trip time (1 RTT).

PoC: Wi-Fi EasyMesh



🔵 : Key to encrypt 🗝

🗝 : Wi-Fi passphrase

🔵 : Public key of 📡

🟡 : Public key of 👤

Overall Results

Vendor	NAPS Protocol	Attack Results
NETGEAR	SOAP over TLS	Root shell
ASUS	AiMesh protocol	Root shell
tp-link	TCP over Dropbear SSH	Root shell
LINKSYS	TLS-SRP	Root shell
WYZE	MQTT with TLS	Wi-Fi password leakage
AMPLIFI	WebSocket with TLS	Wi-Fi password leakage
Wi Fi ALLIANCE	EasyMesh	Wi-Fi password leakage

Logos are from vendor websites

Security Flaws

1. Type I: Missing cross-layer trust (among mesh nodes)

2. **Type II: Cross-layer trust compromise**

Flaw Type II: Cross-layer Trust Compromise

- NAPS endpoints are reachable by attackers

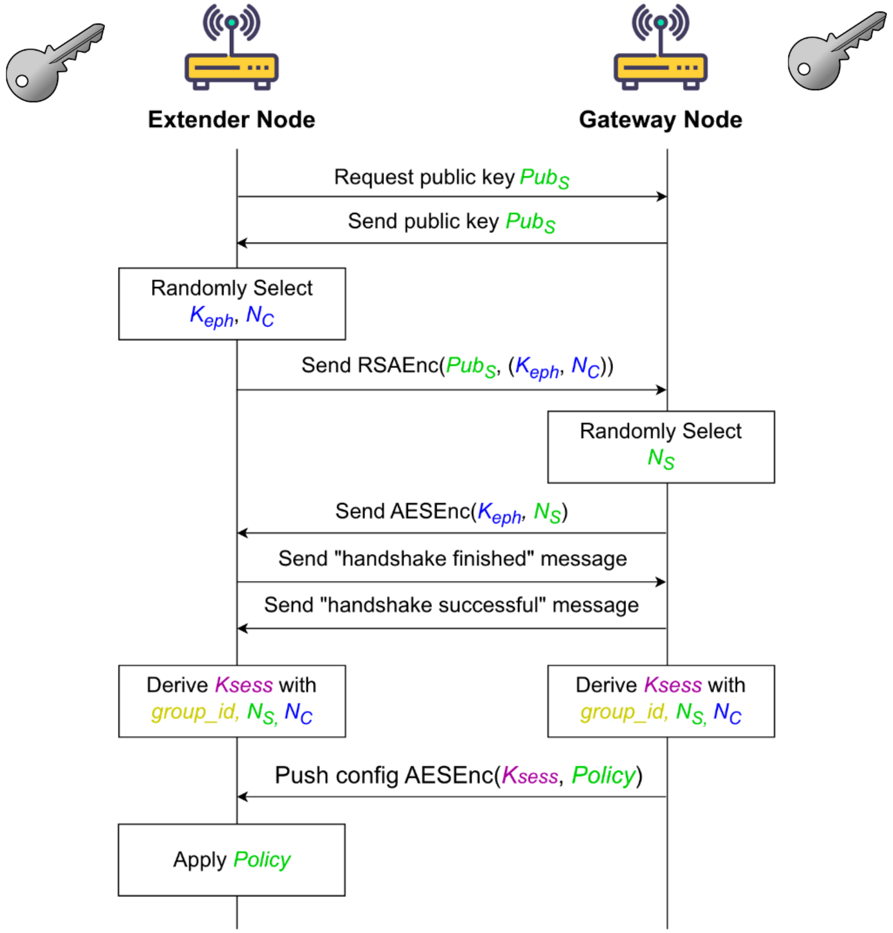
  No logical isolation like VLAN

- Crypto failures and software vulnerabilities are still there

- One layer fails, all layers fail

Case Study: ASUS AiMesh Protocol

1. An encrypted protocol on top of TCP
2. “group_id” 🔑 is the credential



Extender Node

Gateway Node

Request public key Pub_S

Send public key Pub_S

Randomly Select K_{eph}, N_C

Send RSAEnc(Pub_S, (K_{eph}, N_C))

Randomly Select N_S

Send AESEnc(K_{eph}, N_S)

Send "handshake finished" message

Send "handshake successful" message

Derive *Ksess* with *group_id*, N_S, N_C

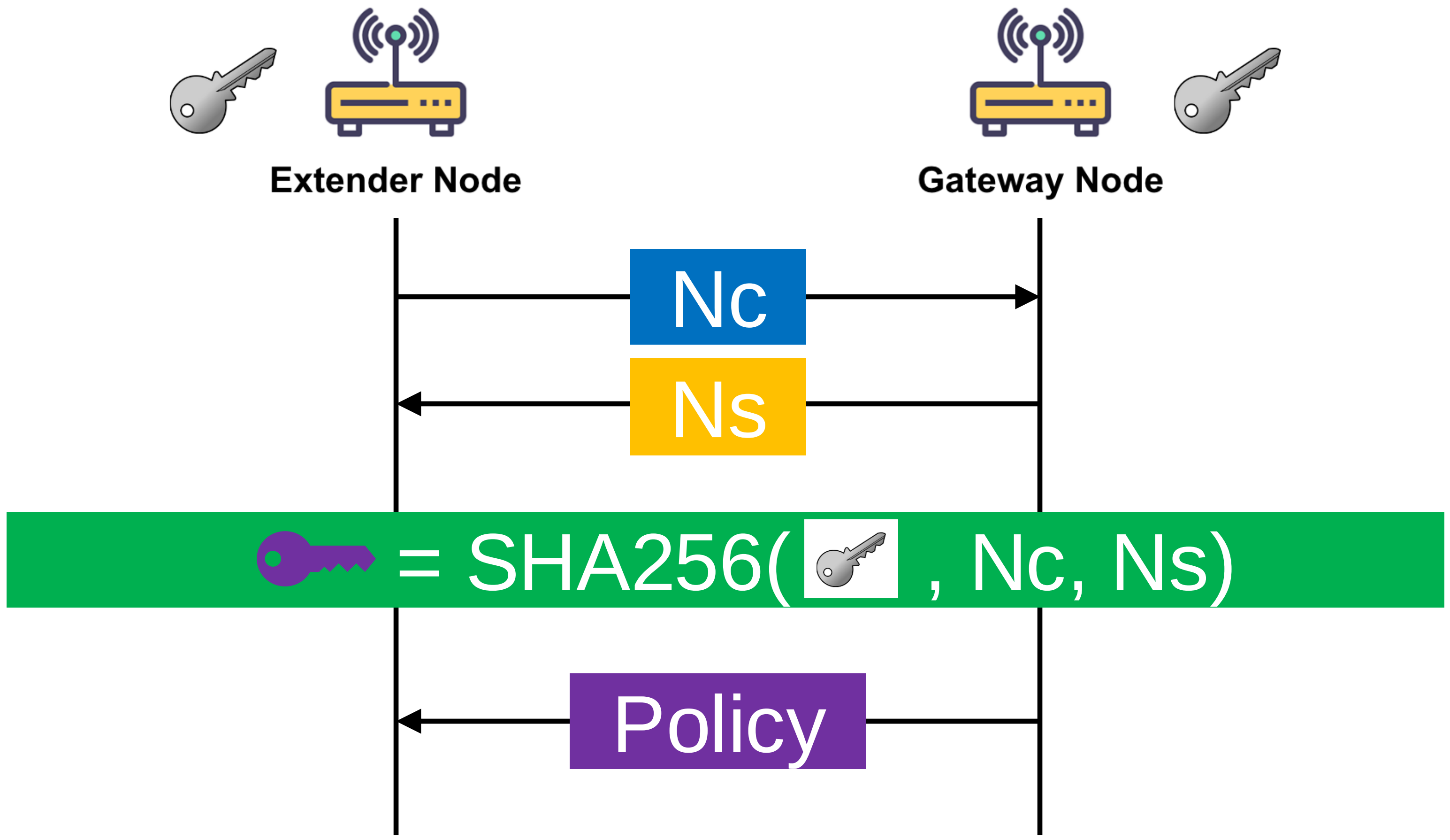
Derive *Ksess* with *group_id*, N_S, N_C

Push config AESEnc(*Ksess*, *Policy*)

Apply *Policy*

Case Study: ASUS AiMesh Protocol

1. An encrypted protocol on top of TCP
2. “group_id” 🔑 is the credential



Extender Node

Gateway Node

Nc

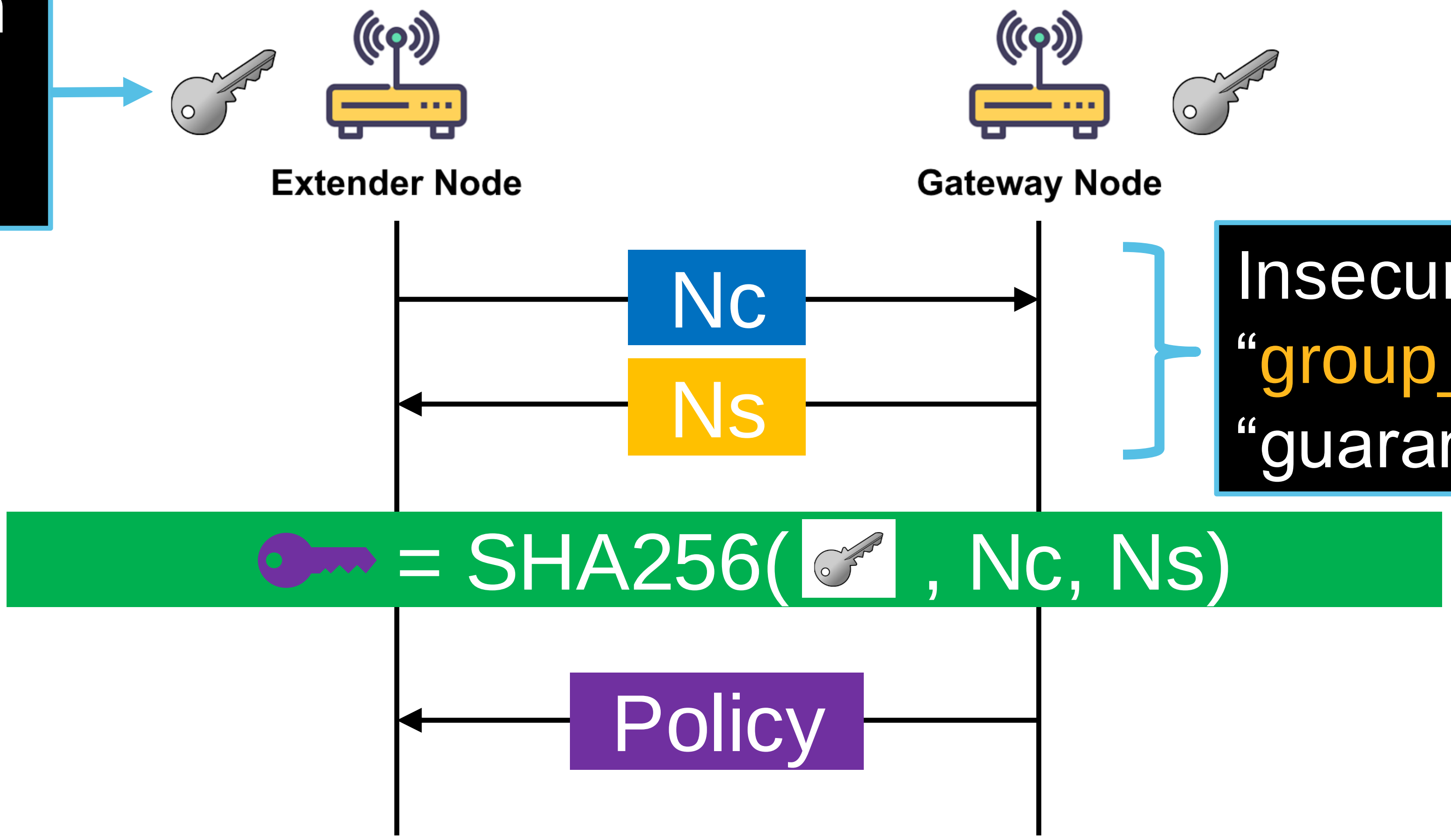
Ns

🔑 = SHA256(🔑 , Nc, Ns)

Policy

ASUS AiMesh protocol is vulnerable to key leakage

“group_id” leaked in the 802.11 layer, breaking security.

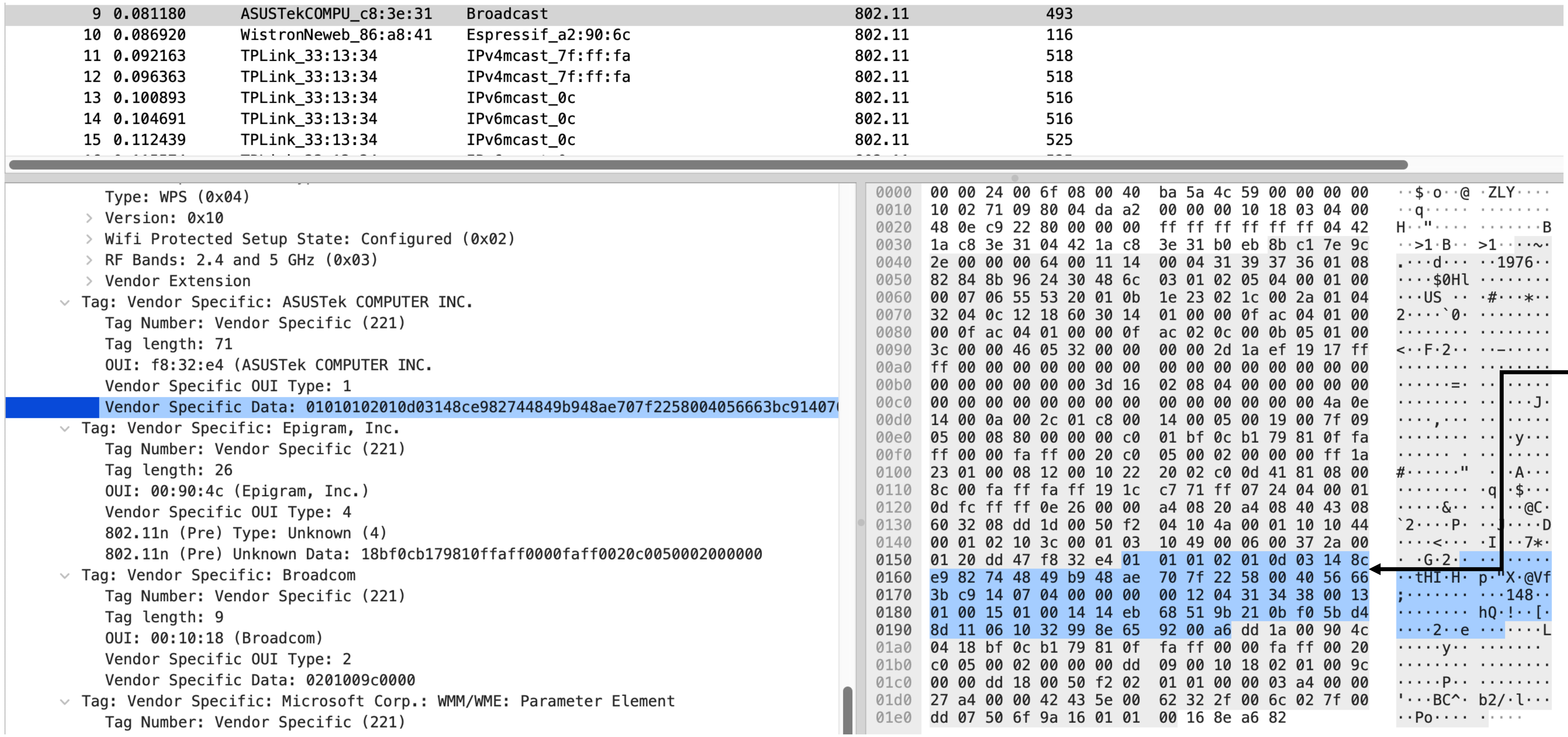


Insecure, however “group_id” “guarantees” security later.

Leaked group_id

1. “group_id” 🔑 is broadcasted at the 802.11 layer
   - Just sniff for the hashed “group_id” over-the-air
   - Offline brute force to crack the “group_id”

Leaked group_id



```
  9 0.081180   ASUSTekCOMPU_c8:3e:31    Broadcast              802.11      493
 10 0.086920   WistronNeweb_86:a8:41    Espressif_a2:90:6c     802.11      116
 11 0.092163   TPLink_33:13:34          IPv4mcast_7f:ff:fa     802.11      518
 12 0.096363   TPLink_33:13:34          IPv4mcast_7f:ff:fa     802.11      518
 13 0.100893   TPLink_33:13:34          IPv6mcast_0c           802.11      516
 14 0.104691   TPLink_33:13:34          IPv6mcast_0c           802.11      516
 15 0.112439   TPLink_33:13:34          IPv6mcast_0c           802.11      525
```

```
    Type: WPS (0x04)
  > Version: 0x10
  > Wifi Protected Setup State: Configured (0x02)
  > RF Bands: 2.4 and 5 GHz (0x03)
  > Vendor Extension
⌄ Tag: Vendor Specific: ASUSTek COMPUTER INC.
    Tag Number: Vendor Specific (221)
    Tag length: 71
    OUI: f8:32:e4 (ASUSTek COMPUTER INC.
    Vendor Specific OUI Type: 1
    Vendor Specific Data: 01010102010d03148ce982744849b948ae707f2258004056663bc91407
⌄ Tag: Vendor Specific: Epigram, Inc.
    Tag Number: Vendor Specific (221)
    Tag length: 26
    OUI: 00:90:4c (Epigram, Inc.)
    Vendor Specific OUI Type: 4
    802.11n (Pre) Type: Unknown (4)
    802.11n (Pre) Unknown Data: 18bf0cb179810ffaff0000faff0020c0050002000000
⌄ Tag: Vendor Specific: Broadcom
    Tag Number: Vendor Specific (221)
    Tag length: 9
    OUI: 00:10:18 (Broadcom)
    Vendor Specific OUI Type: 2
    Vendor Specific Data: 0201009c0000
⌄ Tag: Vendor Specific: Microsoft Corp.: WMM/WME: Parameter Element
    Tag Number: Vendor Specific (221)
```

```
0000  00 00 24 00 6f 08 00 40  ba 5a 4c 59 00 00 00 00   ··$·o··@ ·ZLY····
0010  10 02 71 09 80 04 da a2  00 00 00 10 18 03 04 00   ··q····· ········
0020  48 0e c9 22 80 00 00 00  ff ff ff ff ff ff 04 42   H··"···· ·······B
0030  1a c8 3e 31 04 42 1a c8  3e 31 b0 eb 8b c1 7e 9c   ··>1·B·· >1····~·
0040  2e 00 00 00 64 00 11 14  00 04 31 39 37 36 01 08   .···d··· ··1976··
0050  82 84 8b 96 24 30 48 6c  03 01 02 05 04 00 01 00   ····$0Hl ········
0060  00 07 06 55 53 20 01 0b  1e 23 02 1c 00 2a 01 04   ···US ·· ·#···*··
0070  32 04 0c 12 18 60 30 14  01 00 00 0f ac 04 01 00   2····`0· ········
0080  00 0f ac 04 01 00 00 0f  ac 02 0c 00 0b 05 01 00   ········ ········
0090  3c 00 00 46 05 32 00 00  00 00 2d 1a ef 19 17 ff   <··F·2·· ··–·····
00a0  ff 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
00b0  00 00 00 00 00 00 3d 16  02 08 04 00 00 00 00 00   ······=· ········
00c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 4a 0e   ········ ······J·
00d0  14 00 0a 00 2c 01 c8 00  14 00 05 00 19 00 7f 09   ····,··· ········
00e0  05 00 08 80 00 00 00 c0  01 bf 0c b1 79 81 0f fa   ········ ····y···
00f0  ff 00 00 fa ff 00 20 c0  05 00 02 00 00 00 ff 1a   ······ · ········
0100  23 01 00 08 12 00 10 22  20 02 c0 0d 41 81 08 00   #······"  ···A···
0110  8c 00 fa ff fa ff 19 1c  c7 71 ff 07 24 04 00 01   ········ ·q··$···
0120  0d fc ff ff 0e 26 00 00  a4 08 20 a4 08 40 43 08   ·····&·· ·· ··@C·
0130  60 32 08 dd 1d 00 50 f2  04 10 4a 00 01 10 10 44   `2····P· ··J····D
0140  00 01 02 10 3c 00 01 03  10 49 00 06 00 37 2a 00   ····<··· ·I···7*·
0150  01 20 dd 47 f8 32 e4 01  01 01 02 01 0d 03 14 8c   · ·G·2·· ········
0160  e9 82 74 48 49 b9 48 ae  70 7f 22 58 00 40 56 66   ··tHI·H· p·"X·@Vf
0170  3b c9 14 07 04 00 00 00  00 12 04 31 34 38 00 13   ;······· ···148··
0180  01 00 15 01 00 14 14 eb  68 51 9b 21 0b f0 5b d4   ········ hQ·!··[·
0190  8d 11 06 10 32 99 8e 65  92 00 a6 dd 1a 00 90 4c   ····2··e ·······L
01a0  04 18 bf 0c b1 79 81 0f  fa ff 00 00 fa ff 00 20   ·····y·· ········
01b0  c0 05 00 02 00 00 00 dd  09 00 10 18 02 01 00 9c   ········ ········
01c0  00 00 dd 18 00 50 f2 02  01 01 00 00 03 a4 00 00   ·····P·· ········
01d0  27 a4 00 00 42 43 5e 00  62 32 2f 00 6c 02 7f 00   '···BC^· b2/·l···
01e0  dd 07 50 6f 9a 16 01 01  00 16 8e a6 82            ··Po···· ·····
```

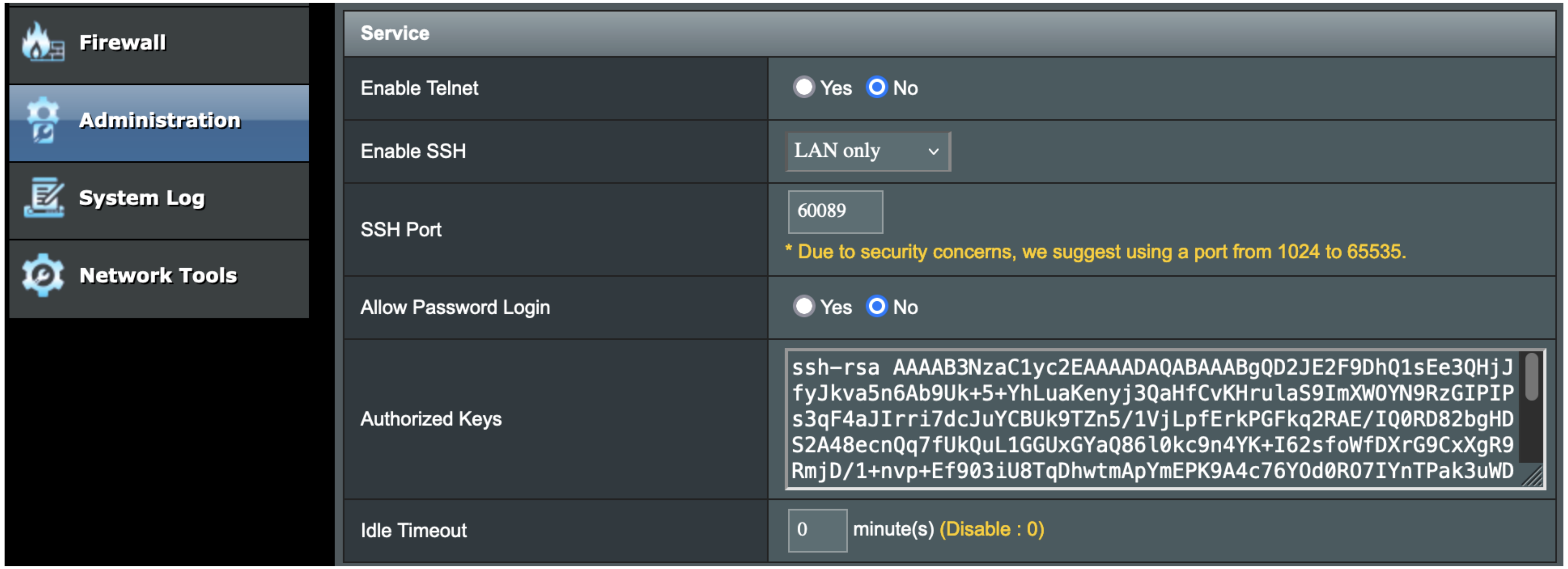
Type-Length-Value (TLV) structure.

Hash of “group_id” is stored at type 0x3

ASUS AiMesh protocol is vulnerable to key leakage

2. The attacker can then tamper with (encrypted) AiMesh connections.

- To exploit `cfg_server`’s SSH management key installation functionality to gain root access.



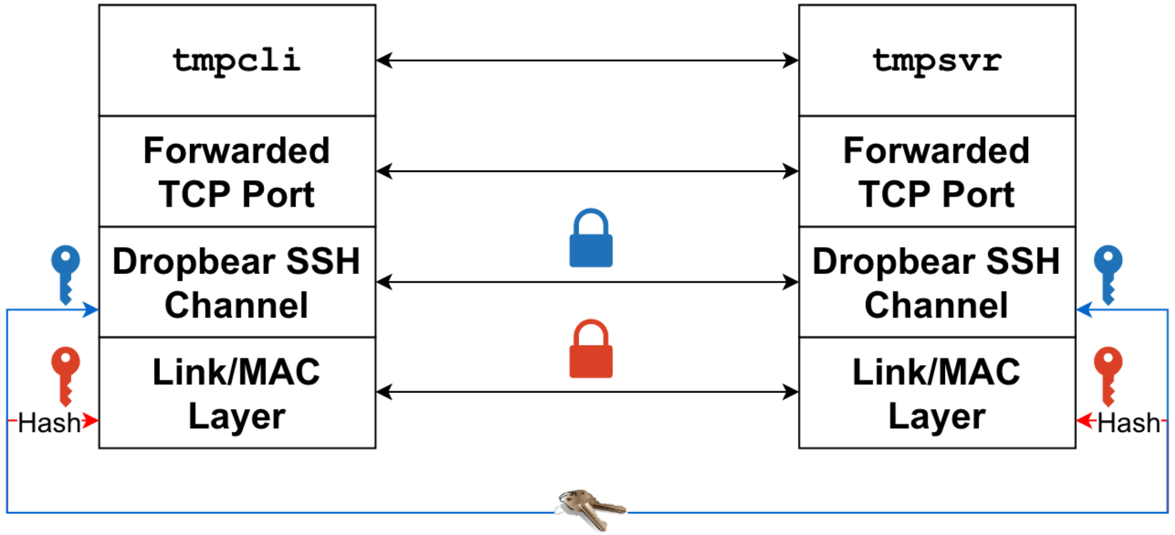
Service

Enable Telnet	○ Yes ● No
Enable SSH	LAN only
SSH Port	60089 * Due to security concerns, we suggest using a port from 1024 to 65535.
Allow Password Login	○ Yes ● No
Authorized Keys	ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQD2JE2F9DhQ1sEe3QHjJfyJkva5n6Ab9Uk+5+YhLuaKenyj3QaHfCvKHrulaS9ImXWOYN9RzGIPIPs3qF4aJIrri7dcJuYCBUk9TZn5/1VjLpfErkPGFkq2RAE/IQ0RD82bgHDS2A48ecnQq7fUkQuL1GGUxGYaQ86l0kc9n4YK+I62sfoWfDXrG9CxXgR9RmjD/1+nvp+Ef903iU8TqDhwtmApYmEPK9A4c76YOd0RO7IYnTPak3uWD
Idle Timeout	0 minute(s) (Disable : 0)

TP-Link Deco: Weak SSH key and command injections

1. Channel: Dropbear SSH with 512-bit RSA key length.

- Brute force an RSA private key in 4 days with a single PC in 2024.
- Software: GGNFS/MSIEVE



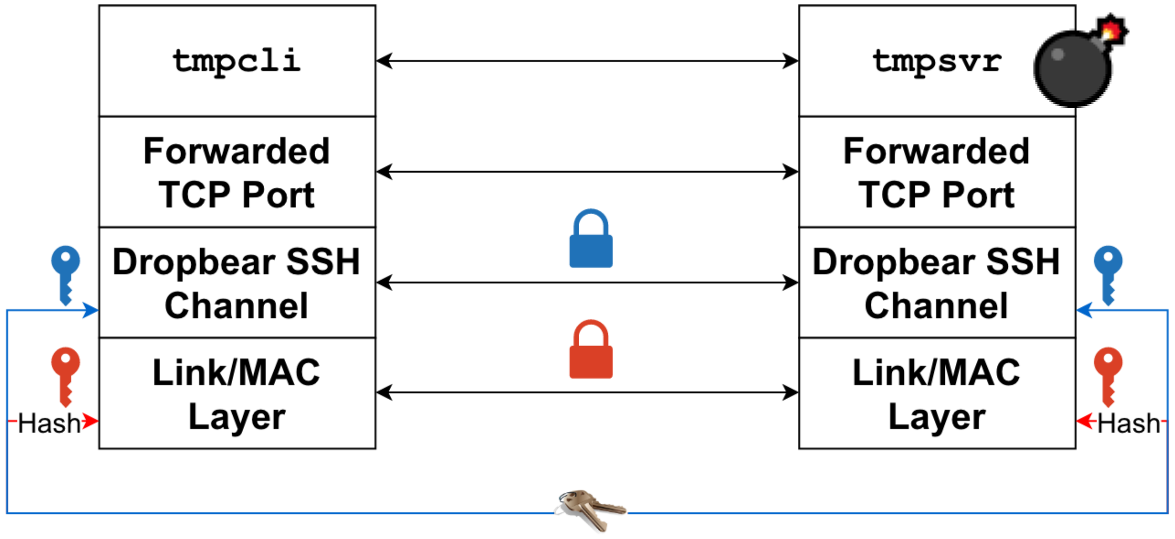
RSA Key Pair

TP-Link Deco: Weak SSH key and command injections

2. Backhaul passphrases are derived from that RSA key pair.

- Irrevocable access to the network through backhaul links!

3. To exploit command injections 💣 in the `tmpsvr` binary



RSA Key Pair

Linksys: TLS-SRP Isn't the Silver Bullet

1. A zero-knowledge (ZK) protocol encrypting all control data.

cryptographic verifiers
≈ public key

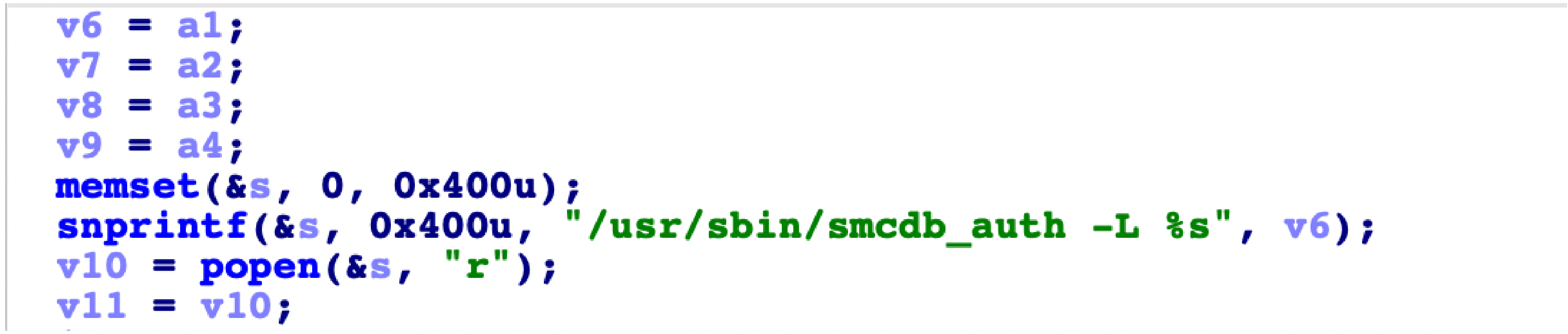
SRP passwords
≈ private key

A machine-in-the-middle truly knows nothing about transmitted data.

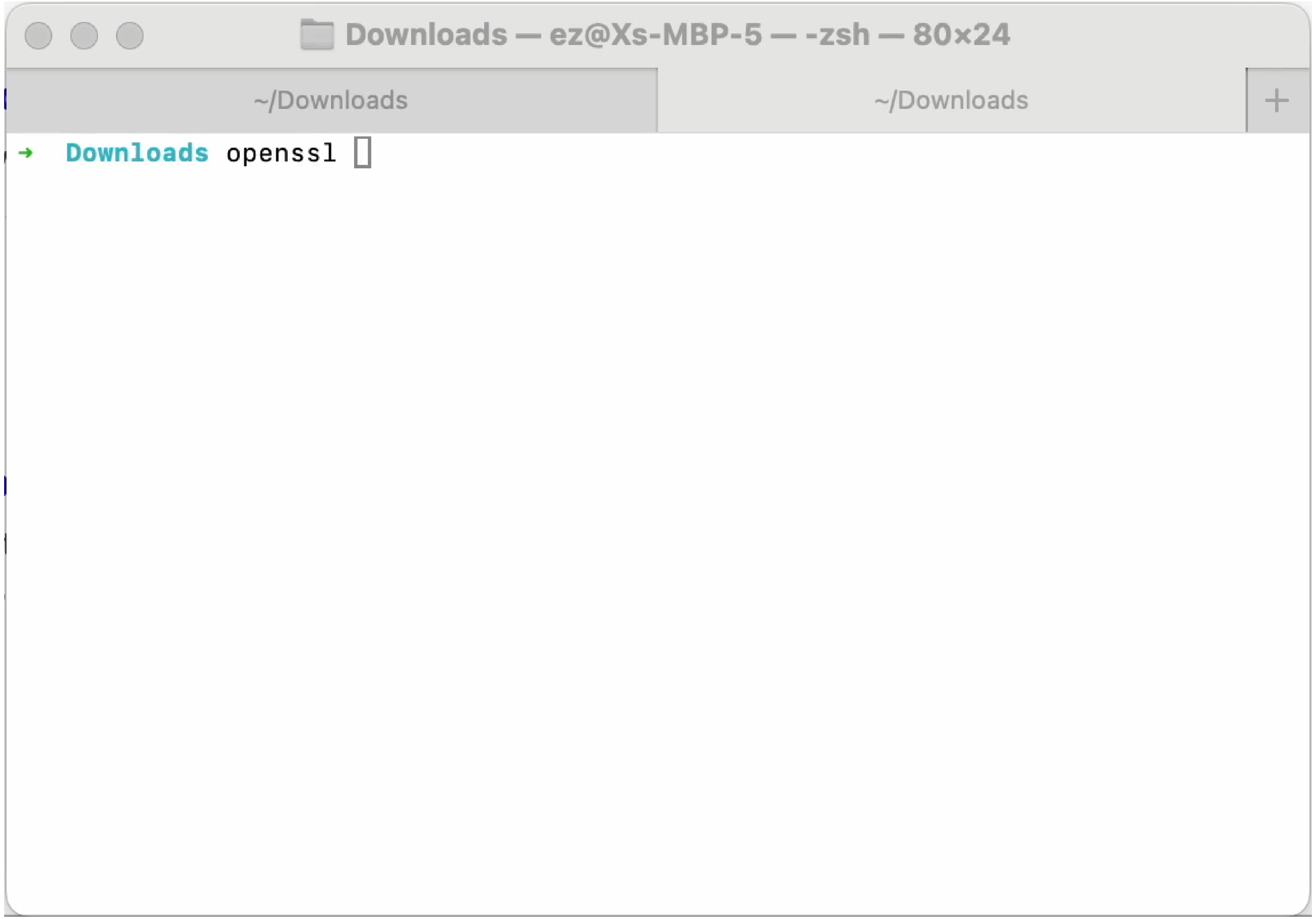
Linksys: TLS-SRP Isn't the Silver Bullet

2. Pre-authentication command injection.

- An attacker can taint the *clientID/srpuser* field
- Steal ***stored SRP passwords*** 💥



```
v6 = a1;
v7 = a2;
v8 = a3;
v9 = a4;
memset(&s, 0, 0x400u);
snprintf(&s, 0x400u, "/usr/sbin/smcdb_auth -L %s", v6);
v10 = popen(&s, "r");
v11 = v10;
```



```
➜  Downloads openssl
```

Mitigation Status (Disclosed > 8 months ago)

Vendor	Attack Results	Patched?
NETGEAR	Root shell	✅
ASUS	Root shell	✅
tp-link	Root shell	✅
LINKSYS	Root shell	⚠
WYZE	Wi-Fi password leakage	✅
AMPLIFI	Wi-Fi password leakage	✅
Wi-Fi ALLIANCE	Wi-Fi password leakage	⚠

Logos are from vendor websites

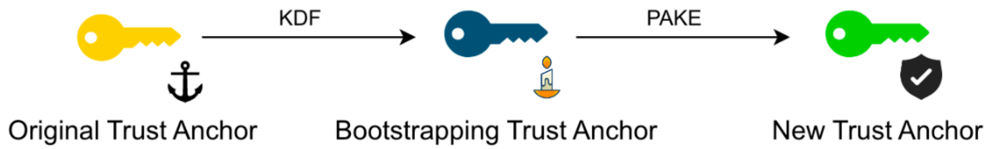
Defenses

Users

- Go home and update the firmware!
- Set a new Wi-Fi password.
- Check your wireless client list for any anomalies.

Network Engineers

- Rotate compromised keys to new values unknown to previous attackers.



KDF

PAKE

Original Trust Anchor

Bootstrapping Trust Anchor

New Trust Anchor

- Add some network isolations.
- Check out our paper for details.

Black Hat Sound Bytes

1. Wireless security is coming back
2. Home WMN control protocols are novel attack surfaces
3. Wireless standards and vendors can do more with security

Thank you!

Github Link:

https://github.com/seclab-ucr/CCS24Mesh

Research Paper:

Untangling the Knot: Breaking Access Control in Home Wireless Mesh Networks, CCS ’24

https://www.cs.ucr.edu/~zhiyunq/pub/ccs24_wireless_mesh.pdf

Feel free to talk to us offline in the hallway!

Contacts:

xinan.zhou@email.ucr.edu X (Twitter): @zhouxinan

zhiyunq@cs.ucr.edu X (Twitter): @pkqzy888