Hop: A Modern Transport and Remote Access Protocol

Paul Flammarion, George Hosono, Wilson Nguyen, Laura Bauman,

Daniel Rebelsky, Gerry Wan, David Adrian, Zakir Durumeric

The History of Remote Communication Protocols

Background

			?
Telnet, 1970s	**SSH-1, 1995**	**SSH-2, 2006**	**?, 2026**
Developed for ARPANET	Tatu Ylönen	Internet Engineering Task Force	Research Community

SSH Vulnerabilities Over Time

Background

Timing Analysis of Keystrokes and Timing Attacks on SSH*

Dawn Xiaodong Song David Wagner Xuqing Tian
University of California, Berkeley

2001

Do Users Verify SSH Keys?

PETER GUTMANN

2011

Peter Gutmann is a researcher in the Department of Computer Science at the University of Auckland. He works on design and analysis

Abstract

No.

Discussion

Plaintext Recovery Attacks Against SSH

2009

Martin R. Albrecht, Kenneth G. Paterson and Gaven J. Watson
Information Security Group
Royal Holloway, University of London
Egham, Surrey, UK
Email: {m.r.albrecht,kenny.paterson,g.watson}@rhul.ac.uk

Transcript Collision Attacks: Breaking Authentication in TLS, IKE, and SSH

Karthikeyan Bhargavan
INRIA
karthikeyan.bhargavan@inria.fr

Gaëtan Leurent
INRIA
gaetan.leurent@normalesup.org

2016

Terrapin Attack: Breaking SSH Channel Integrity By Sequence Number Manipulation

Fabian Bäumer, Marcus Brinkmann, and Jörg Schwenk, *Ruhr University Bochum*

https://www.usenix.org/conference/usenixsecurity24/presentation/bäumer

2024

Finding SSH Strict Key Exchange Violations by State Learning

Fabian Bäumer
Ruhr University Bochum
Bochum, Germany
fabian.baeumer@rub.de

Marcel Maehren
Ruhr University Bochum
Bochum, Germany
marcel.maehren@rub.de

Marcus Brinkmann
Ruhr University Bochum
Bochum, Germany
marcus.brinkmann@rub.de

Jörg Schwenk
Ruhr University Bochum
Bochum, Germany
joerg.schwenk@rub.de

2025

On the Security of SSH Client Signatures

Fabian Bäumer
Ruhr University Bochum
Bochum, Germany
fabian.baeumer@rub.de

Marcus Brinkmann
Ruhr University Bochum
Bochum, Germany
marcus.brinkmann@rub.de

Maximilian Radoy
University Paderborn
Paderborn, Germany
maximilian.radoy@upb.d

Jörg Schwenk
Ruhr University Bochum
Bochum, Germany
joerg.schwenk@rub.de

Juraj Somorovsky
University Paderborn
Paderborn, Germany
juraj.somorovsky@upb.de

2025

Catch-22: Uncovering Compromised Hosts using SSH Public Keys

2025

Cristian Munteanu
Max Planck Institute for Informatics

Georgios Smaragdakis
Delft University of Technology

Anja Feldmann
Max Planck Institute for Informatics

Tobias Fiebig
Max Planck Institute for Informatics

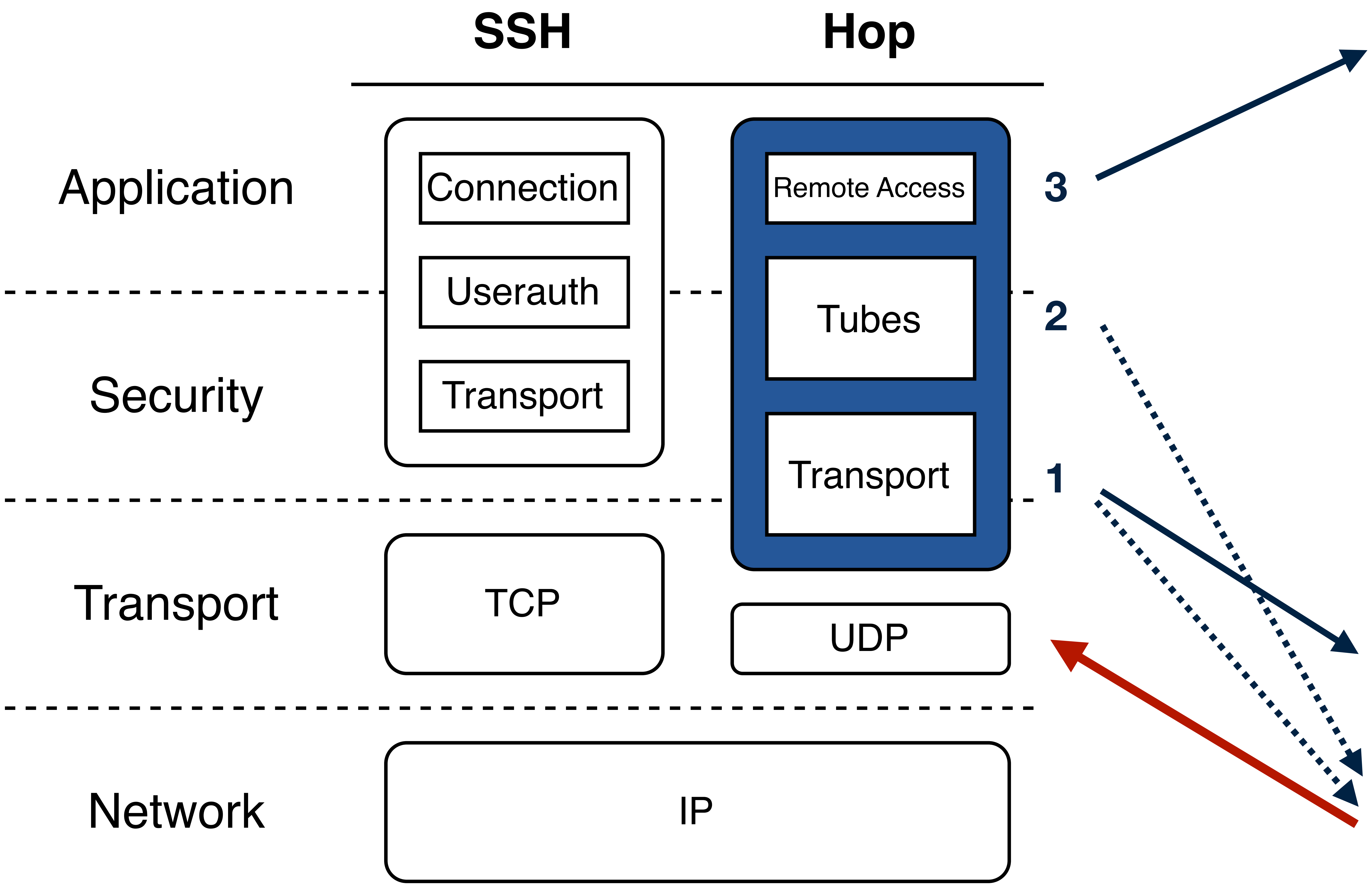
Introduction to Hop

Background

Telnet, 1970s	**SSH-1, 1995**	**SSH-2, 2006**	**Hop, 2026**
Developed for ARPANET	Tatu Ylönen	Internet Engineering Task Force	Research Community

Three Inner Sub-Protocols

Protocol Overview



Protocol Requirements

8 - Secure Credential Delegation

7 - Constrained Environment Support

6 - Extensible Client Verification

5 - Trustworthy Host Identification

4 - Post-Quantum Security

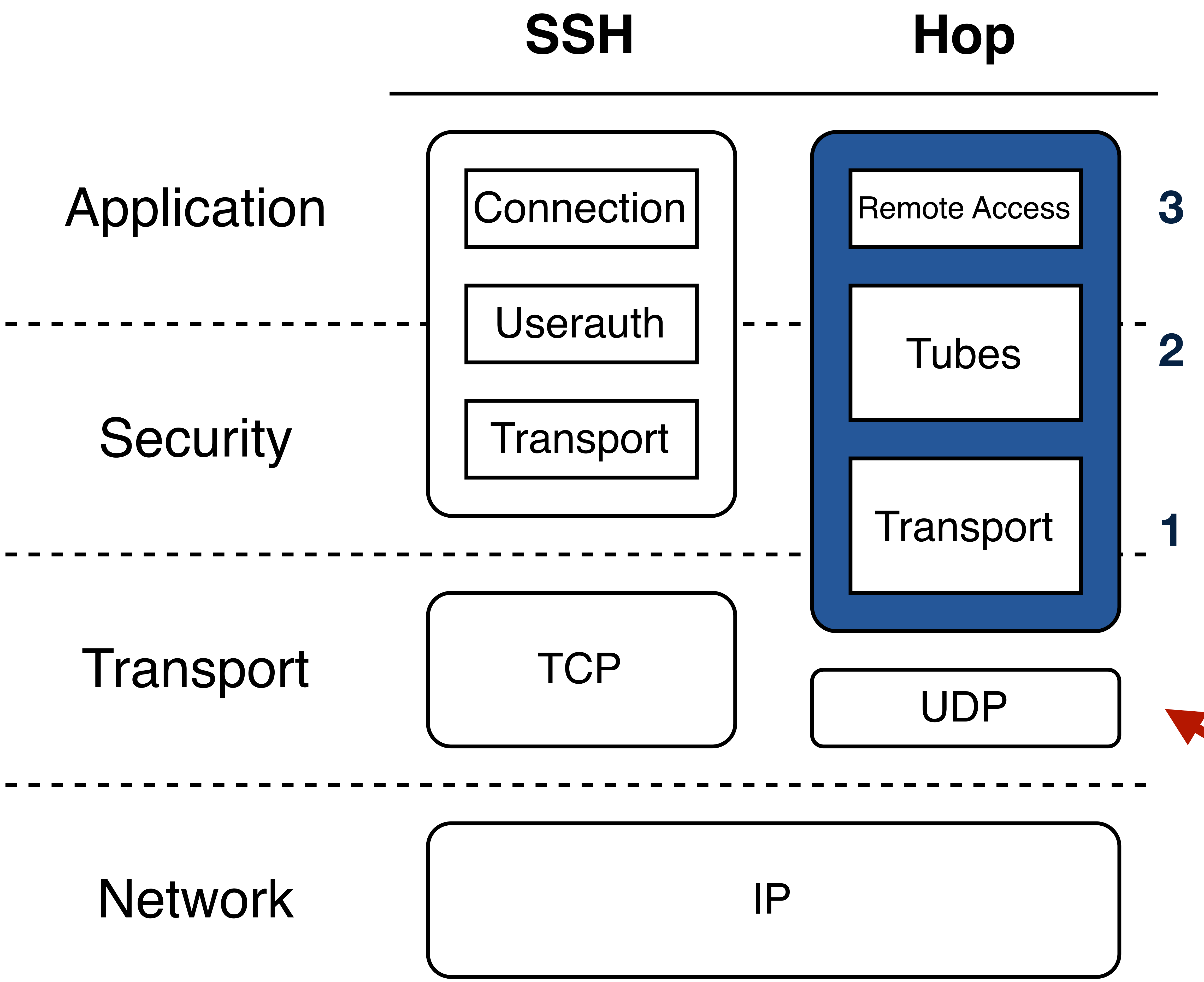
3 - Privacy and Confidentiality

2 - Simple Cryptographic Protocol

1 - Secure Transport for Unreliable Traffic

Three Inner Sub-Protocols

Protocol Overview



Protocol Requirements

8 - Secure Credential Delegation

7 - Constrained Environment Support

6 - Extensible Client Verification

5 - Trustworthy Host Identification

4 - Post-Quantum Security

3 - Privacy and Confidentiality

2 - Simple Cryptographic Protocol

1 - Secure Transport for Unreliable Traffic

Req. 1 - Secure Transport for Unreliable Traffic

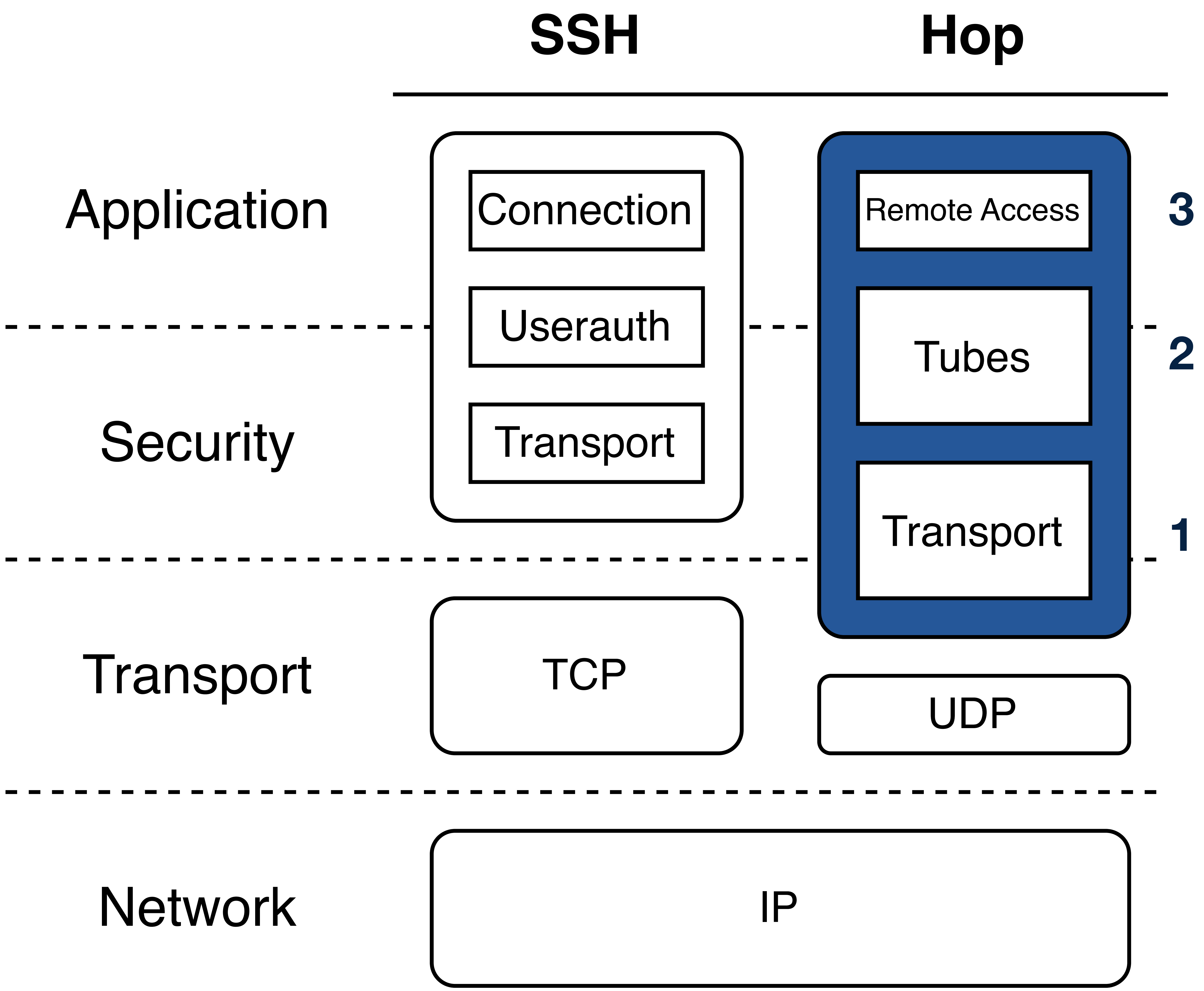
Motivation

UDP vs TCP

- ❌ Three-way TCP handshake
- ❌ Port scanning
- ❌ TCP over TCP slowdown
- ✅ Roaming
- ✅ Intermittent connectivity
- ✅ Fast session resumption
- ✅ Ideal for transmission of small amounts of data at a time (IoT)
- ✅ Tunneling of UDP-based protocols
- ✅ Enable native support of UDP-based applications (e.g., Mosh)

Three Inner Sub-Protocols

Protocol Overview



Protocol Requirements

8 - Secure Credential Delegation

7 - Constrained Environment Support

6 - Extensible Client Verification

5 - Trustworthy Host Identification

4 - Post-Quantum Security

3 - Privacy and Confidentiality

2 - Simple Cryptographic Protocol

1 - Secure Transport for Unreliable Traffic

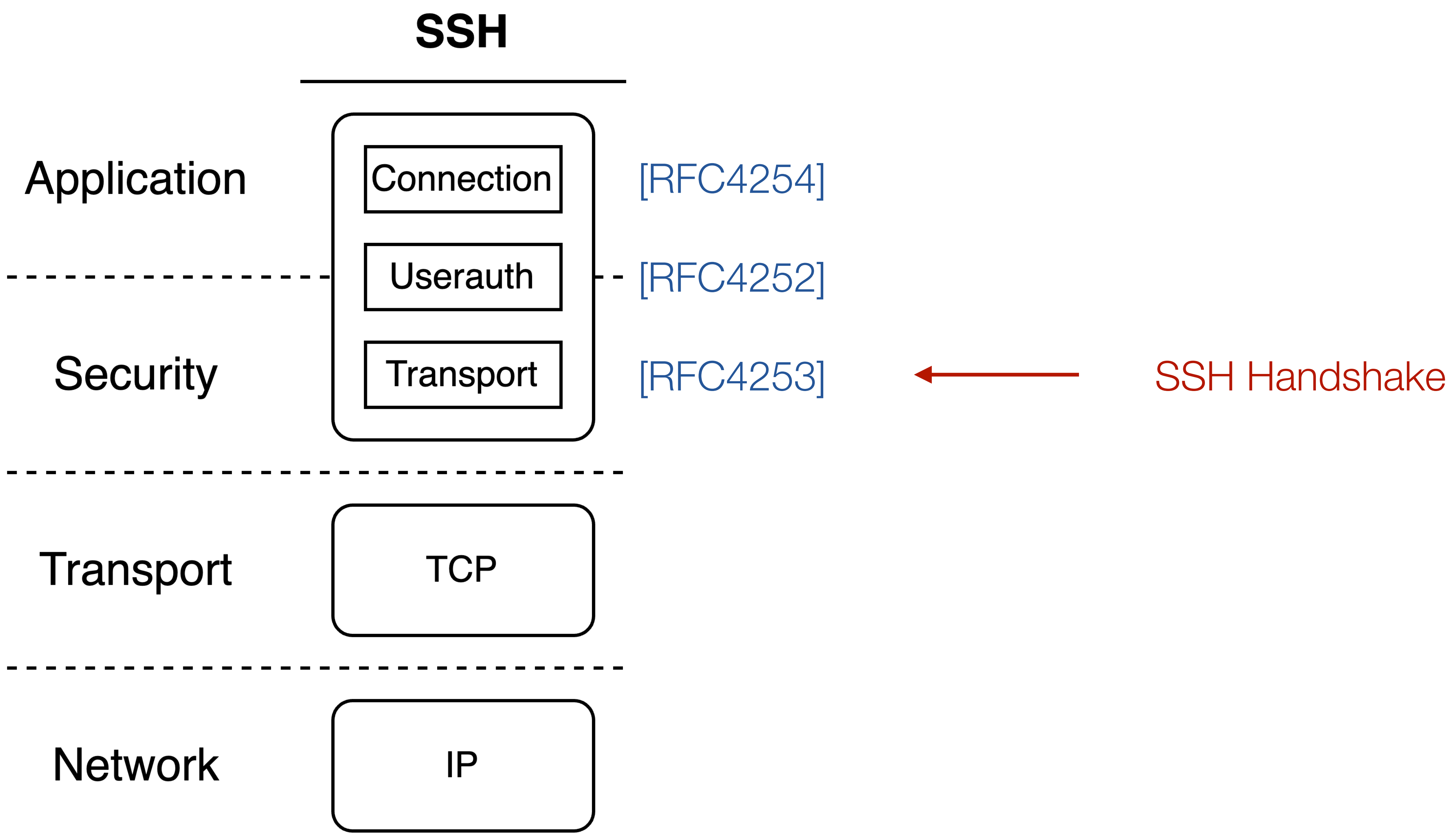
Req. 2 - Simple Cryptographic Protocol

Motivation

Terrapin Attack: Breaking SSH Channel Integrity By Sequence Number Manipulation

Fabian Bäumer, Marcus Brinkmann, and Jörg Schwenk, *Ruhr University Bochum*

https://www.usenix.org/conference/usenixsecurity24/presentation/bäumer



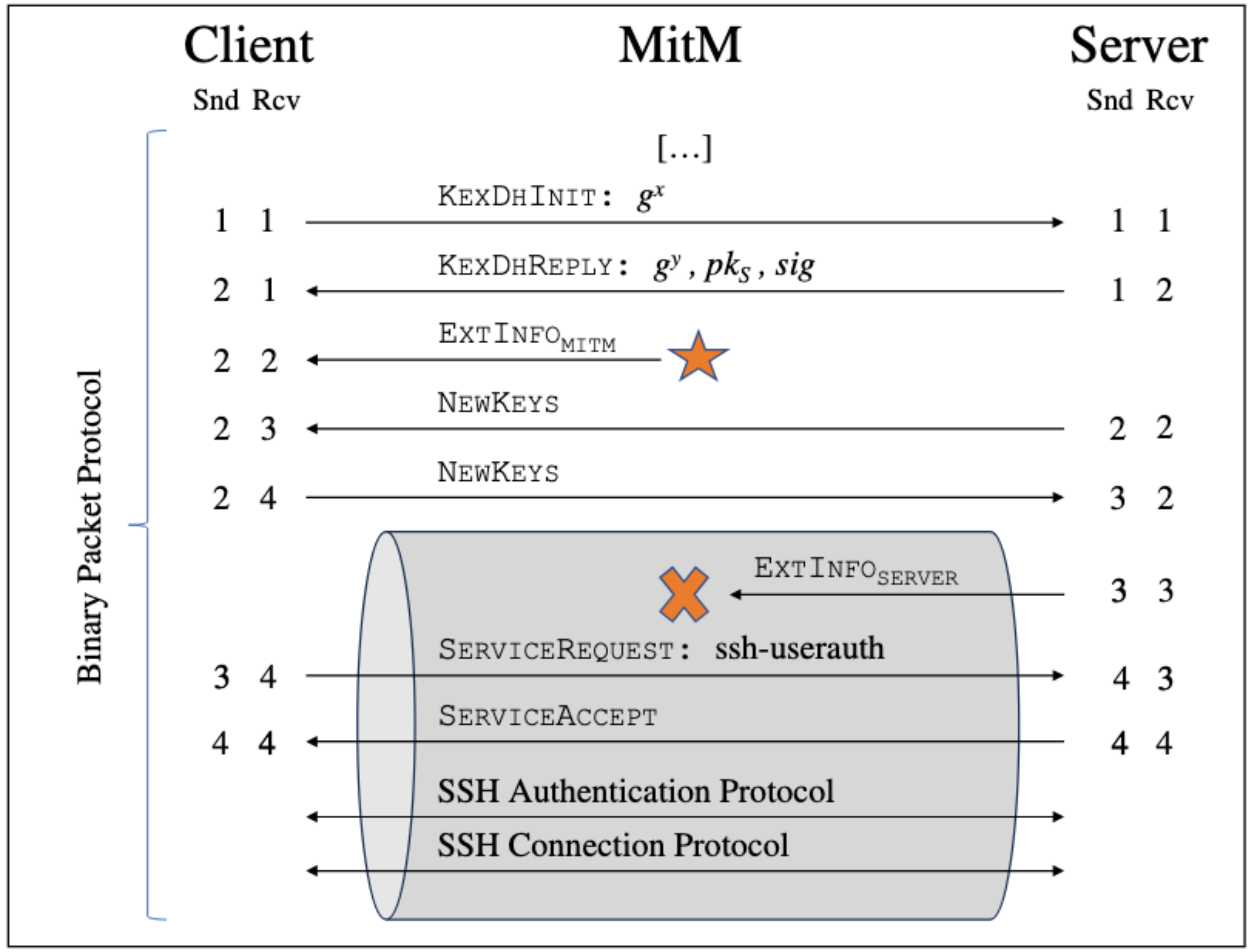


Figure 6: Rogue Extension Negotiation Attack on AsyncSSH: The MitM injects a malicious extension information message before the key exchange completes and deletes the server's ExtInfo message to account for the change in sequence numbers. This attack relates to the generic extension downgrade attack in Section 5.2.

Req. 3 - Privacy and Confidentiality

Motivation

A deeper understanding of SSH: Results from Internet-wide scans

Oliver Gasser, Ralph Holz, Georg Carle
Technische Universität München
Faculty of Informatics
Chair for Network Architectures and Services
Email: {gasser,holz,carle}@net.in.tum.de

Catch-22: Uncovering Compromised Hosts using SSH Public Keys

Cristian Munteanu
Max Planck Institute for Informatics

Georgios Smaragdakis
Delft University of Technology

Anja Feldmann
Max Planck Institute for Informatics

Tobias Fiebig
Max Planck Institute for Informatics

Findings of potential vulnerabilities:

- Old protocol versions
- Weak keys
- Small keys
- Duplicated keys
- Weak cryptography

With only 52 public keys, 3 usernames, ports 22 and 2222

→ 21700 compromised servers

Req. 4 - Post-Quantum Security

Post-quantum Cryptographic Analysis of SSH

Benjamin Benčina
Royal Holloway, University of London, UK
Email: benjamin.bencina.2022@live.rhul.ac.uk

Benjamin Dowling
King's College London, UK
Email: benjamin.dowling@kcl.ac.uk

Varun Maram
SandboxAQ, UK
Email: varun.maram@sandboxaq.com

Keita Xagawa
Technology Innovation Institute, UAE
Email: keita.xagawa@tii.ae

Post-quantum WireGuard

September 25, 2023

Andreas Hülsing
Eindhoven University of Technology
The Netherlands
andreas@huelsing.net

Kai-Chun Ning
KPN B.V.
The Netherlands
kaichun.ning@kpn.com

Peter Schwabe
Max Planck Institute for Security and Privacy, Germany &
Radboud University, The Netherlands
peter@cryptojedi.org

Fiona Johanna Weber
Eindhoven University of Technology
The Netherlands
crypto@fionajw.de

Philip R. Zimmermann
Delft University of Technology & KPN B.V.
The Netherlands
prz@mit.edu

A Comprehensive Survey on Post-Quantum TLS

Nouri Alnahawi[2], Johannes Müller[1,3,4], Jan Oupický[1] and Alexander Wiesmaier[2]

[1] University of Luxembourg, Esch-sur-Alzette, Luxembourg
[2] Darmstadt University of Applied Sciences, Darmstadt, Germany
[3] LORIA, Nancy, France
[4] CNRS, Paris, France

QUIC Protocol with Post-Quantum Authentication

Manohar Raavi, Simeon Wuthier, Pranav Chandramouli, Xiaobo Zhou, and Sang-Yoon Chang

University of Colorado, Colorado Springs, USA
Department of Computer Science
{mraavi,swuthier,pchandra,xzhou,schang2}@uccs.edu

+

FIPS 203

Federal Information Processing Standards Publication

Module-Lattice-Based Key-Encapsulation Mechanism Standard

Category: Computer Security **Subcategory: Cryptography**

Natural Institute of Standards and Technology

Building a Handshake

Hop Transport

✓ Low network round-trips

✓ No cryptographic agility or sequences numbers

✓ Not discoverable to scanners

✓ Post-Quantum secure (forward secrecy)

Noise Protocol Framework

- Describes a series of handshake patterns
- To create secure communication protocols
- Based on Diffie-Hellman key exchange

PQNoise

- Post-Quantum adaptation of Noise
- Replaces DH by NIST standardization of ML-KEM

Hop PQNoise Adaptation

Hop Transport

PQNoise IK

Out of band Static ML-KEM key

<- skem

-> Encaps(skem), ekem, skem

<- Encaps(ekem), Encaps(skem)

Hop

<- skem

-> Encaps(skem), ekem, s

<- Encaps(ekem), s, DH(ss)

e = ephemeral

s = static

Client

Server

Why?

- Diffie-Hellman keys 32bytes vs ~ 800bytes
- We don't *require* PQ authentication (NIST)

Hop Noise Extension

Hop Transport

Hop

Noise XX

-> e

<- e, DH(ee), s, DH(es)

-> s, DH(se)

- Mutual authentication
- Static public key transmission

-> ekem

<- Encaps(ekem), cookie

-> e, ekem, cookie

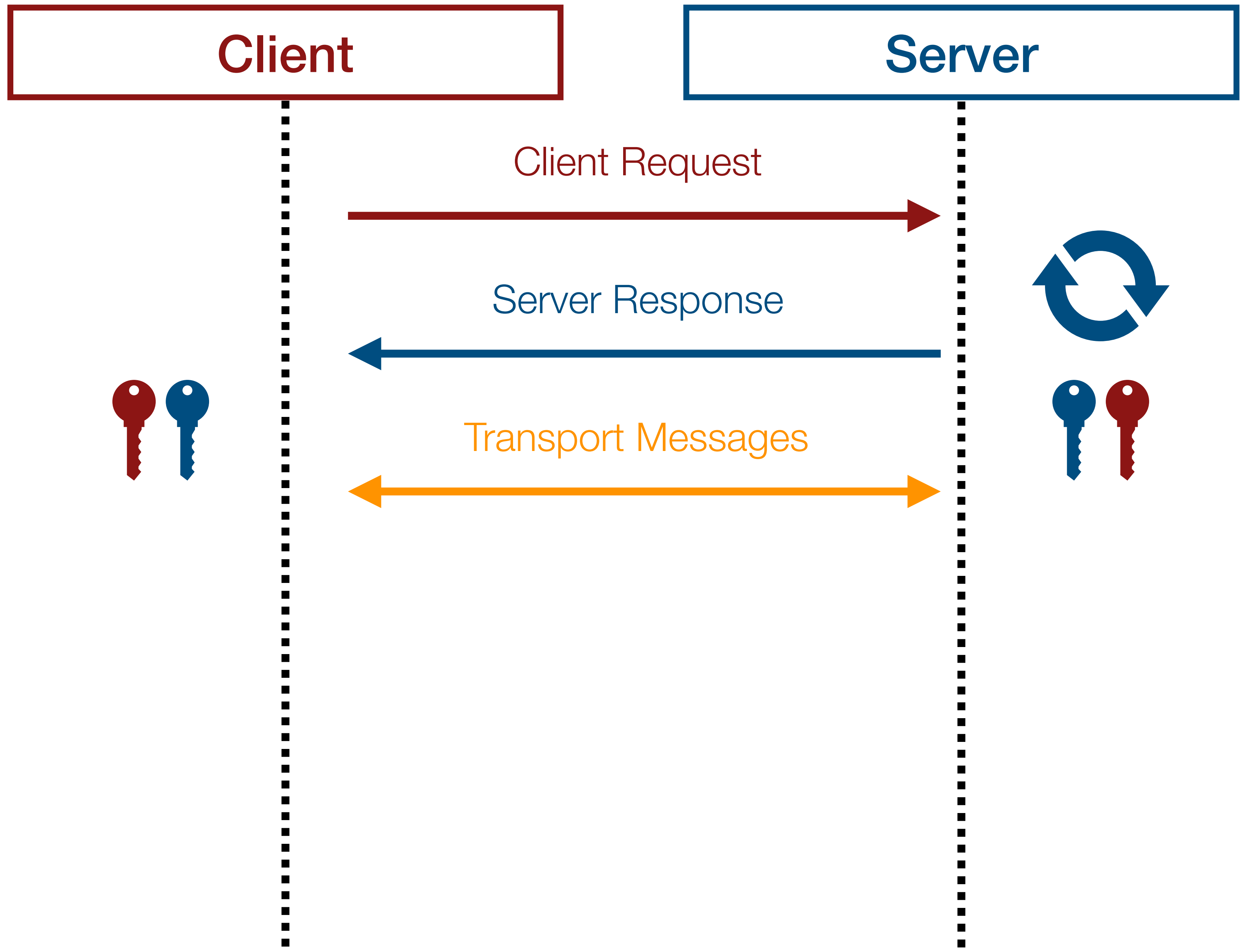
<- e, DH(ee), s, DH(es)

-> s, DH(se)

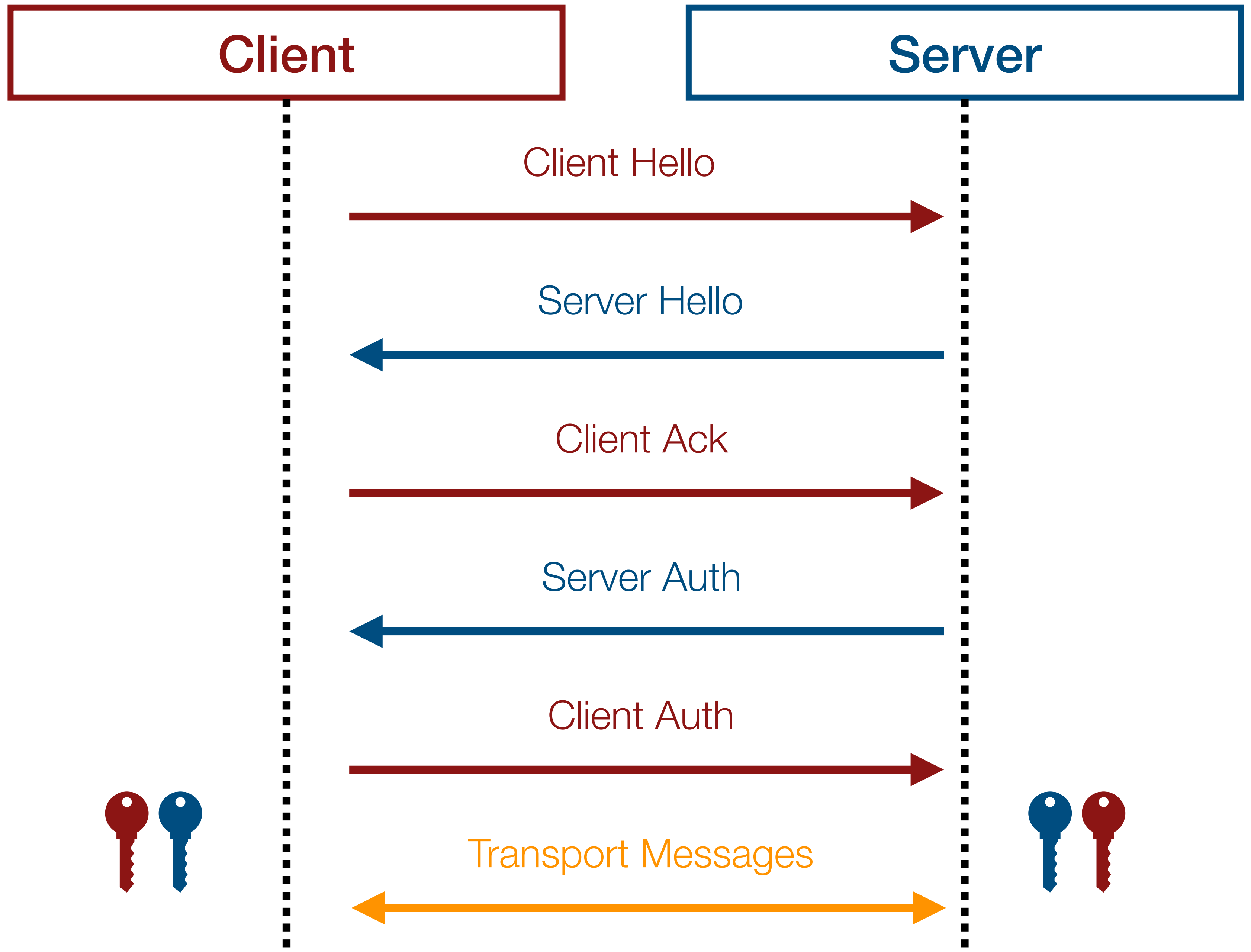
Cookie: To prevent denial of service amplifier (Req. 1)

Hop Discoverable and Hidden Modes

Hop Transport



Hop Hidden

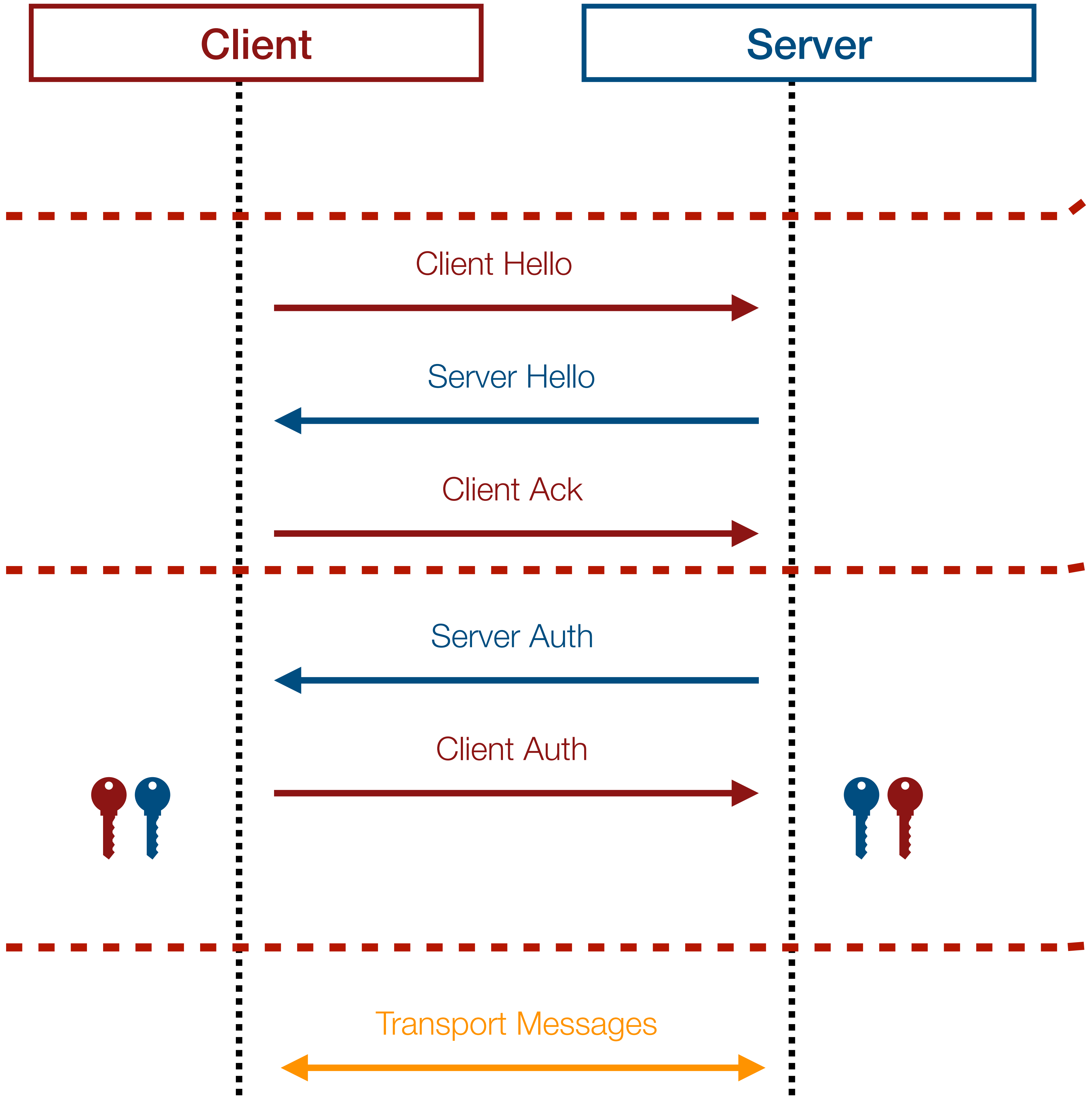


Hop Discoverable


= derivation of the final symmetric keys

PQ Handshake Comparison

Hop Transport



0 vs 4

3 vs 8

2 vs 4

Hop Discoverable

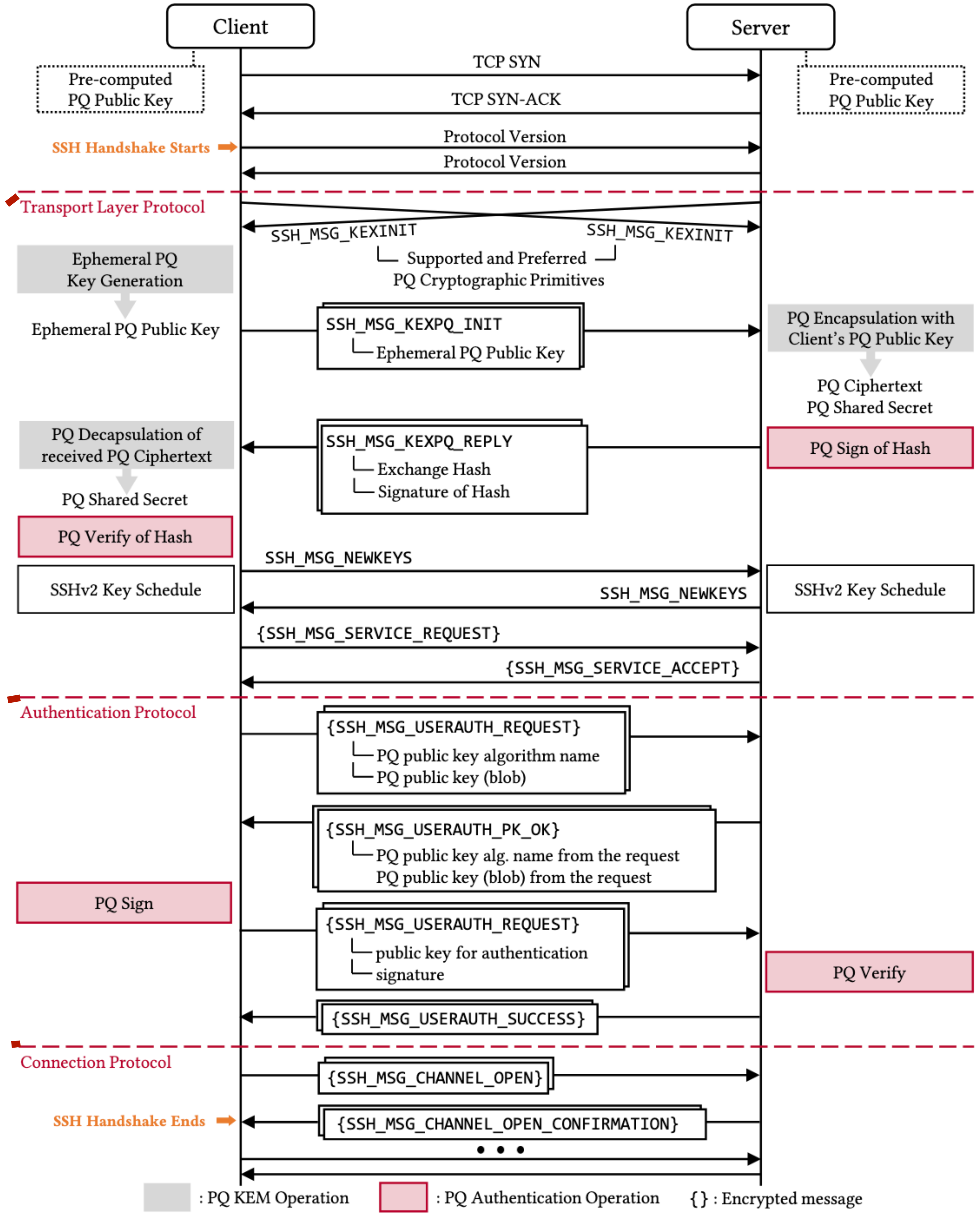
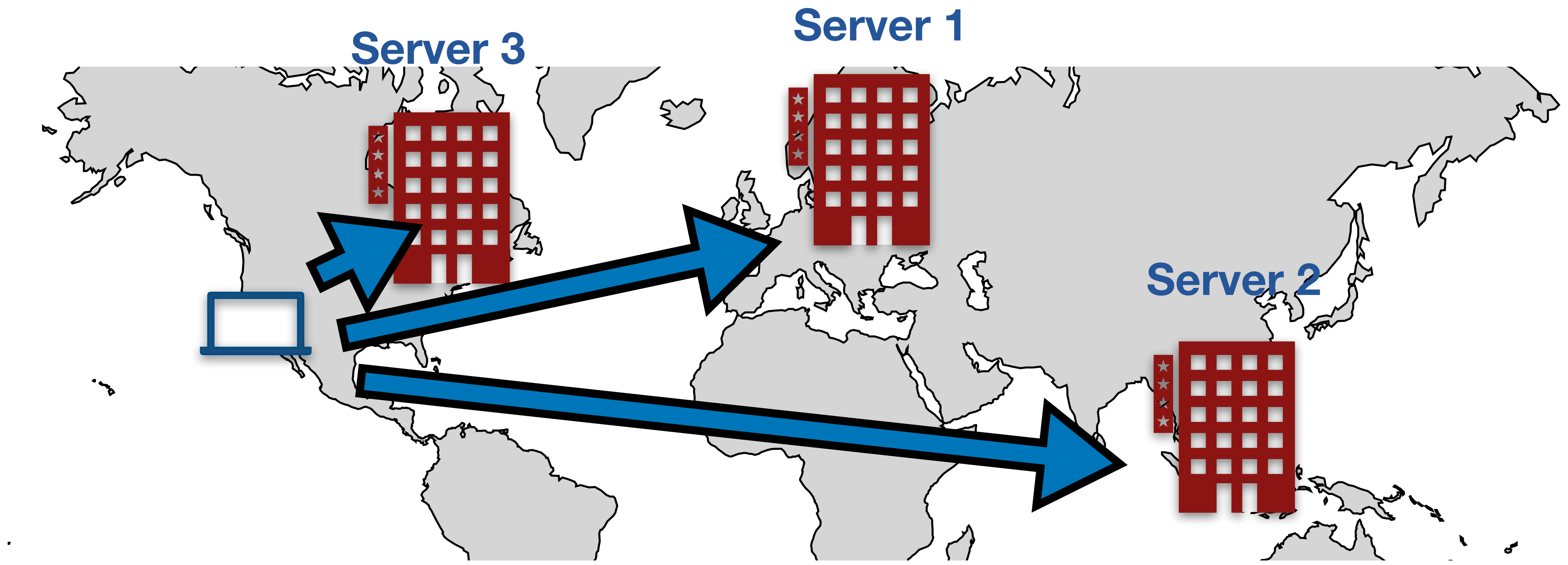


Figure 2: Post-Quantum SSH Handshake Overview

Assessing the Overhead of Post-Quantum Cryptography in TLS 1.3 and SSH

Session Establishment

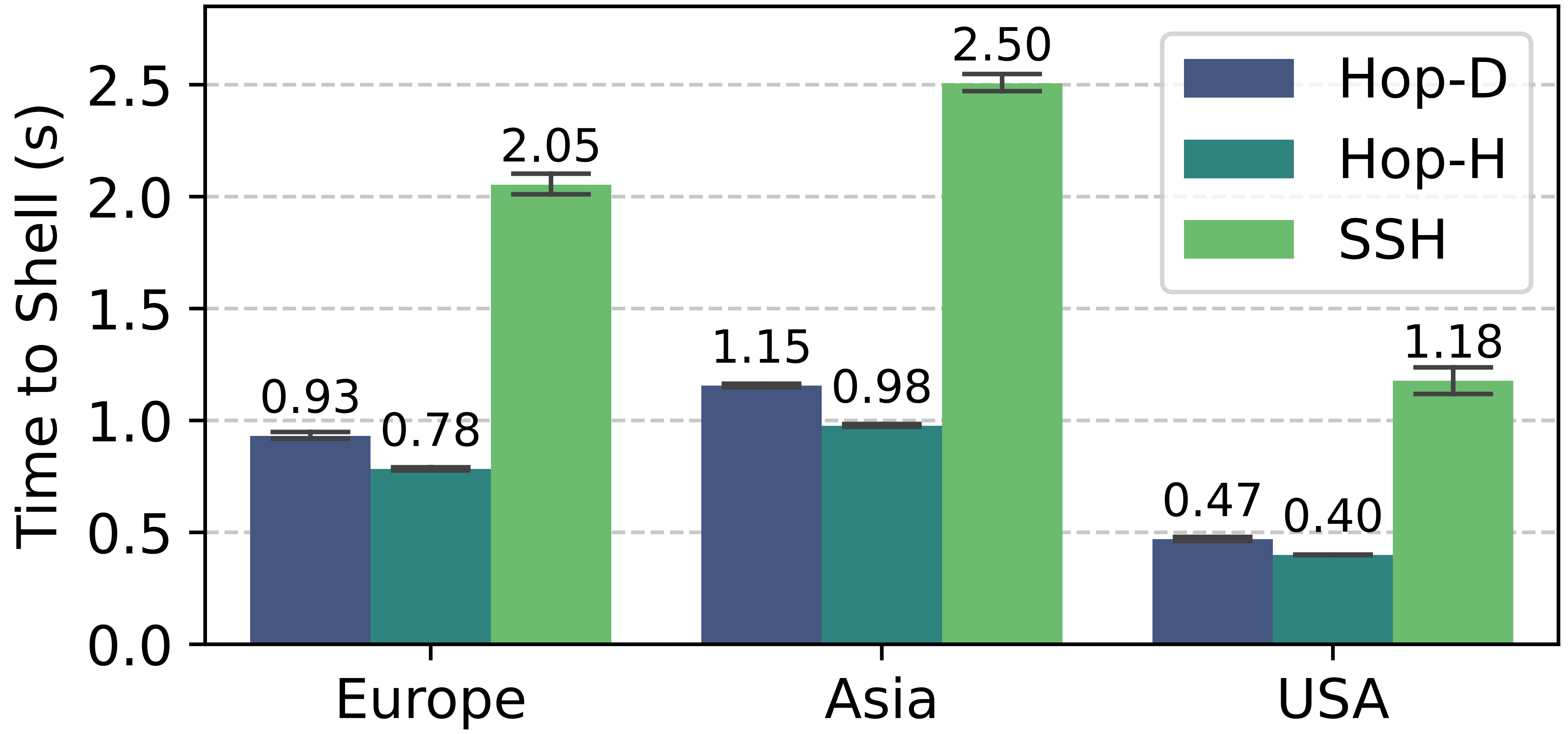
Evaluation



- Time to establish a new session and execute a command

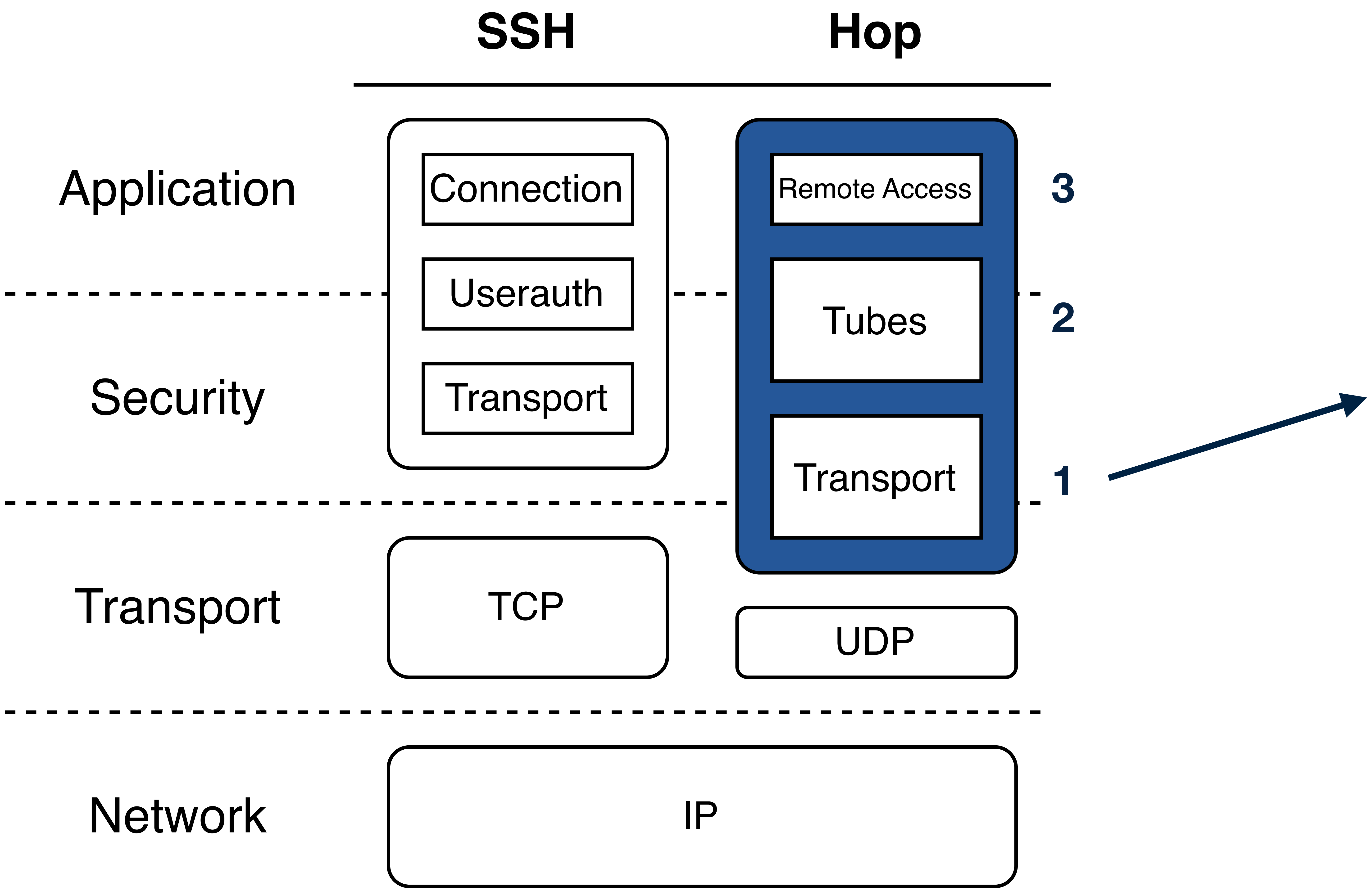
- Round-trips:
  - 5 Hop Hidden,
  - 6 Hop Discoverable,
  - 12 SSH (No PQ)

- Significant improvement due to Hop's handshake



Three Inner Sub-Protocols

Protocol Overview



Protocol Requirements

8 - Secure Credential Delegation

7 - Constrained Environment Support

6 - Extensible Client Verification

5 - Trustworthy Host Identification

4 - Post-Quantum Security

3 - Privacy and Confidentiality

2 - Simple Cryptographic Protocol

1 - Secure Transport for Unreliable Traffic

Req. 5 - Trustworthy Host Identification

Motivation

Do Users Verify SSH Keys?

PETER GUTMANN

No →

SSH Key Management Challenges and Requirements

Tatu Ylonen
University of Helsinki and SSH Communications Security
ylo@ssh.com

```
➜  ~ ssh root@[redacted].compute.amazonaws.com -p 32774
The authenticity of host '[[redacted].compute.amazonaws.com]:[redacted]
0]:32774)' can't be established.
ED25519 key fingerprint is: SHA256:e9yjdPTWoJtIiBTx43wOwcPEvy[redacted]
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

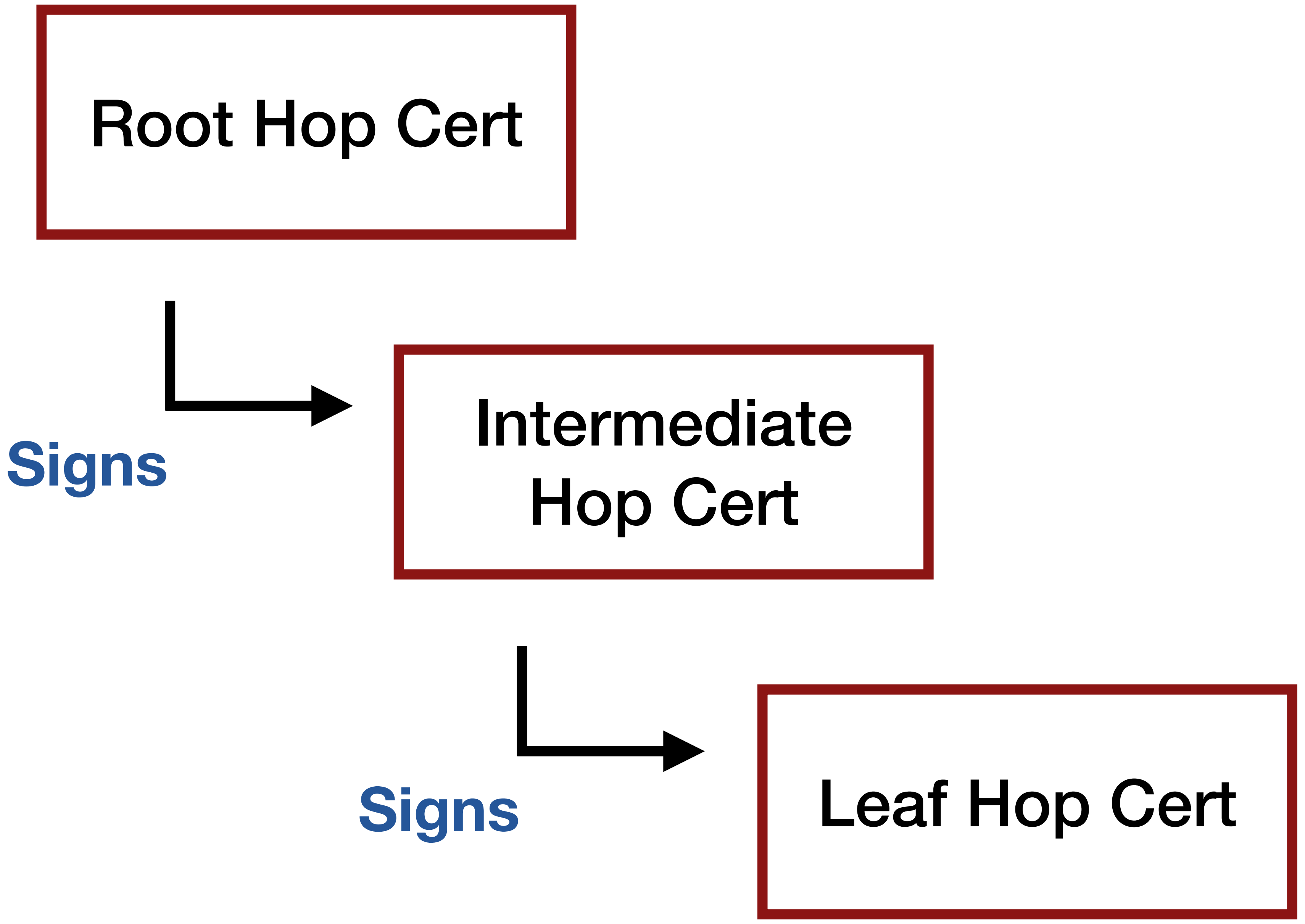
"Users do not understand the warnings about changed host keys and even for experts, verifying the keys is too cumbersome to do reliably." *Tatu Ylönen*

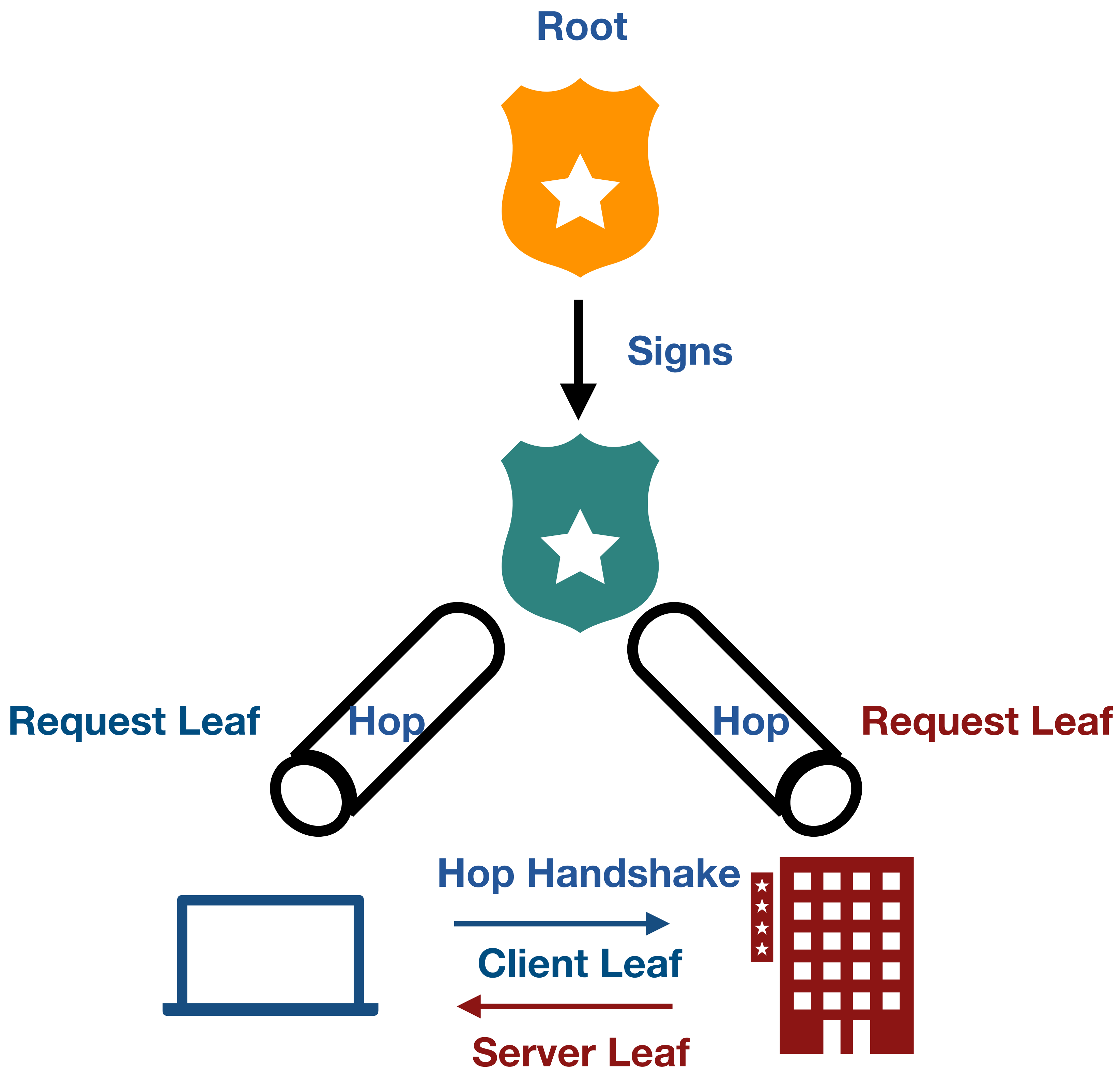
➡ There is a need in having a reliable way of identifying the server

➡ **Certificates with a chain of trust**

Hop Automatic Certificate Management Environment

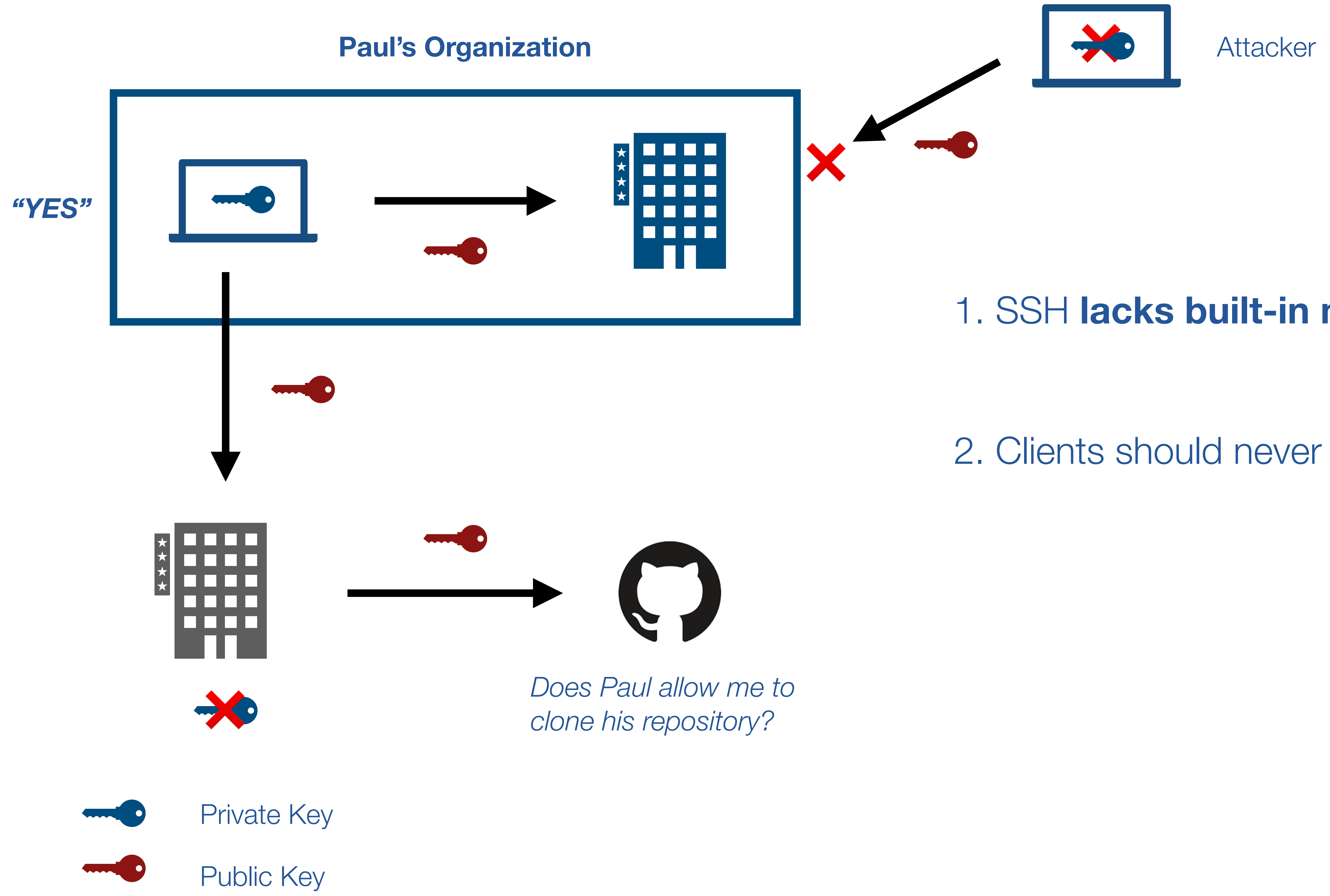
Hop Transport





Req. 6 - Extensible Client Verification

Motivation

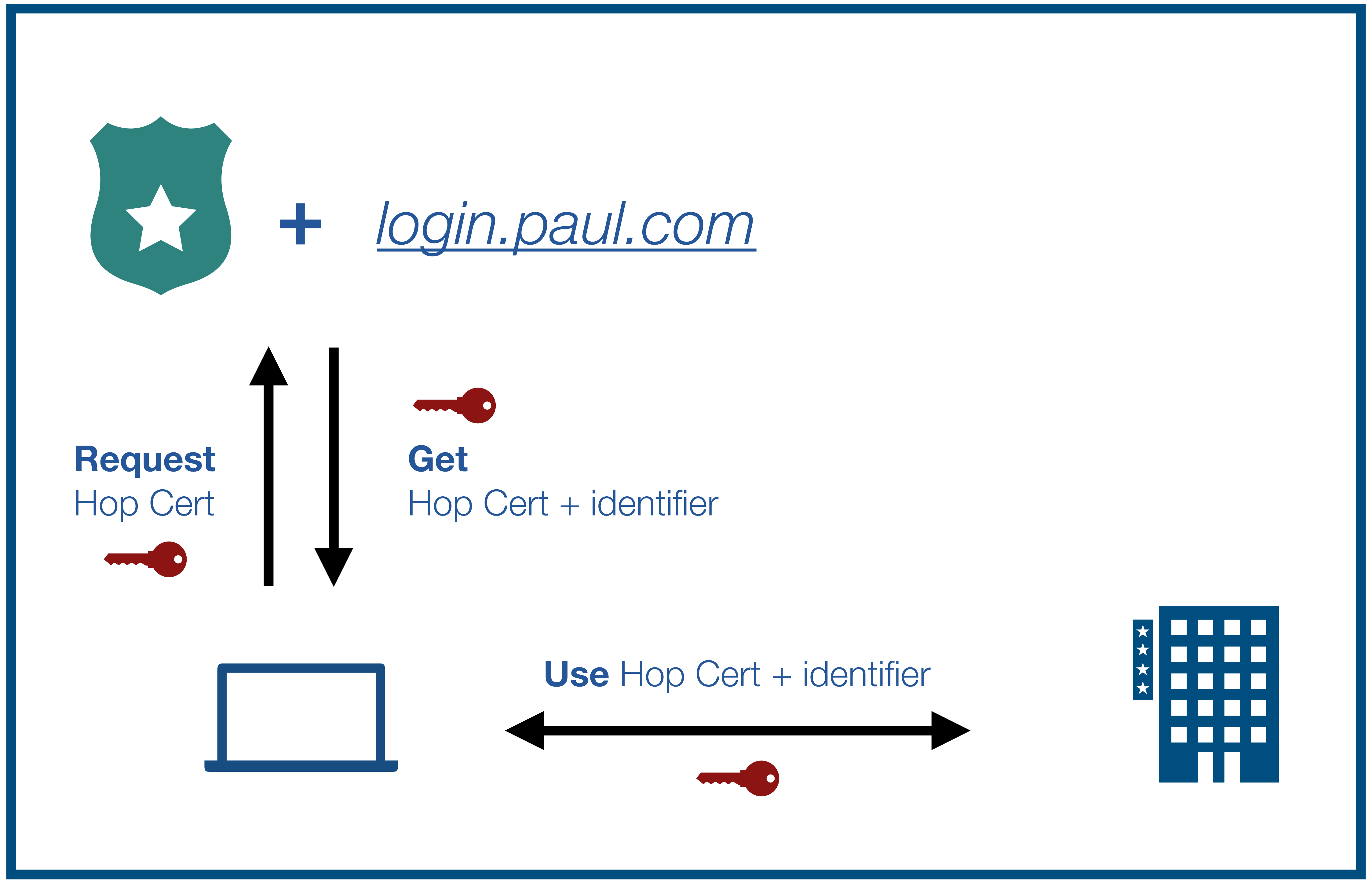


1. SSH **lacks built-in mechanisms** for verifying keys/identity shared out-of-band

2. Clients should never expose private keys, even when delegating actions

Hop Client Identification with Web Login

Transport



Req. 7 - Constrained Environment Support

Motivation

Hop

No ASN.1 or X.509

Z. Durumeric, J. Kasten, M. Bailey, and J. A. Halderman. Analysis of the https certificate ecosystem. In *ACM Internet Measurement Conference*, 2013.

C. Brubaker, S. Jana, B. Ray, S. Khurshid, and V. Shmatikov. Using frankencerts for automated adversarial testing of certificate validation in ssl/tls implementations. In *IEEE Symposium on Security and Privacy*, 2014.

K. Kleine and D. E. Simos. Coveringcerts: Combinatorial methods for x. 509 certificate testing. In *IEEE International conference on software testing, verification and validation (ICST)*, 2017.

Y. Chen and Z. Su. Guided differential testing of certificate validation in ssl/tls implementations. In *10th Joint Meeting on Foundations of Software Engineering*, 2015.

H. Sardeshmukh and D. Ambawade. A DTLS based lightweight authentication scheme using symmetric keys for Internet of Things. In *International Conference on Wireless Communications, Signal Processing and Networking*, 2017.

- Flexibility
- Inherent complexity

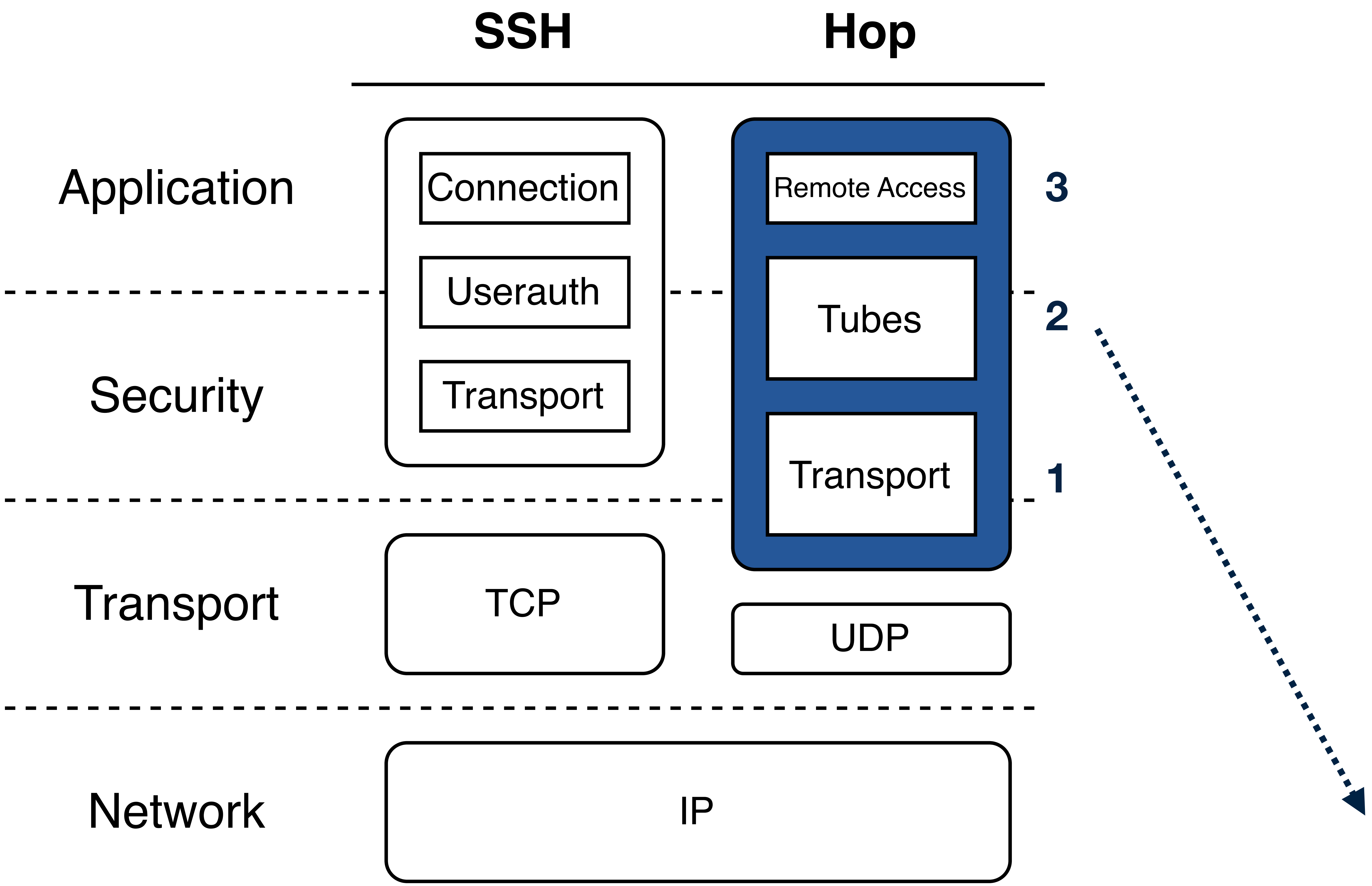
C. Tian, C. Chen, Z. Duan, and L. Zhao. Differential testing of certificate validation in SSL/TLS implementations: an rfc-guided approach. *ACM Transactions on Software Engineering and Methodology*, 2019.

D. Kumar, Z. Wang, M. Hyder, J. Dickinson, G. Beck, D. Adrian, J. Mason, Z. Durumeric, J. A. Halderman, and M. Bailey. Tracking certificate misissuance in the wild. In *IEEE Symposium on Security and Privacy*, 2018.

Field	Size (bytes)
Certificate Protocol Version	1
Certificate Type	1
Reserved	2
IssuedAt	8
ExpiresAt	8
Public Static Identity Key	32
Parent Certificate Fingerprint	32
ID Chunk Size	2
ID Chunk	4-512
ID Block	4-256
ID Block Size	1
ID Type	1
ID Label Size	1
ID Label	1..253
Parent Signature	64

Three Inner Sub-Protocols

Protocol Overview



Protocol Requirements

8 - Secure Credential Delegation

7 - Constrained Environment Support

6 - Extensible Client Verification

5 - Trustworthy Host Identification

4 - Post-Quantum Security

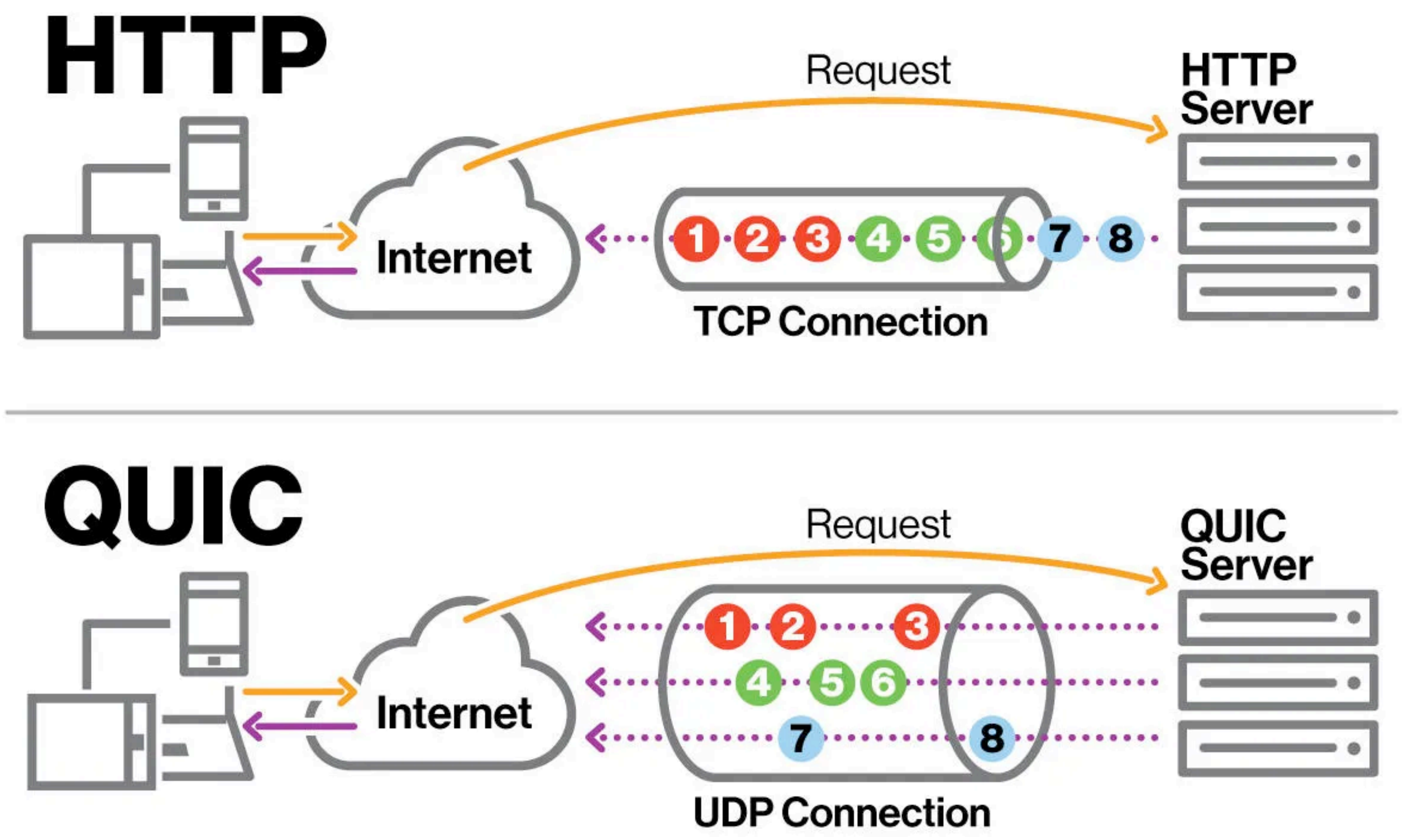
3 - Privacy and Confidentiality

2 - Simple Cryptographic Protocol

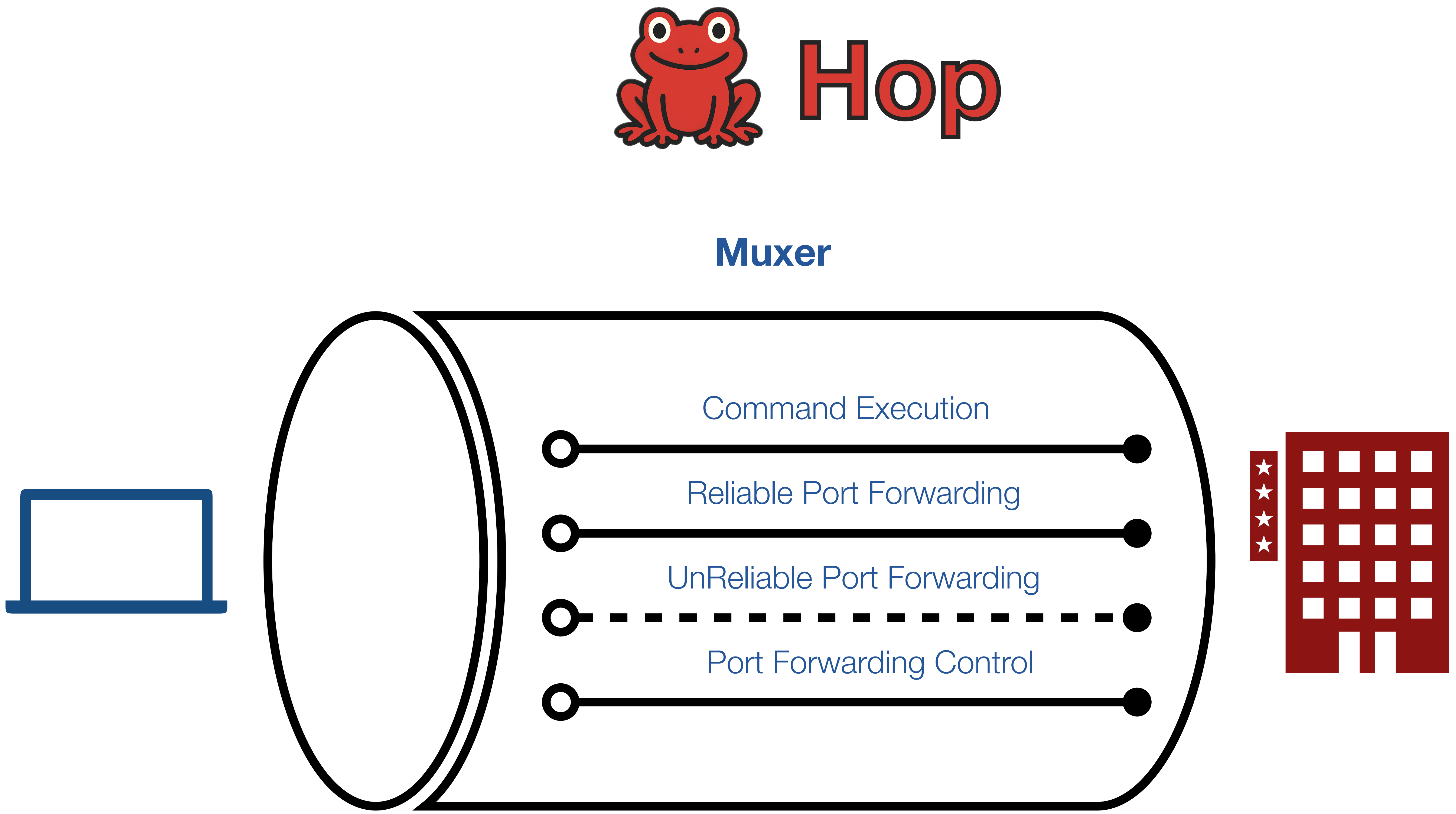
1 - Secure Transport for Unreliable Traffic

Tube Abstraction

Hop Tubes

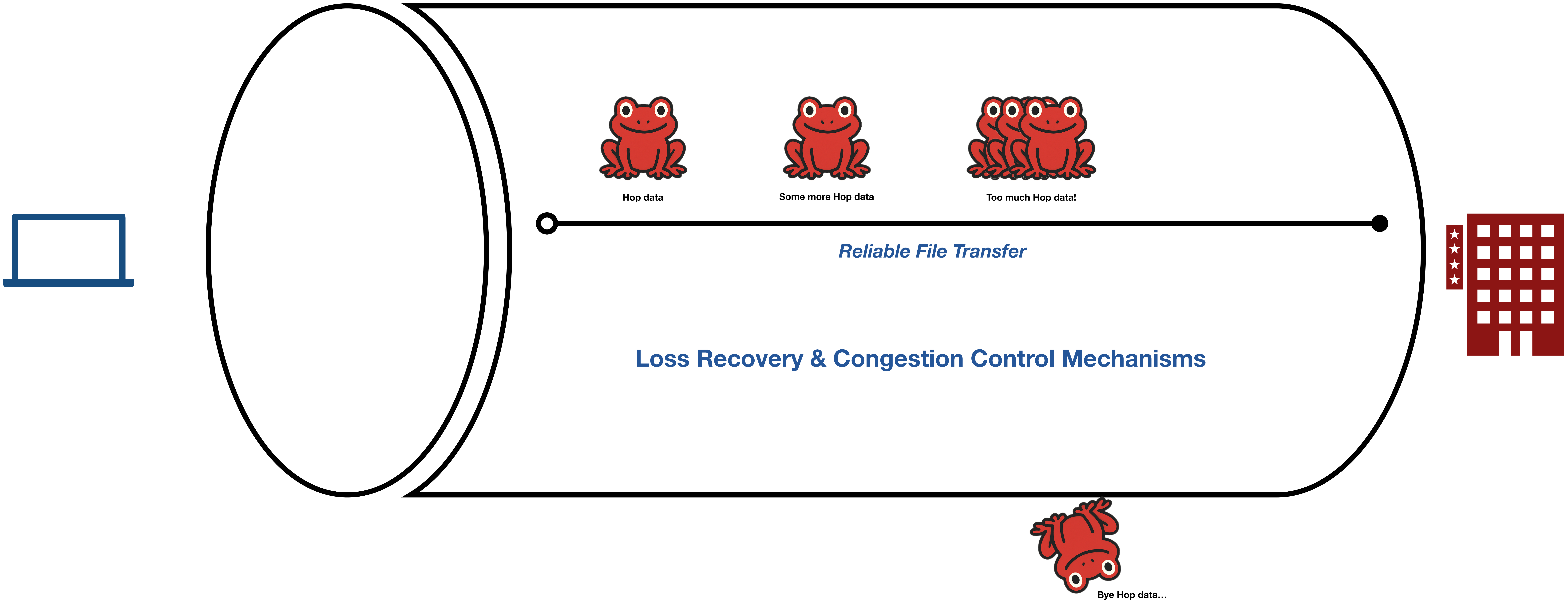


How QUIC speeds up all web applications - Medium Post By Frank Orozco



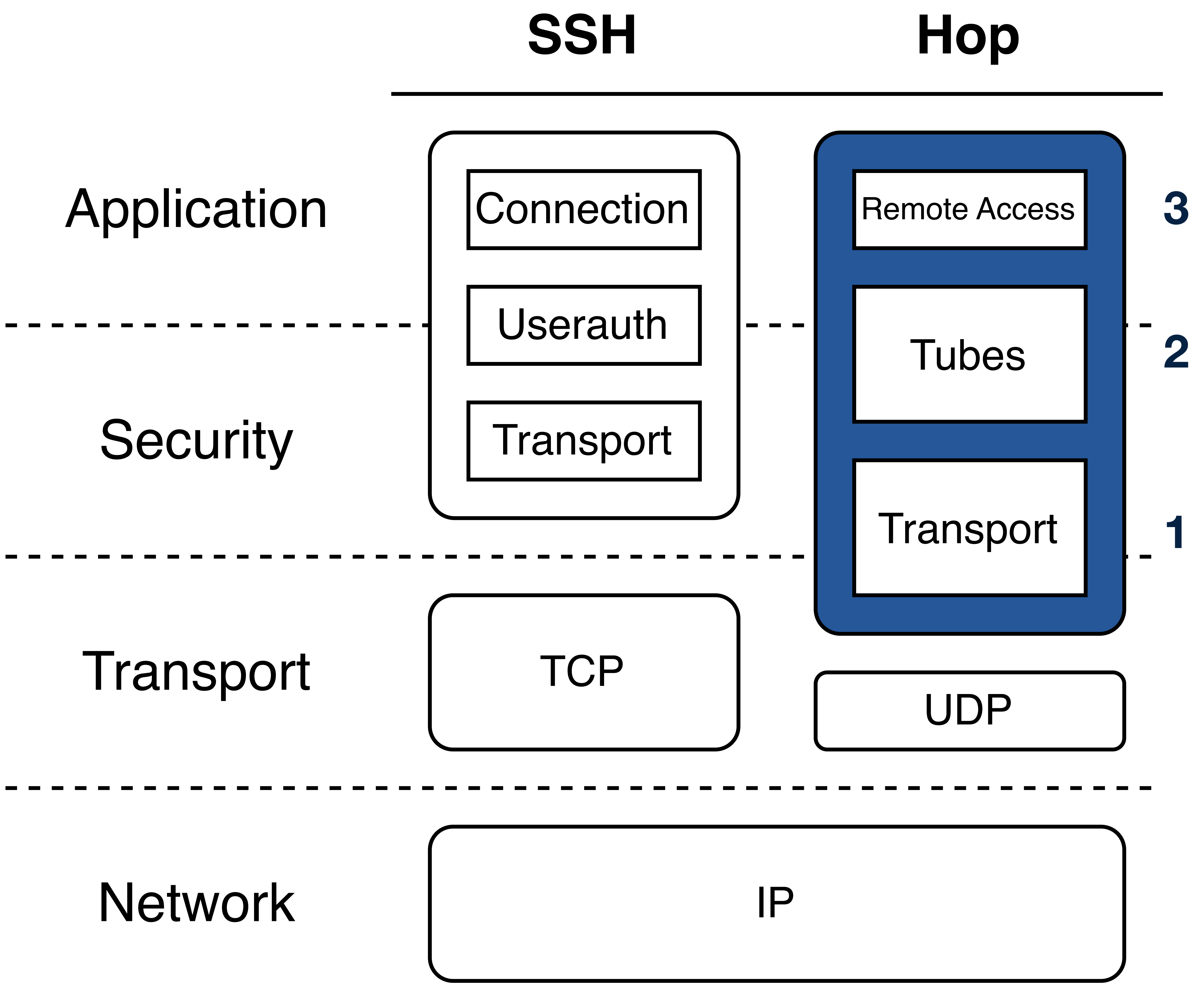
Loss Recovery and Congestion Control

Hop Tubes



Three Inner Sub-Protocols

Protocol Overview



Protocol Requirements

8 - Secure Credential Delegation

7 - Constrained Environment Support

6 - Extensible Client Verification

5 - Trustworthy Host Identification

4 - Post-Quantum Security

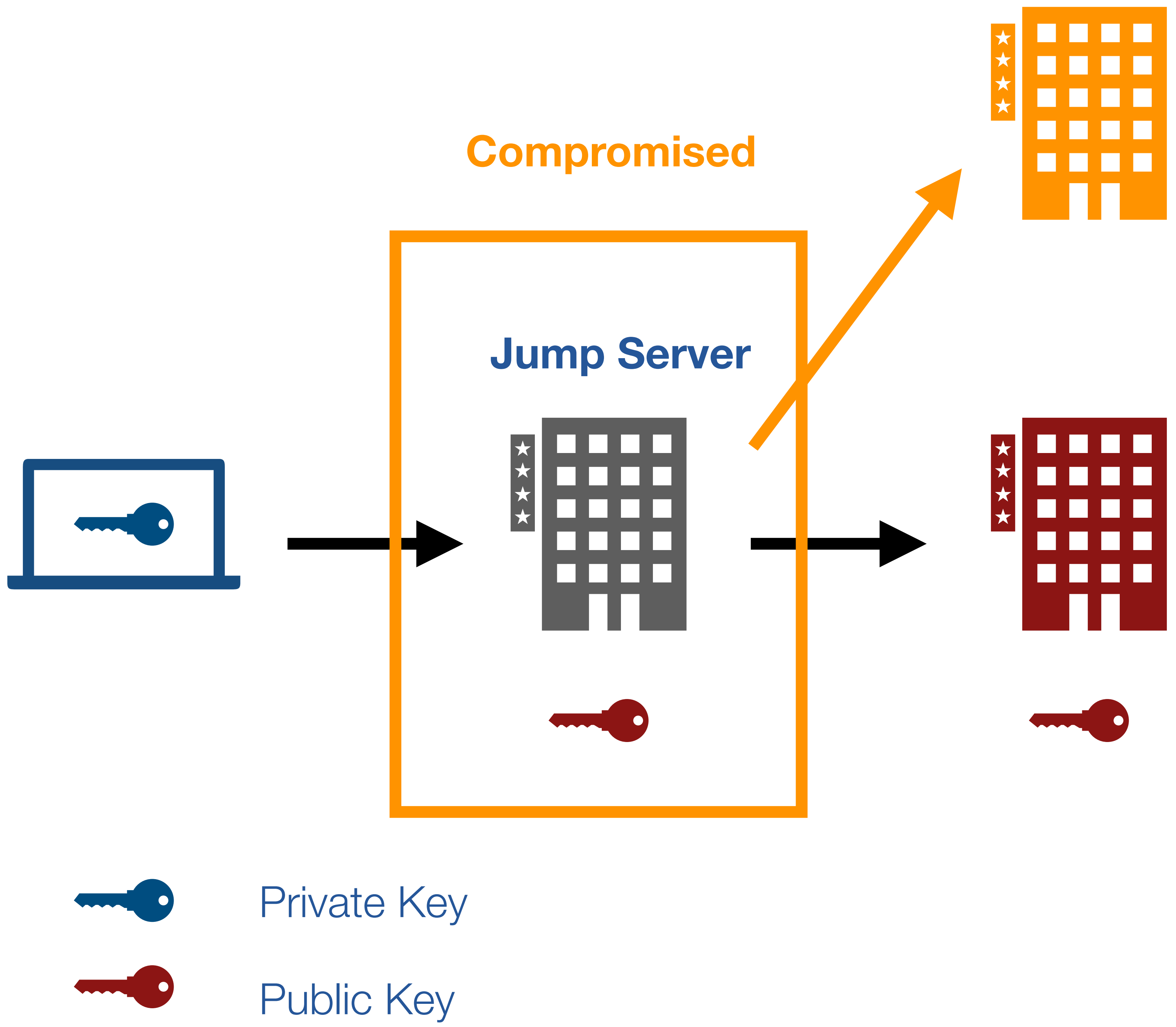
3 - Privacy and Confidentiality

2 - Simple Cryptographic Protocol

1 - Secure Transport for Unreliable Traffic

Req. 8 - Secure Credential Delegation

Motivation



The Case For Secure Delegation

Dmitry Kogan, Henri Stern, Ashley Tolbert, David Mazières, and Keith Winstein
Stanford University

Figure 1: ssh-agent forwarding vs. Guardian Agent

(a) Current ssh-agent forwarding: when granting permission, the user doesn't know the identity of the delegate, the commands the delegate will run, or the server it will run them on.

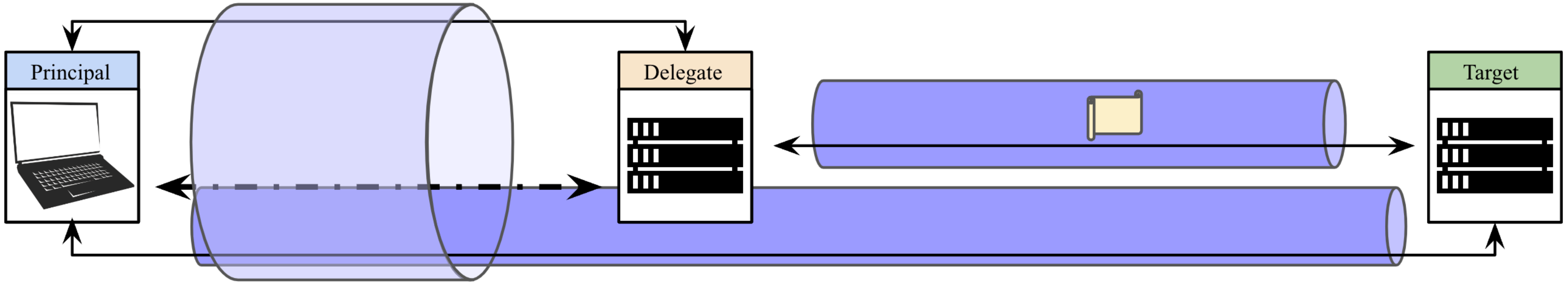
(b) With Guardian Agent, the user has explicit control over the **who**, **what**, and **to whom** of the delegated authority, and can approve each execution individually (the **when**). The system works with existing OpenSSH servers.

Delegation & AuthGrant

Hop Remote Access

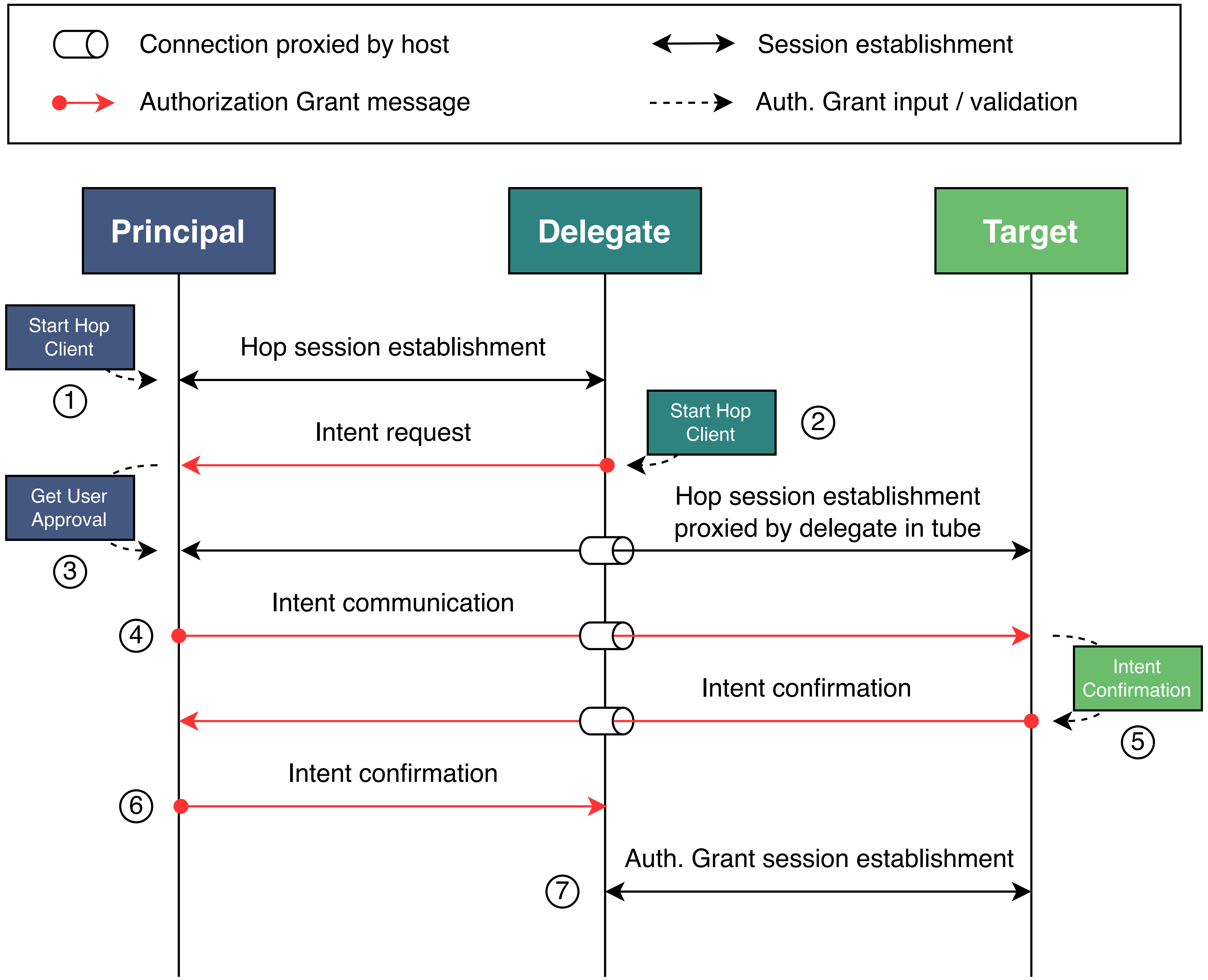
Who,
what,
to whom,
and when?

```
Would you like to
| allow bob.cloud.com
| to run the command 'sudo reboot'
| as alice
| on private.server.com
| from Wed Apr  9 16:03:28 EDT 2025
| until Wed Apr  9 16:04:28 EDT 2025?
    Yes
  > No
```



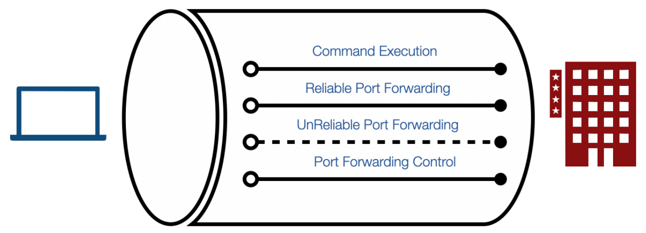
Hop Authorization Grant Protocol

Hop Remote Access

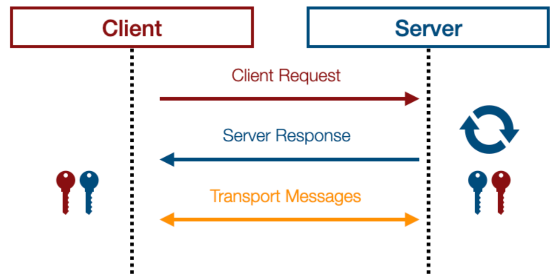


Protocol Requirements

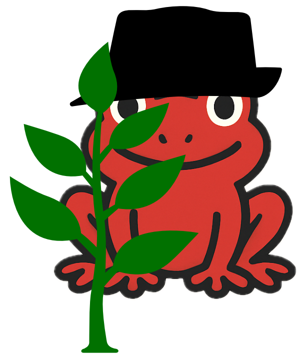
Overview



Secure Transport for Unreliable Traffic



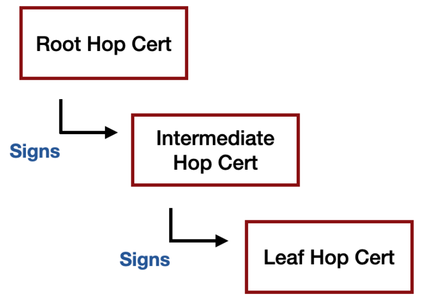
Simple Cryptographic Protocol



Privacy and Confidentiality

ML-KEM

Post-Quantum Security



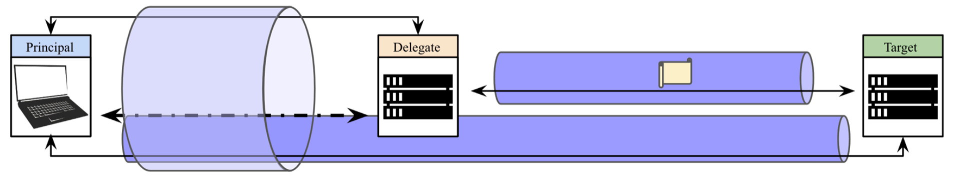
Trustworthy Host Identification

\+ login.paul.com

Extensible Client Verification

Field	Size (bytes)
Certificate Protocol Version	1
Certificate Type	1
Reserved	2
IssuedAt	8
ExpiresAt	8
Public Static Identity Key	32
Parent Certificate Fingerprint	32
ID Chunk Size	2
ID Chunk	4-512
ID Block	4-256
ID Block Size	1
ID Type	1
ID Label Size	1
ID Label	1..253
Parent Signature	64

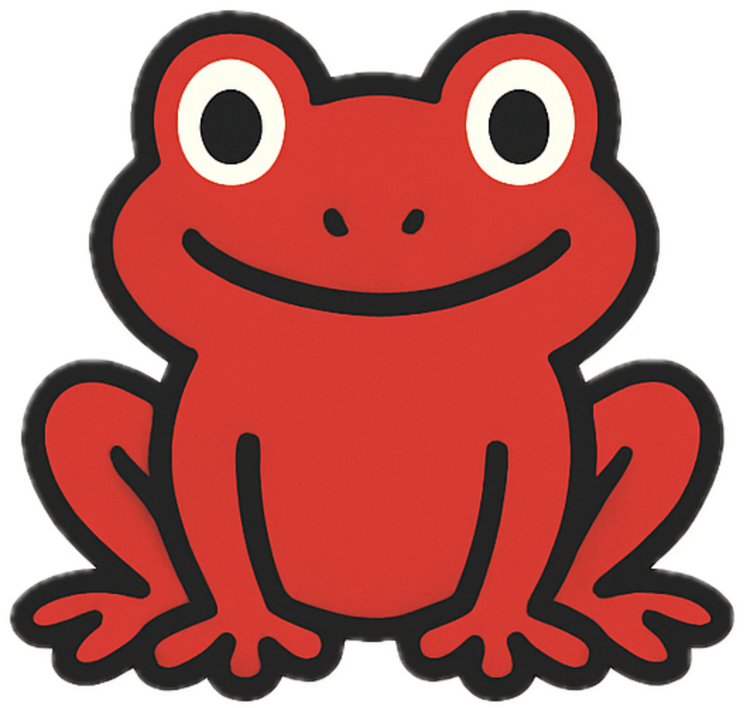
Constrained Environment Support



Secure Credential Delegation

Hop: A Modern Transport and Remote Access Protocol

Takeaways



- We defined 8 design requirements to support today's needs
- We introduce Hop, a three-layer protocol as a secure SSH alternative
- We evaluate Hop's reference implementation under real-world conditions

→ We hope that our work prompts conversation on the future of server remote access

Paul Flammarion

paul.f@uci.edu

Questions?

github.com/hop-proto/hop-go