Zero-Knowledge Middleboxes

Presented by Paul Grubbs

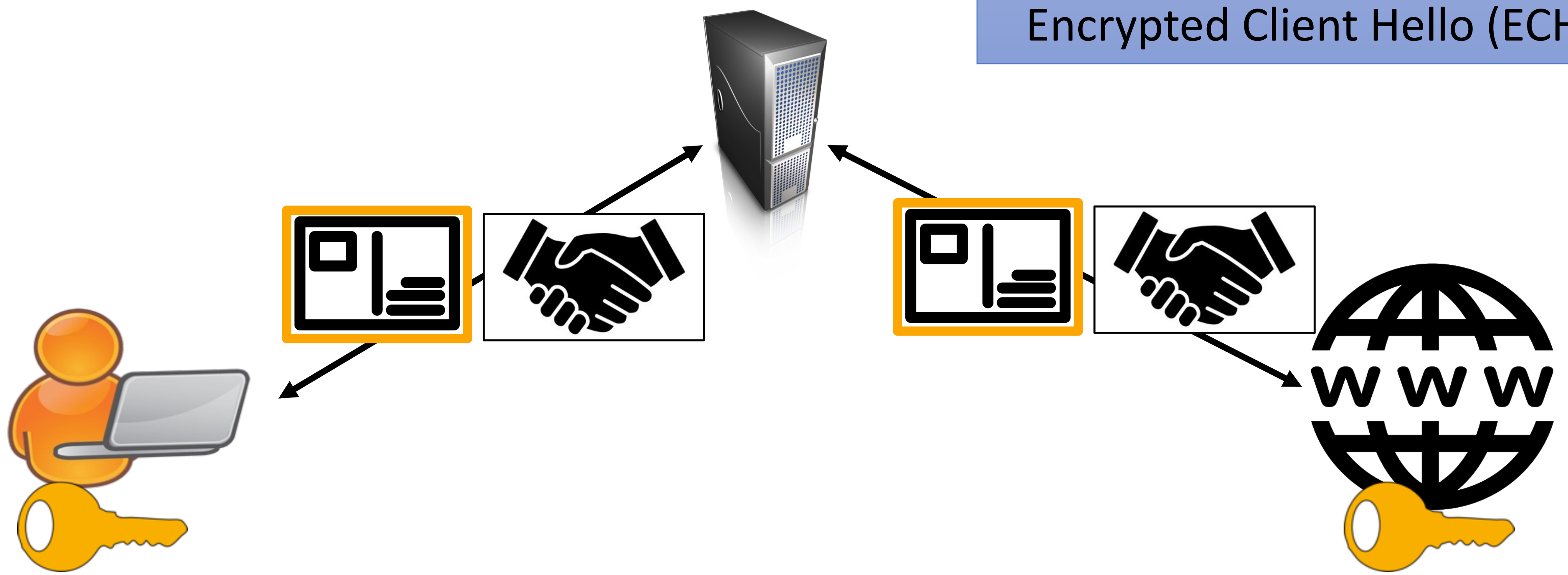
Joint with:

Arasu Arun, Ye Zhang, Joseph Bonneau, Michael Walfish,
Zachary DeStefano, Collin Zhang

Privacy via Encryption

Encryption hides data; increasingly hides metadata too.

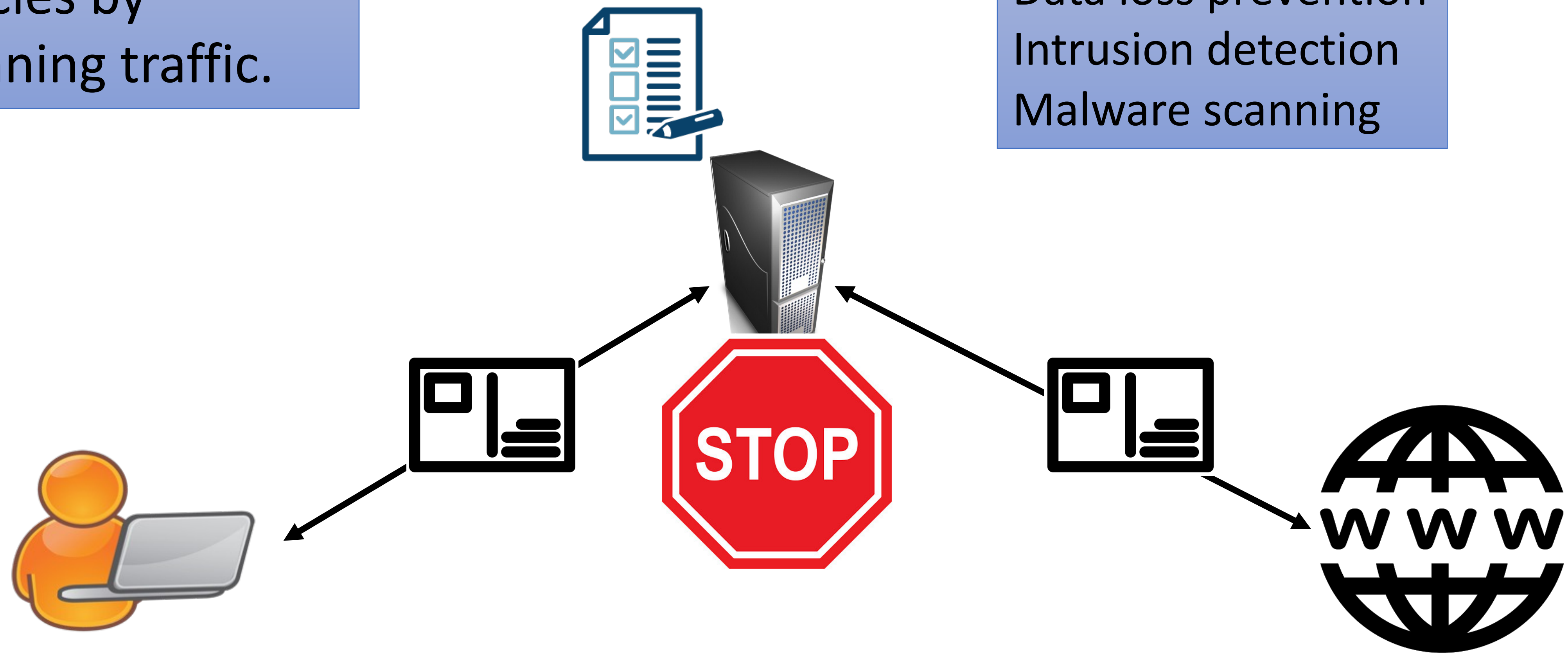
Encrypted DNS (DoH/DoT)
Oblivious DoH
TLS 1.3:
- Encrypted certificates
- Encrypted Client Hello (ECH)



Policy Enforcement

Networks enforce policies by scanning traffic.

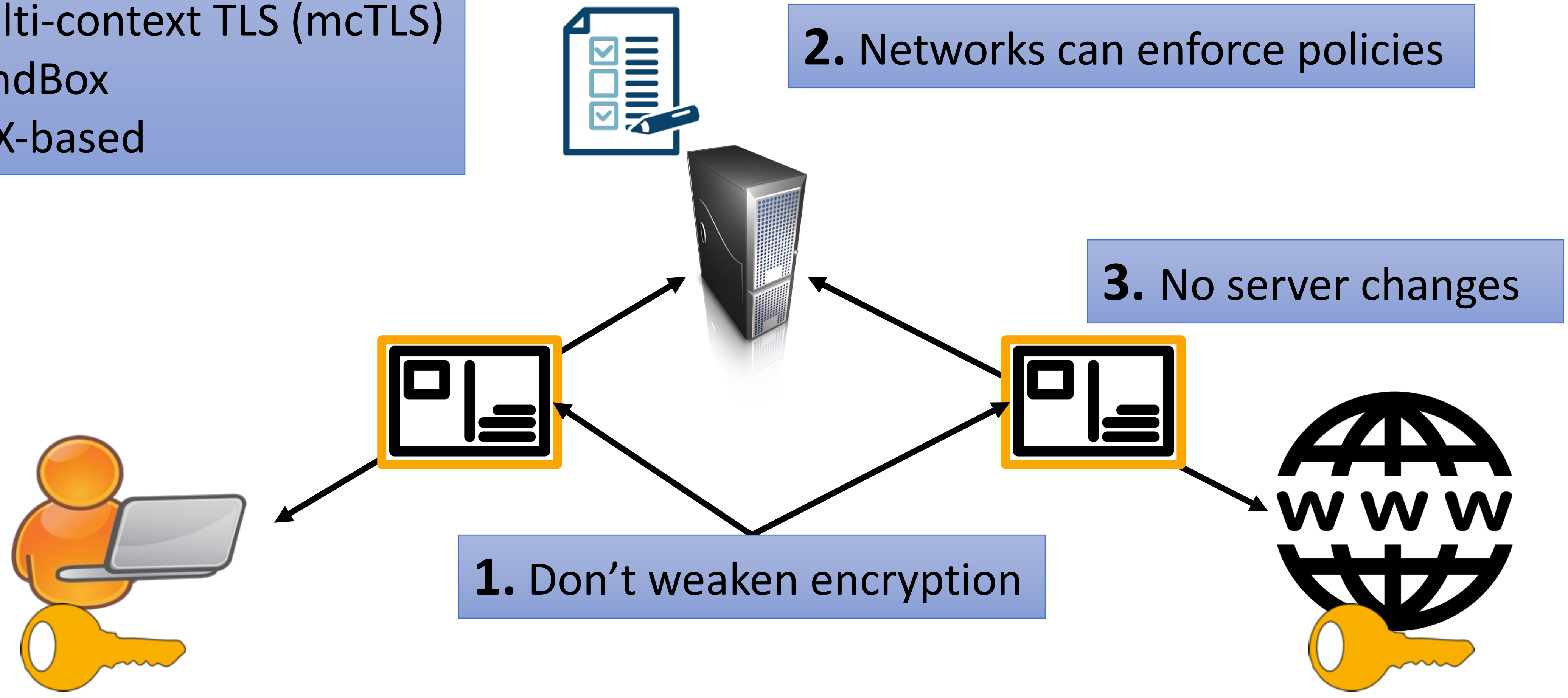
DNS filtering
Data loss prevention
Intrusion detection
Malware scanning



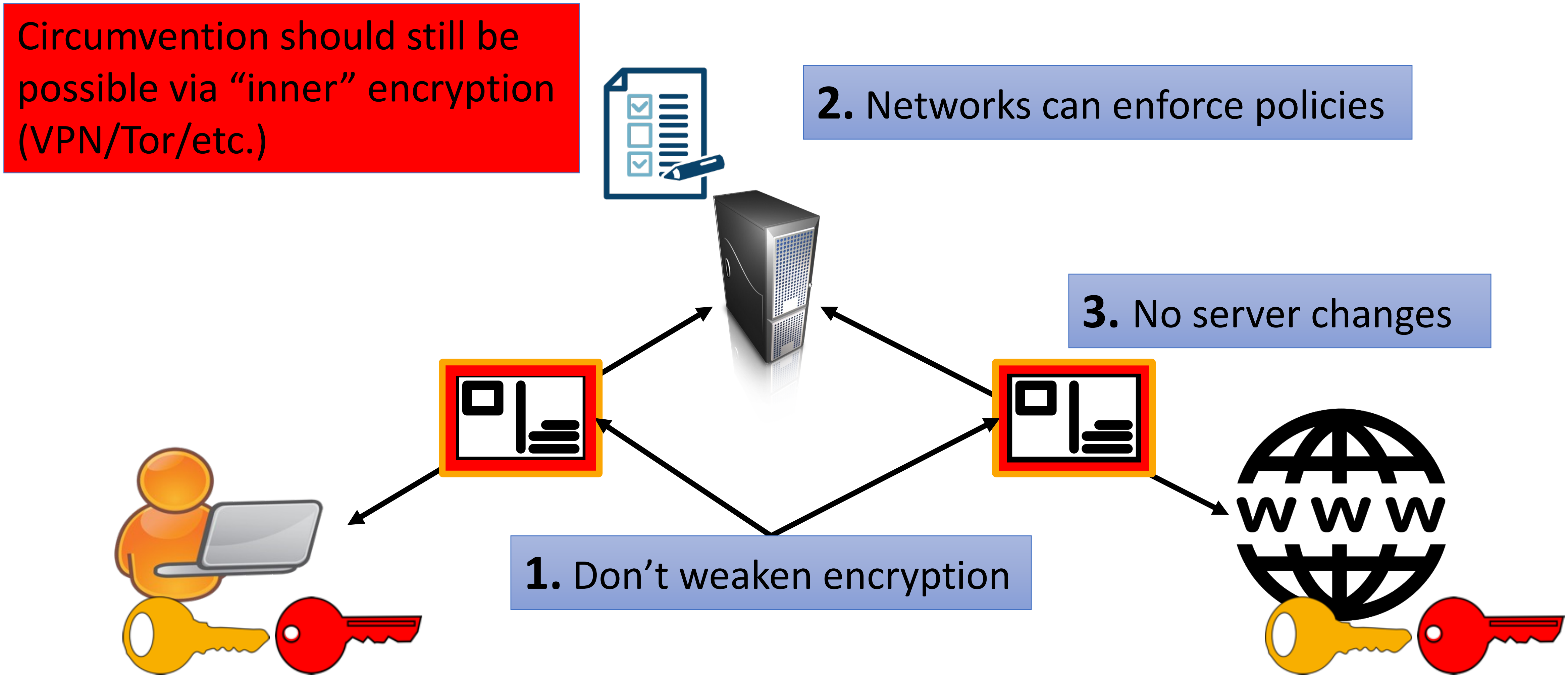
Is it possible to have both privacy and policy enforcement?

Requirements

Prior work:
- multi-context TLS (mcTLS)
- BlindBox
- SGX-based



Circumvention




ZKP

Zero-Knowledge Proofs (ZKPs)

ZKPs let a prover convince a verifier a public statement is true:
1. Without revealing why (zero-knowledge)
2. ***Only*** convince ***if*** statement is true (soundness)

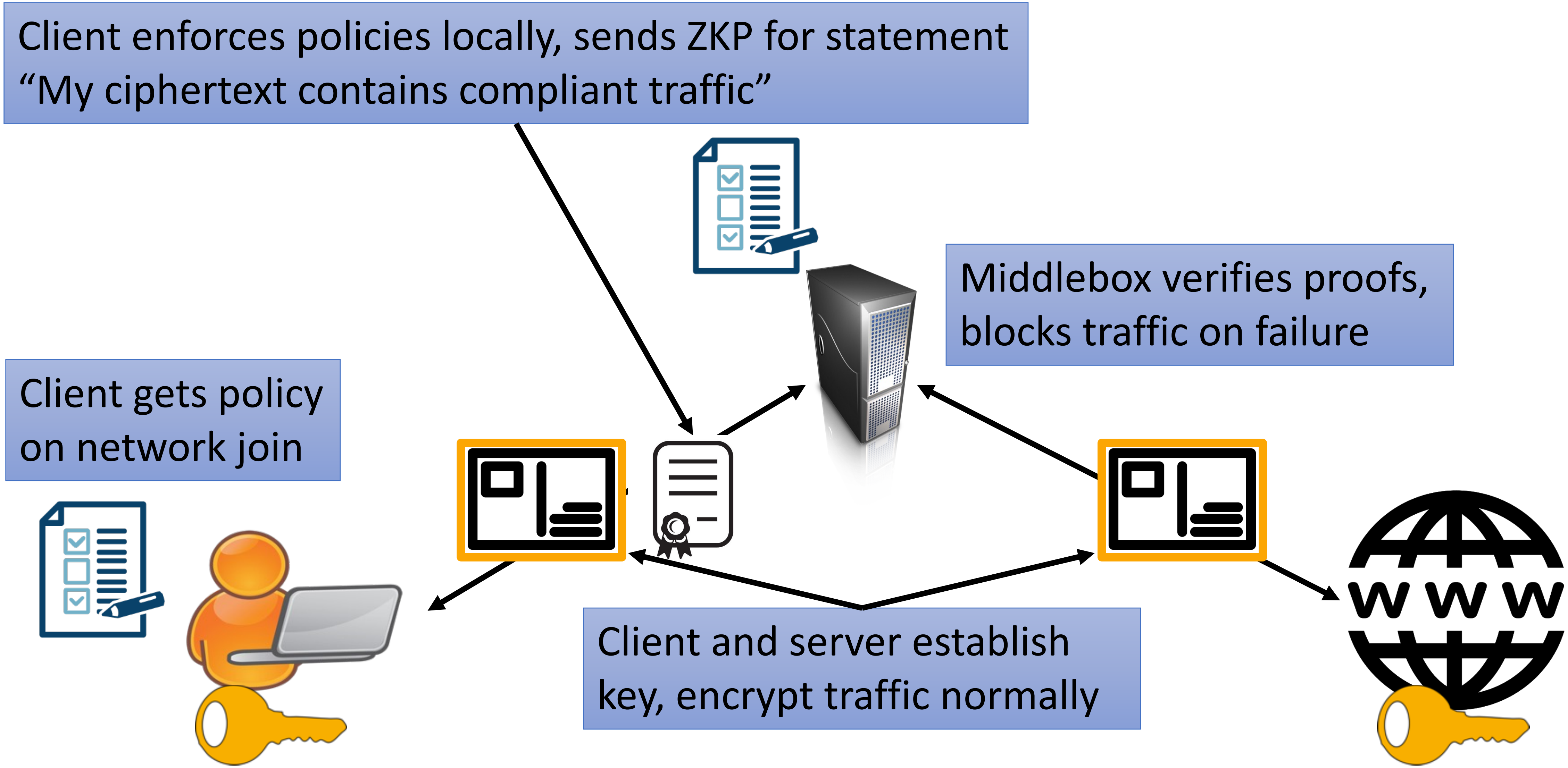
Prover

Prover generates **zero-knowledge proof,** sends to verifier.

Verifier

Verifier checks  , learns if statement is true

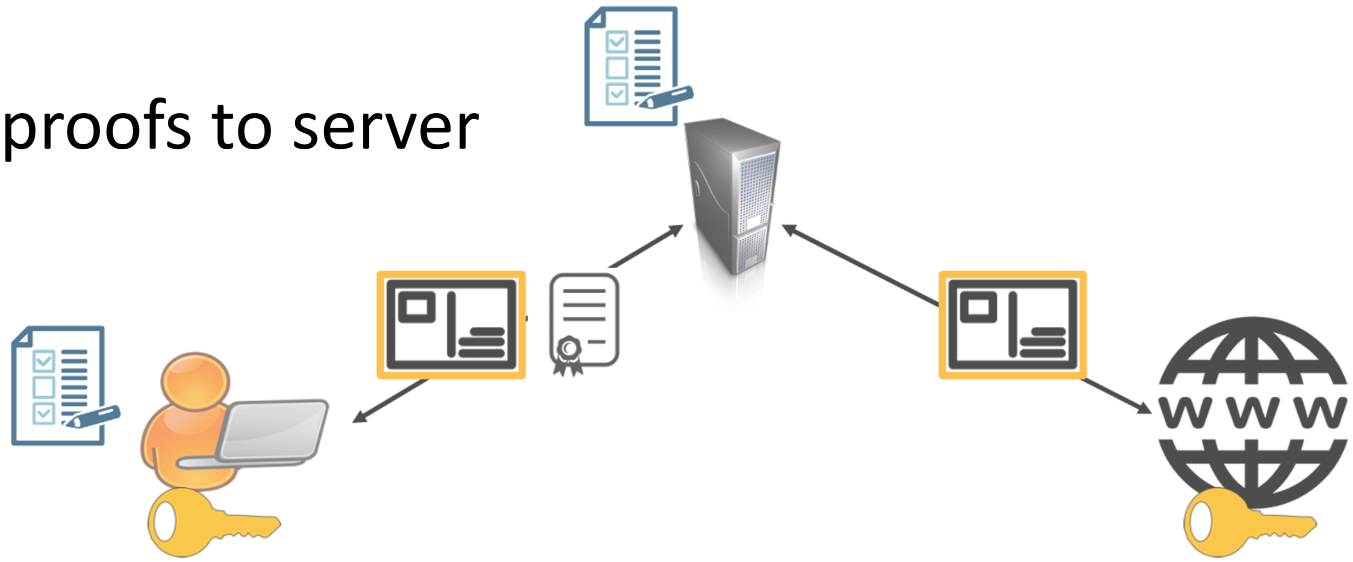
Zero-Knowledge Middleboxes

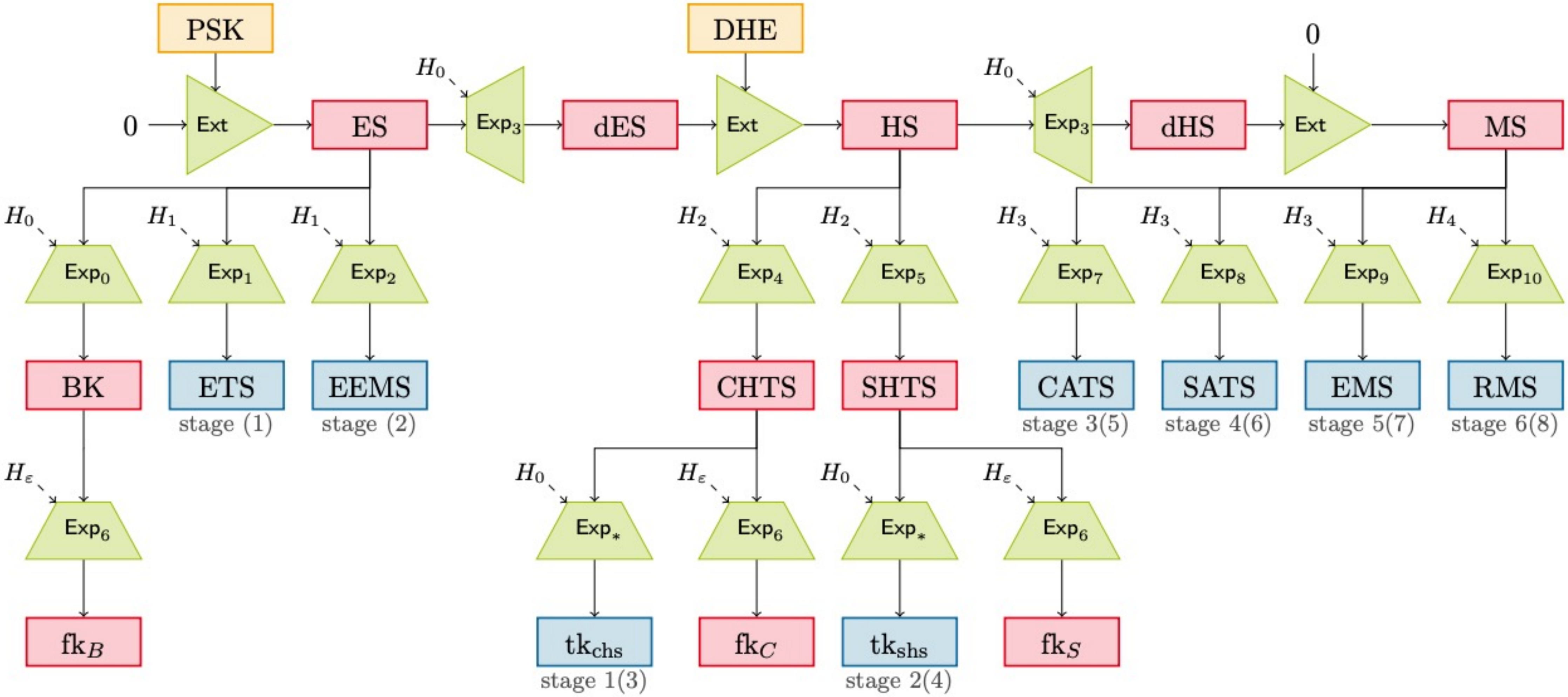


Zero-Knowledge Middleboxes

Requirements:

1. Don't weaken encryption
   - ✓ Using standard encryption + zero-knowledge property of ZKP
2. Middlebox can enforce policies
   - ✓ ZKP soundness
3. No server changes
   - ✓ Middlebox doesn't forward proofs to server



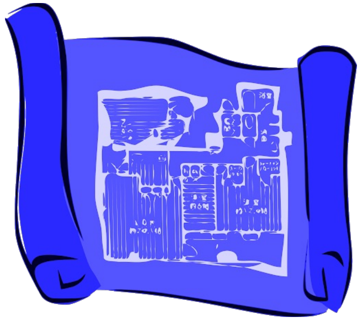


"A Cryptographic Analysis of the TLS 1.3 Handshake Protocol" – Dowling et al.

ZKPs of properties of TLS 1.3 traffic are close to practical!

Zero-Knowledge Middleboxes

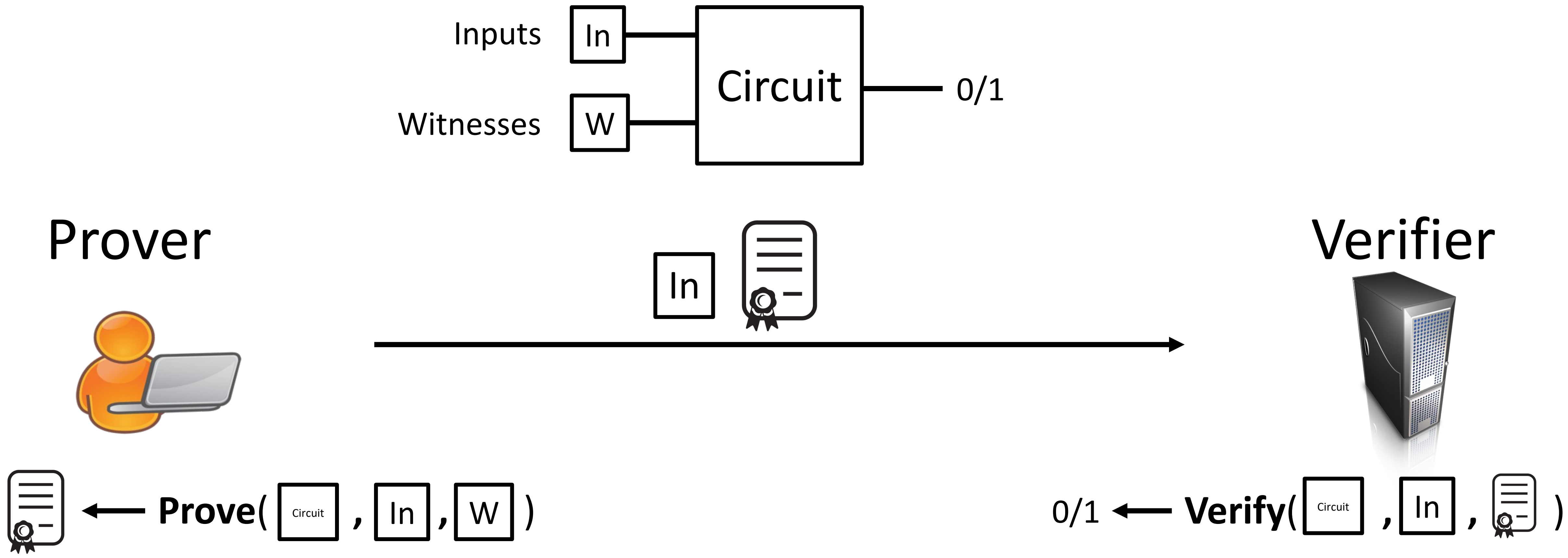
Circuits for ZKMBs,
channel opening



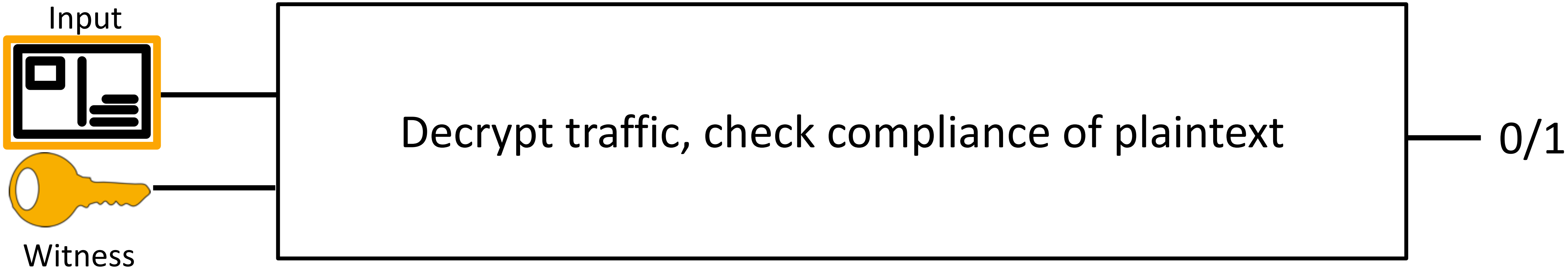
ZKMBs for
encrypted DNS

Future work

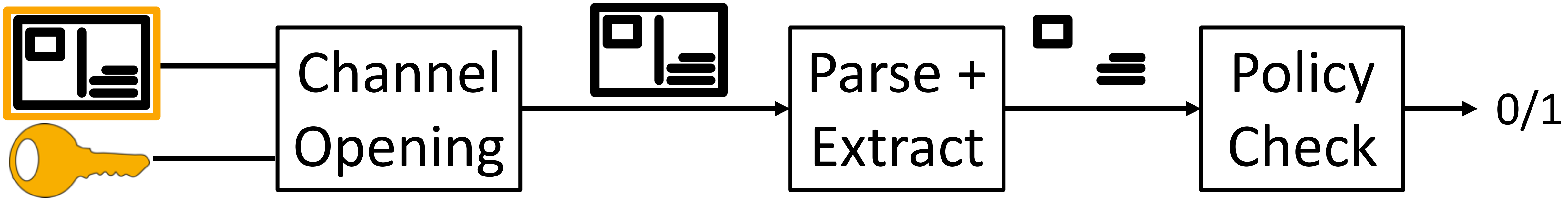
Circuits for ZKPs



ZKMB Circuits



ZKMB Circuits

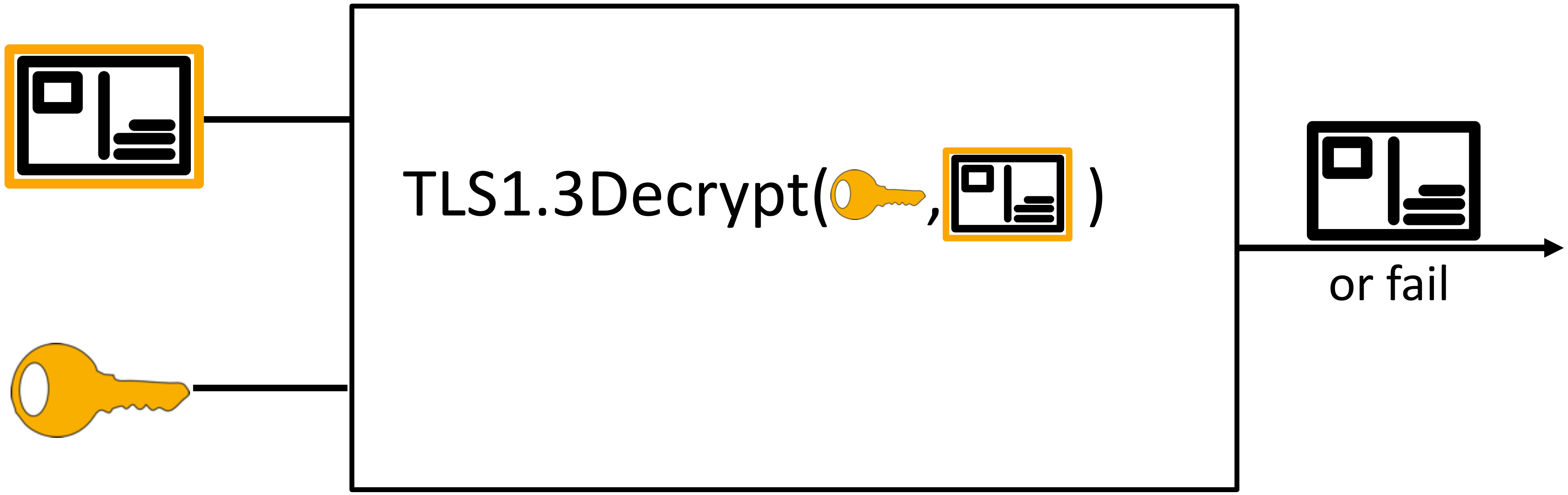


Function	Decrypts ciphertext, outputs message	Finds, outputs relevant data from message	Verifies data is compliant

Channel Opening for TLS 1.3

How to open a TLS 1.3 ciphertext?

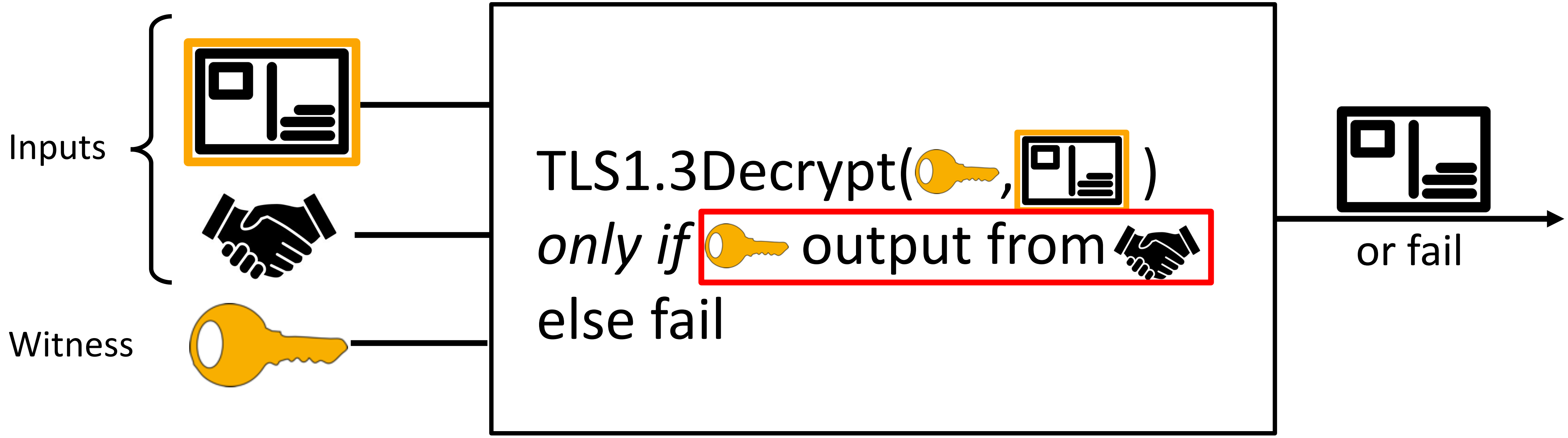
Problem: TLS 1.3 AEADs are not ***binding***: ciphertexts have multiple correct decryptions.



Channel Opening for TLS 1.3

How to open a TLS 1.3 ciphertext?

Idea to fix: client must prove key was handshake output.



Key Consistency Check for TLS 1.3

(the short version)

- Simple, inefficient: re-run most of client's key derivation in circuit.
  - Diffie-Hellman values are binding to shared secret.
- Observation: handshake "commits to" intermediate steps of key derivation. Check these to shortcut key derivation.
- Key consistency check can be done once per TLS 1.3 session
  - Work *amortizes* for long-lived connections (e.g. encrypted DNS)

Zero-Knowledge Middleboxes

Circuits for ZKMBs, channel opening

ZKMBs for encrypted DNS

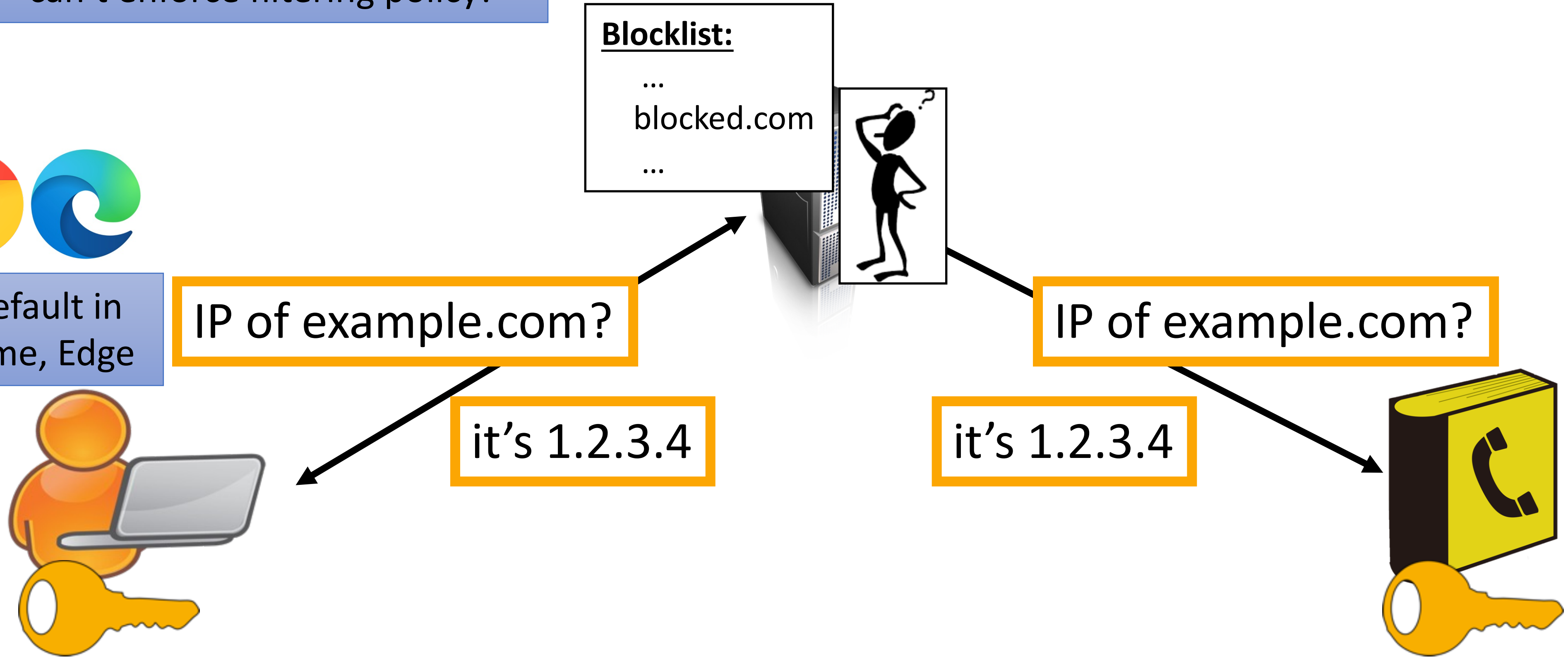
Future work

Encrypted DNS

DNS-over-{HTTPS, TLS}: DNS queries sent to a trusted resolver via TLS 1.3. Bypass local network's resolver.

By design, local network can't see client DNS traffic - can't enforce filtering policy!

Enabled by default in Firefox, Chrome, Edge



Children's Internet Protection Law (2000)

TITLE XVII—CHILDREN'S INTERNET PROTECTION

The protection measures must block or filter Internet access

support.mozilla.org/en-US/kb/configuring-networks-disable-dns-over-https

Download Firefox

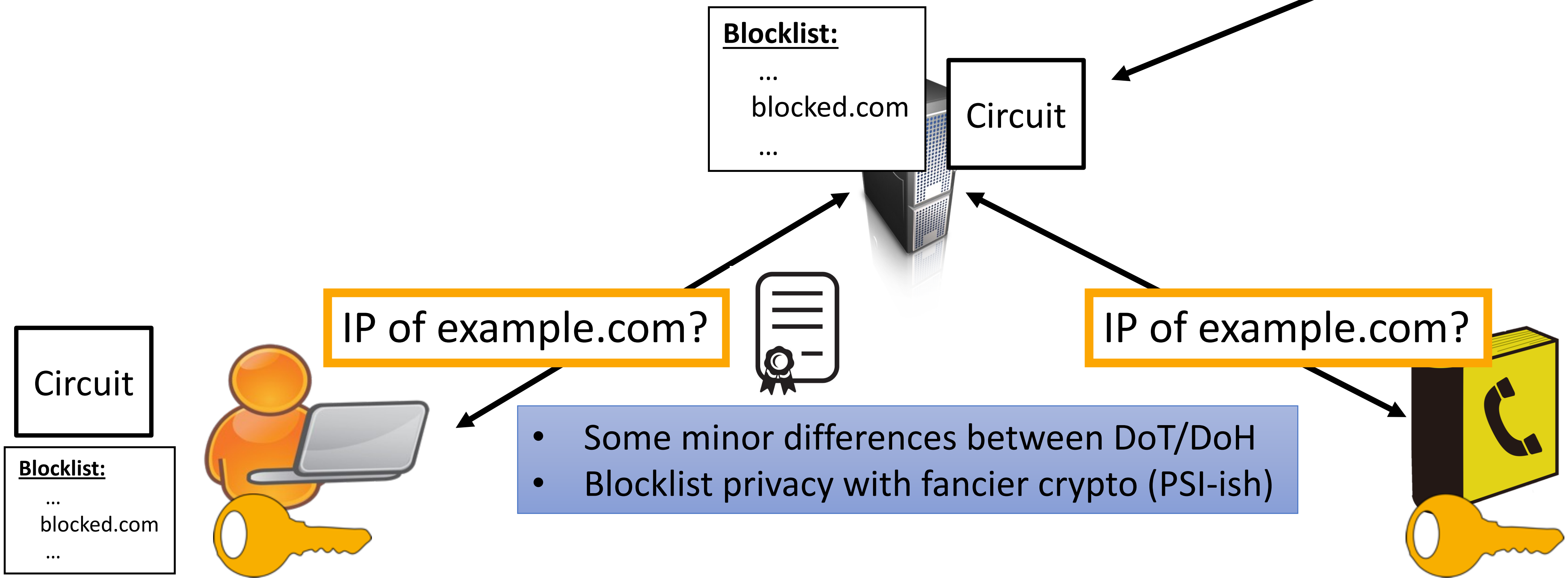
Systems and Languages What's New Privacy

Configuring Networks to Disable DNS over HTTPS

ZKMB for Filtering Encrypted DNS

1. Network creates DNS blocklist and circuit
2. Clients get circuit + blocklist on network join
3. TLS1.3 handshake
4. Client sends query ciphertext + ZKP

Channel opening: decrypt DNS query
Parse+Extract: deserialize domain name
Policy Check: verify set non-membership proof

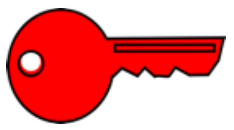


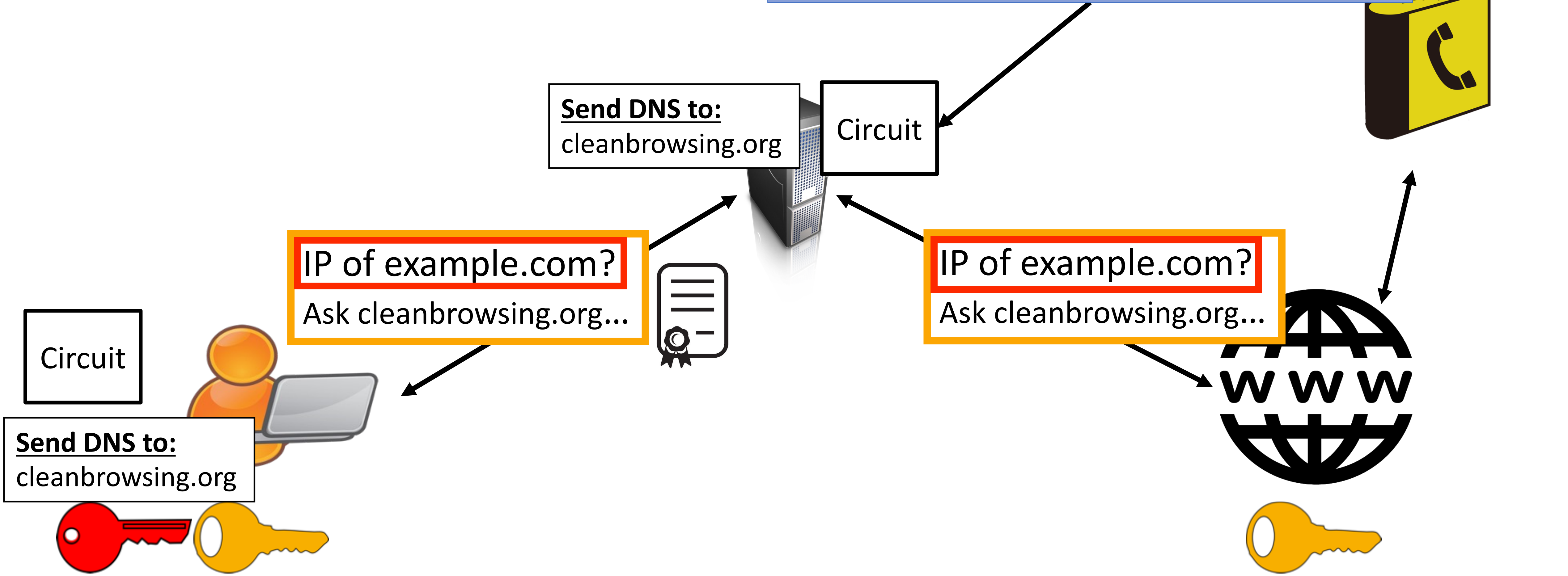
- Some minor differences between DoT/DoH
- Blocklist privacy with fancier crypto (PSI-ish)

ZKMB for Oblivious DoH Allowlist

Policy is that clients must send ODoH to cleanbrowsing.org

Channel opening: decrypt HTTPS request
Parse+Extract: read destination resolver
Policy Check: verify == “cleanbrowsing.org”





Experimental Results

Groth16 ZKP, 8 threads

Key Consistency Proof (once-per-session)

Method	#Gates (mil)	Prv time (s)	SRS (MB)	Proof size (b)	Vf time (ms)
Baseline	7.5	94.0	1200	128	~5
Optimized	1.1	16.5	149	128	~5

7x 6x 8x

DNS Case Studies (excluding once-per-session setup)

Case Study	Ctxt size	#Gates (k)	Prv time (s)	SRS (MB)	Proof size (b)	Vf time (ms)
DoH (AES)	500	495	6.8	75	128	~5
DoT (ChaCha)	255	195	3.1	32	128	~5

Prototype can generate proof for nontrivial ZKMB in 3.1 seconds.

Zero-Knowledge Middleboxes

Circuits for ZKMBs,
channel opening

ZKMBs for
encrypted DNS

Future work

New ZKPs

Key Consistency Proof (once-per-session setup)

Method	#Gates (mil)	Prv time (s)	SRS (MB)	Proof size (b)	Vf time (ms)
Baseline	7.5	94.0	1200	128	~5
Optimized	1.1	16.5	149	128	~5
Optimized, Spartan	1.1	1.7	0.07	49,100	227

10x

DNS Case Studies (excluding once-per-session setup)

Case Study	Ctxt size	#Gates (k)	Prv time (s)	SRS (MB)	Proof size (b)	Vf time (ms)
DoH (AES)	500	495	6.8	75	128	~5
DoT (ChaCha)	255	195	3.1	32	128	~5

ChaCha Decryption (excluding once-per-session setup)

ChaCha, Spartan	255	85	0.2	0.02	21,600	28

Conclusion

- Initiated a new line of work on *zero-knowledge middleboxes*, which use ZKPs to enable privacy-preserving enforcement of network policies
- One application is DNS filtering. We designed ZKMB for DoT/DoH blocklisting and Oblivious DoH allowlisting. See paper for HTTPS firewall case study
- Zero-knowledge middleboxes have other exciting applications, and raise many interesting open questions in networking, security, systems, and cryptography

Thanks for listening! Any questions?