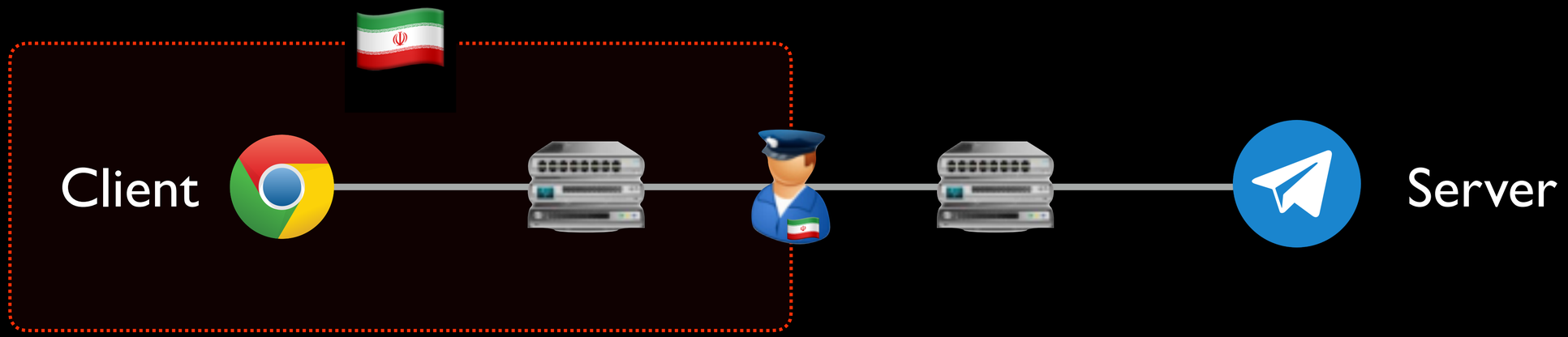
Detecting and Evading Censorship in Depth: A Case Study of Iran's Protocol Filter

Kevin Bock

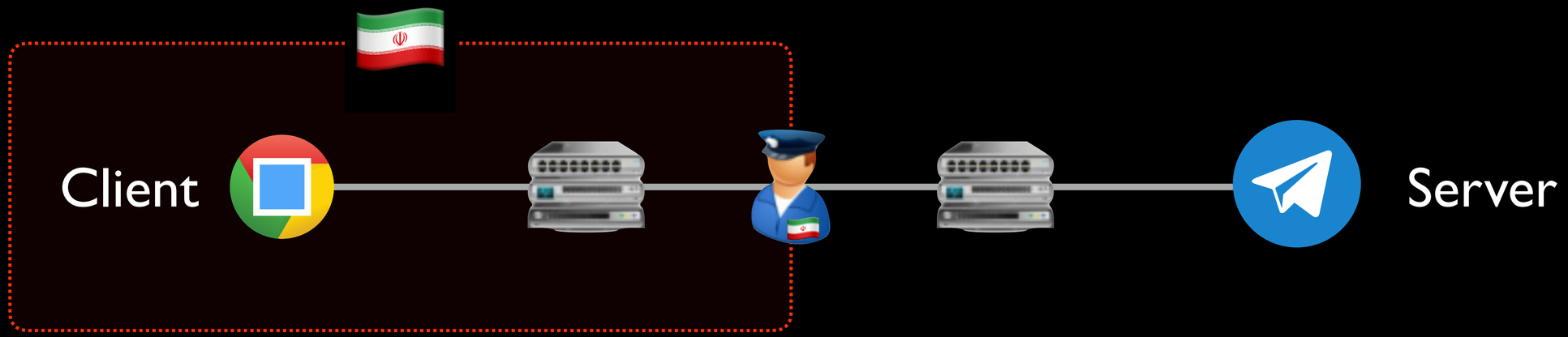
Yair Fax, Kyle Reese, Jasraj Singh, Dave Levin

UNIVERSITY OF MARYLAND

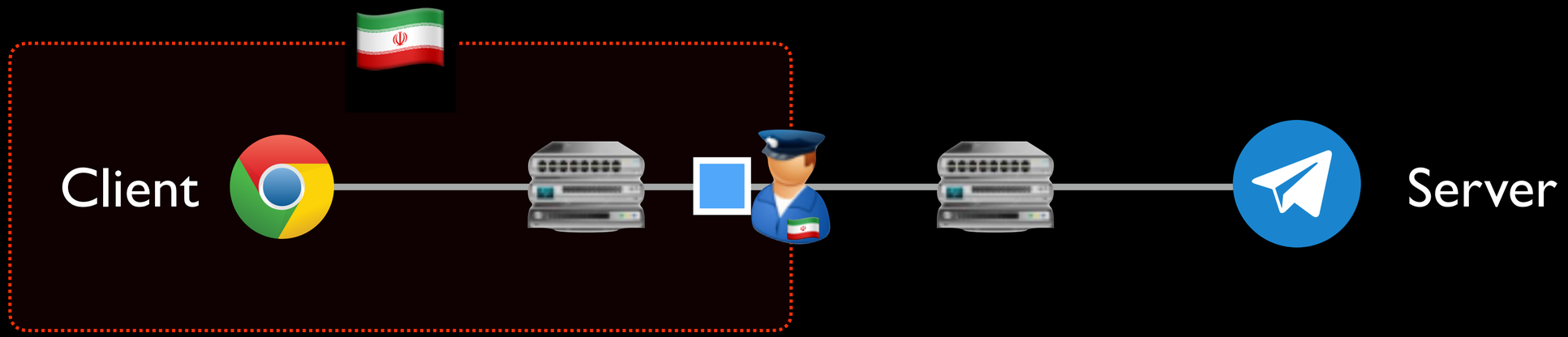
Iranian Censorship



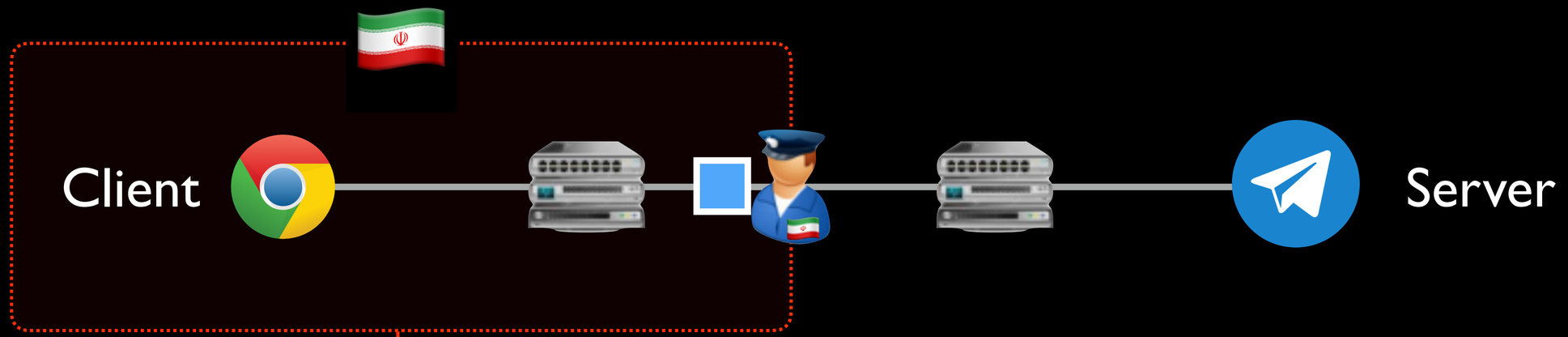
Iranian Censorship



Iranian Censorship

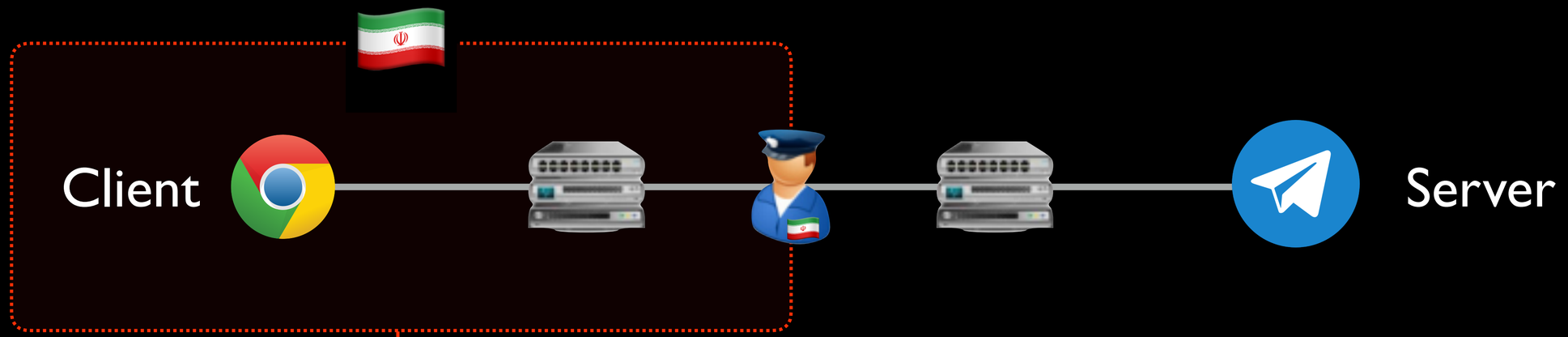


Iranian Censorship



- Censors DNS, HTTP, HTTPS
- Deep-packet inspection for keywords
- Typically injects dummy responses or blackholes
- Applies to all destination IP addresses

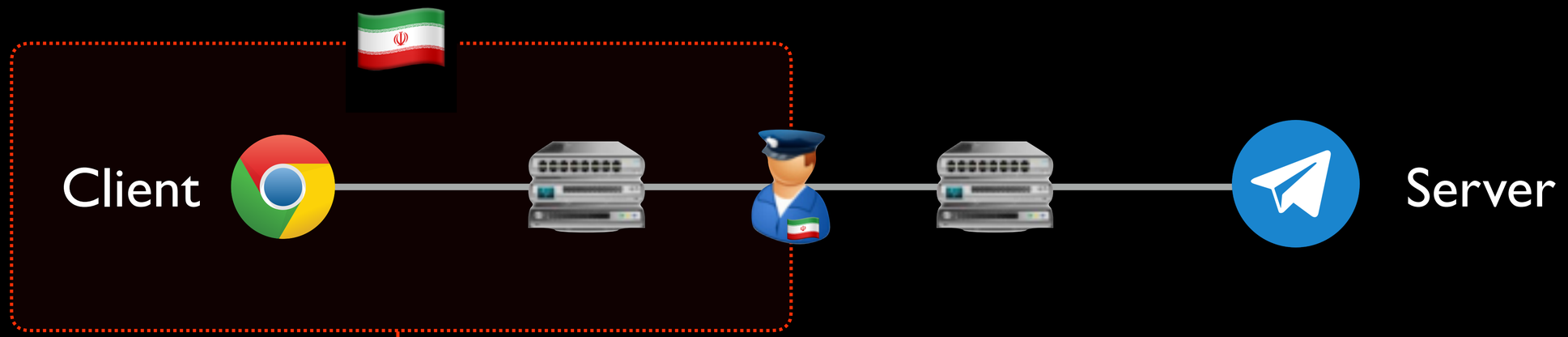
Iranian Censorship



- Censors DNS, HTTP, HTTPS
- Deep-packet inspection for keywords
- Typically injects dummy responses or blackholes
- Applies to all destination IP addresses

February 2020

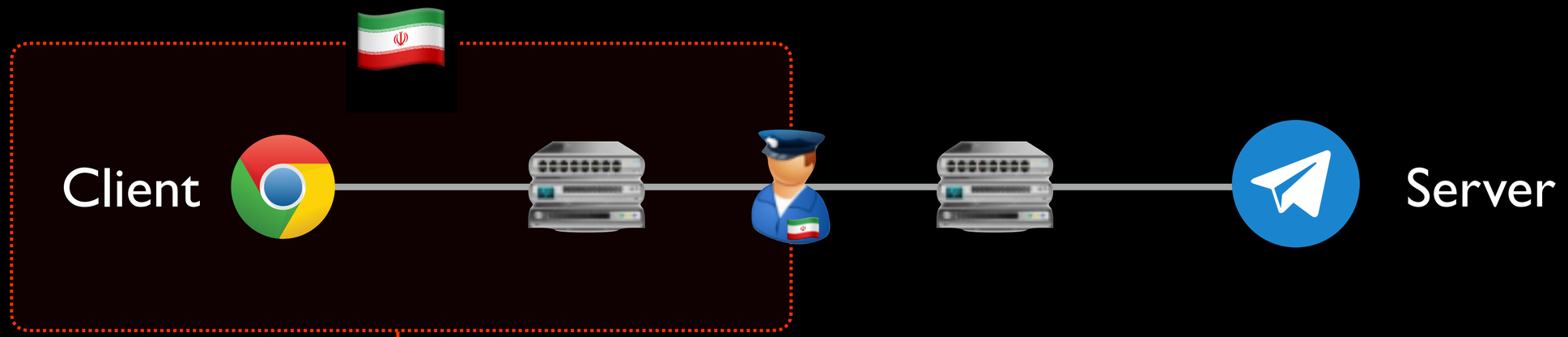
Iranian Censorship



- Censors DNS, HTTP, HTTPS
- Censored *before* keywords
- Typically injects dummy responses or blackholes
- Applies to all destination IP addresses

February 2020

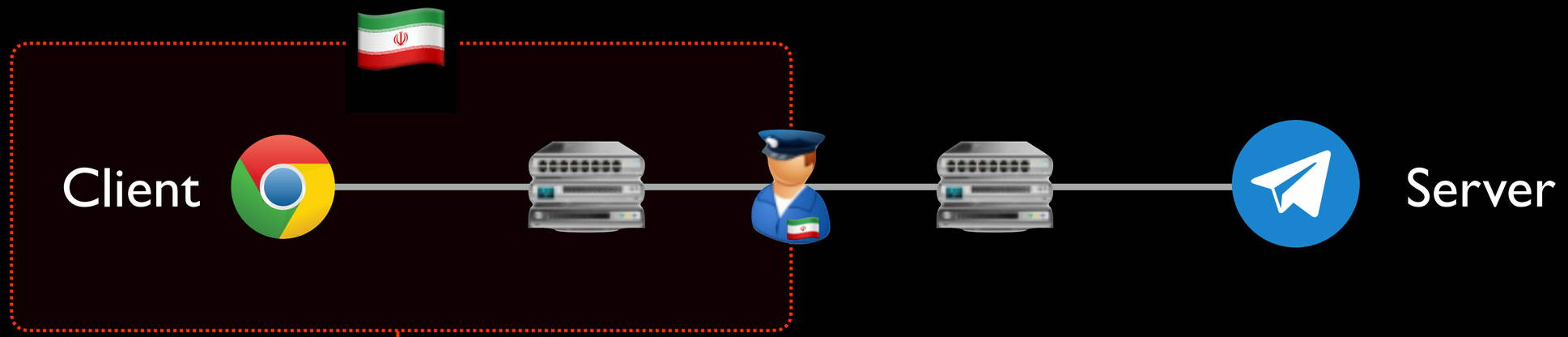
Iranian Censorship



- Censors DNS, HTTP, HTTPS
- Censored *before* keywords
- No injections; only packet drops
- Applies to all destination IP addresses

February 2020

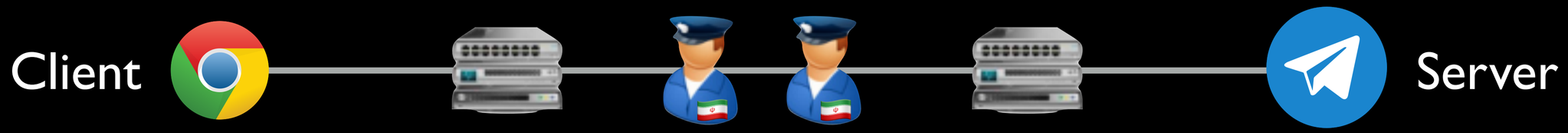
Iranian Censorship



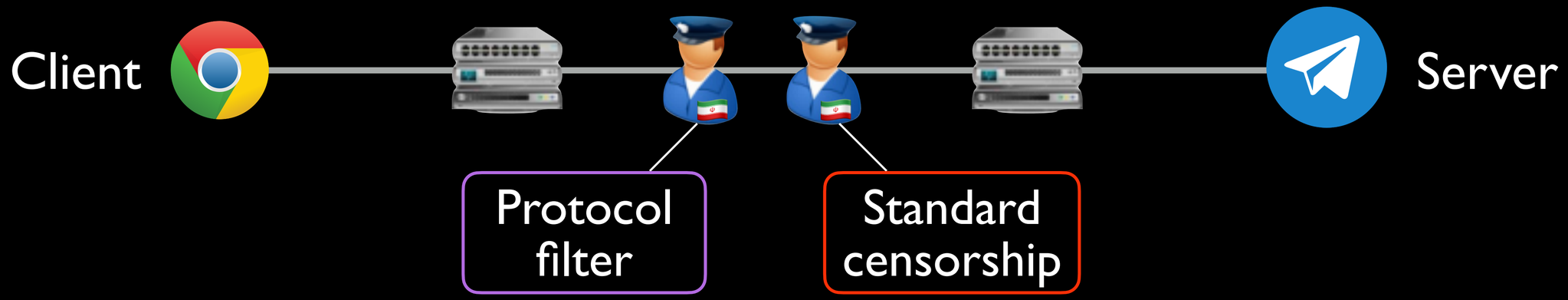
- Censors DNS, HTTP, HTTPS
- Censored *before* keywords
- No injections; only packet drops
- Applied only to *some* destinations

February 2020

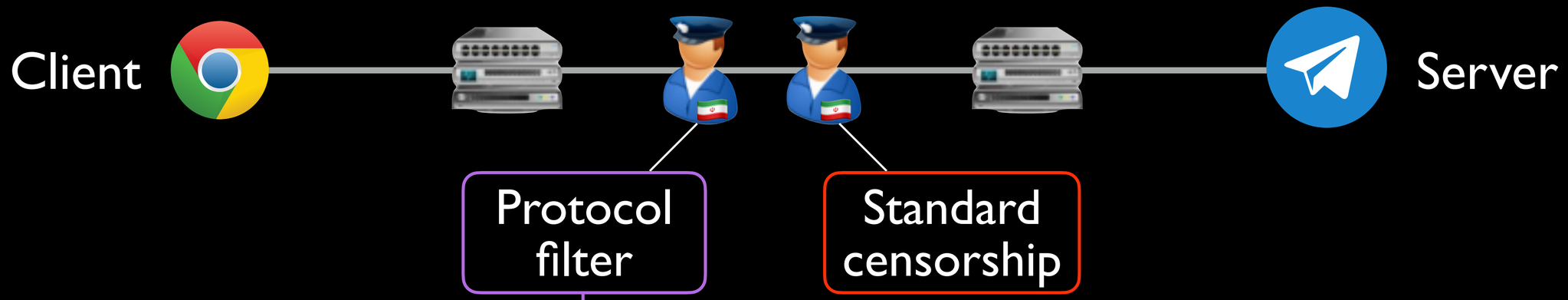
Iran's Protocol Filter



Iran's Protocol Filter

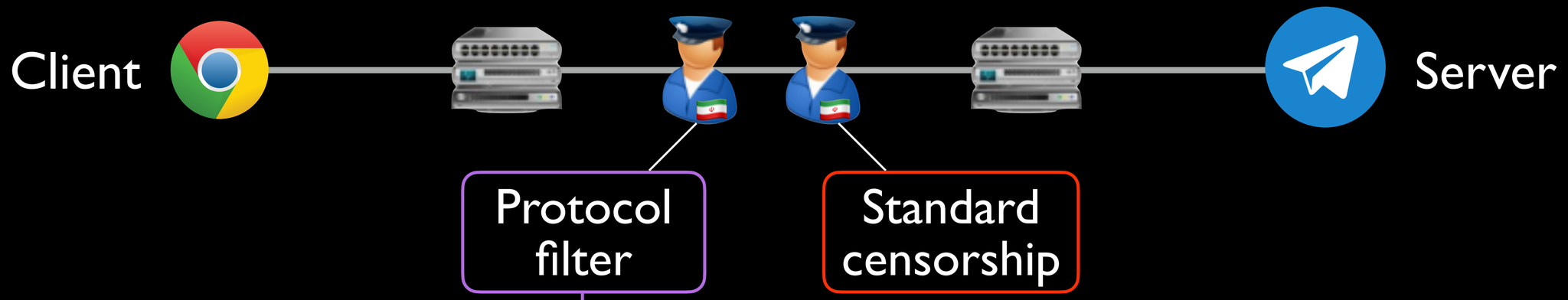


Iran's Protocol Filter



- Matches connections to protocol fingerprints
- Drops connections from unrecognized protocols
- Applies only to *some* destinations

Iran's Protocol Filter



- Matches connections to protocol fingerprints
- Drops connections from unrecognized protocols
- Applies only to *some* destinations

Composes with standard censorship, creating censorship-in-depth

Iran's Protocol Filter

- How the protocol filter works
  - How it fingerprints connections
  - How it censors connections
- How the filter can be defeated
  - Packet manipulation strategies
  - Client-side and server-side

Iran's Protocol Filter

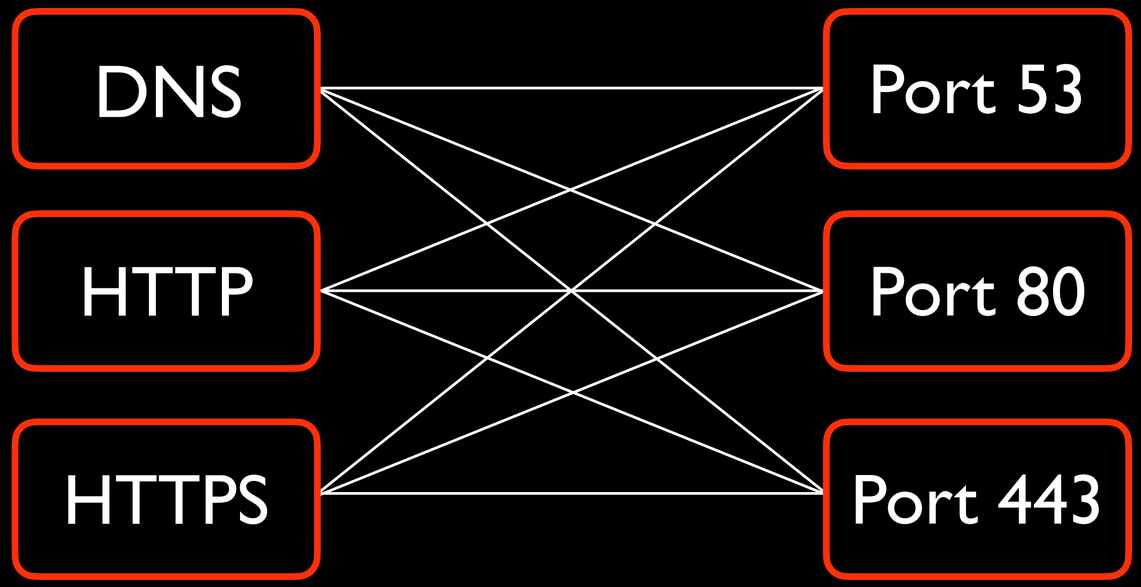
- How the protocol filter works
  - How it fingerprints connections
  - How it censors connections
- How the filter can be defeated
  - Packet manipulation strategies
  - Client-side and server-side

How it fingerprints connections

1. Performs deep packet inspection
2. Only matches on TCP
3. Only allows:
   - DNS
   - HTTP
   - HTTPS
4. Only monitors:
   - Port 53
   - Port 80
   - Port 443

How it fingerprints connections

1. Performs deep packet inspection
2. Only matches on TCP
3. Only allows:
4. Only monitors:



Protocol-specific Fingerprints

DNS

HTTP

HTTPS

DNS Fingerprint

1. TCP payload ≥ 12 bytes long
2. Query/response field = 0
3. Question count < 15
4. Answer count = 0
5. Must have a valid DNS-over-UDP header

Violating any of these results in censorship

DNS Fingerprint

5 Must have a valid DNS-over-UDP header

```
DNS header
```

DNS Fingerprint

5 Must have a valid DNS-over-UDP header

```
DNS header
```

The filter only applies to DNS-over-TCP

DNS Fingerprint

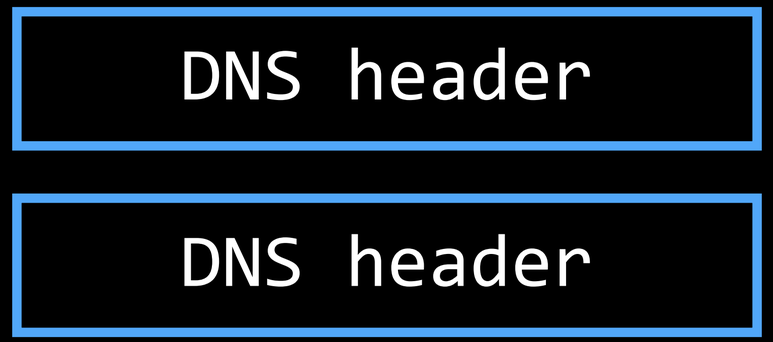
5 Must have a valid DNS-over-UDP header

```
DNS header
```

The filter only applies to DNS-over-TCP

DNS Fingerprint

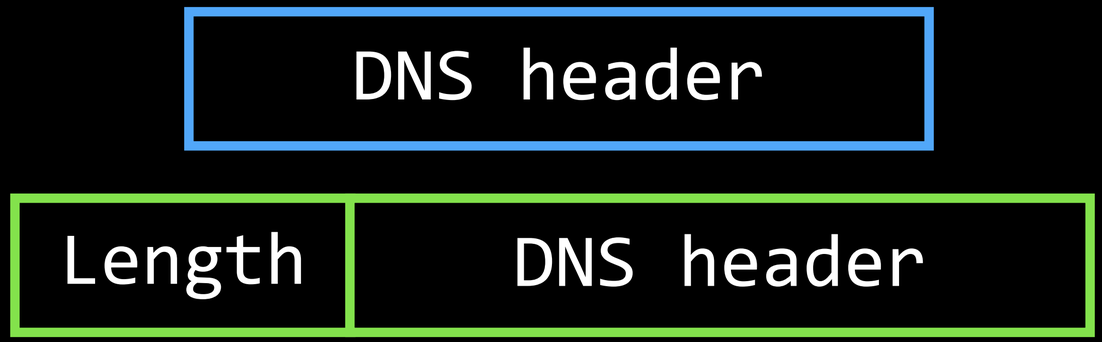
5 Must have a valid DNS-over-UDP header



The filter only applies to DNS-over-TCP

DNS Fingerprint

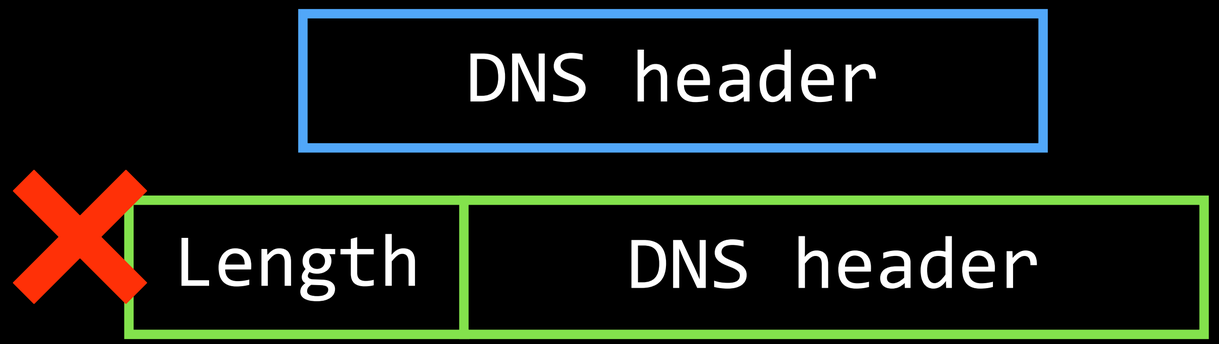
5 Must have a valid DNS-over-UDP header



The filter only applies to DNS-over-TCP

DNS Fingerprint

5 Must have a valid DNS-over-UDP header



The filter only applies to DNS-over-TCP

Iran's protocol filter *never* permits DNS-over-TCP

Protocol-specific Fingerprints

DNS	Matches for *DNS-over-UDP* header
HTTP	Matches for *most* HTTP verbs
HTTPS	Matches on the TLS ClientHello

Details for each in the paper

How the filter censors connections

Drops all packets from the client

How the filter censors connections

Drops all packets from the client

How the filter censors connections

Drops all packets from the client

How the filter censors connections

Drops all packets from the client

How the filter censors connections

Drops all packets from the client

How the filter censors connections

Drops all packets from the client

How the filter censors connections

Drops all packets from the client
after the first non-matching packet

How the filter censors connections

Drops all packets from the client
after the first non-matching packet

How the filter censors connections

Drops all packets from the client
after the first non-matching packet

How the filter censors connections

Drops all packets from the client
after the first non-matching packet

The filter is not bidirectional

Filters and drops only outbound packets

The filter is not bidirectional

Filters and drops only outbound packets

The filter is not bidirectional

Filters and drops only outbound packets

The filter is not bidirectional

Filters and drops only outbound packets

The filter is not bidirectional

Filters and drops only outbound packets

The filter is not bidirectional

Filters and drops only outbound packets

The filter is not bidirectional

Filters and drops only outbound packets

Requires vantage points within Iran to study

Other details in the paper

Protocol fingerprints	Full details on DNS, HTTP, HTTPS
Whom does it censor?	Some entire /24 and /16 prefixes But no clear pattern as to whom
How many packets?	First two data-carrying packets
How long does it block?	60 sec (resets with each packet)

Iran's Protocol Filter

- How the protocol filter works
  - How it fingerprints connections
  - How it censors connections
- How the filter can be defeated
  - Packet manipulation strategies
  - Client-side and server-side

Iran's Protocol Filter

How the protocol filter works

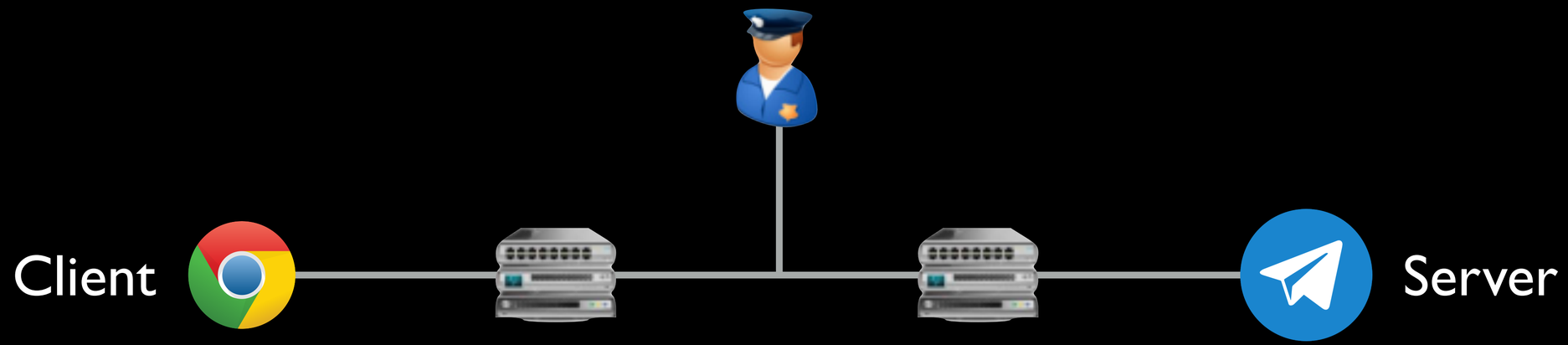
- How it fingerprints connections
- How it censors connections

How the filter can be defeated

- Packet manipulation strategies
- Client-side and server-side

Geneva

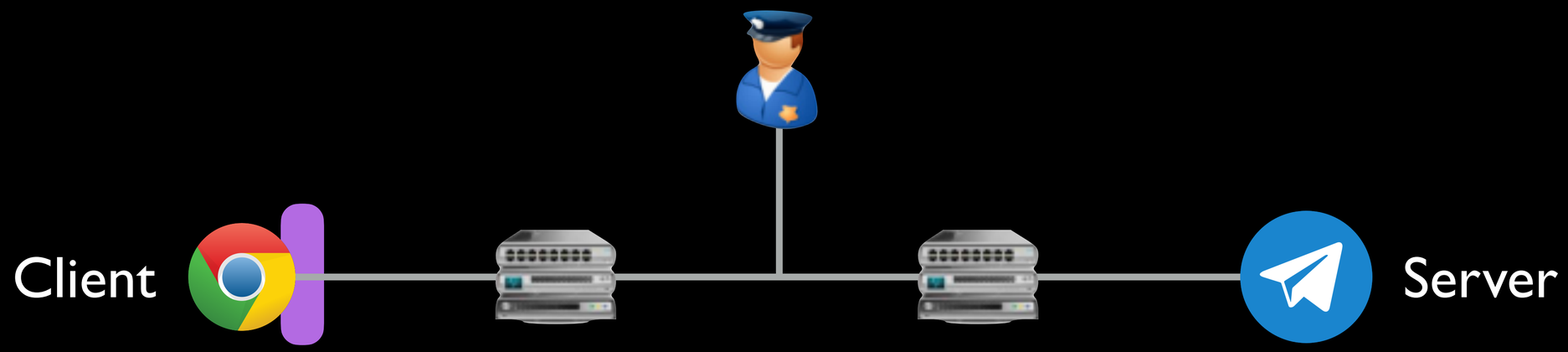
Genetic Evasion



Bock et al. CCS'19

Geneva

Genetic Evasion

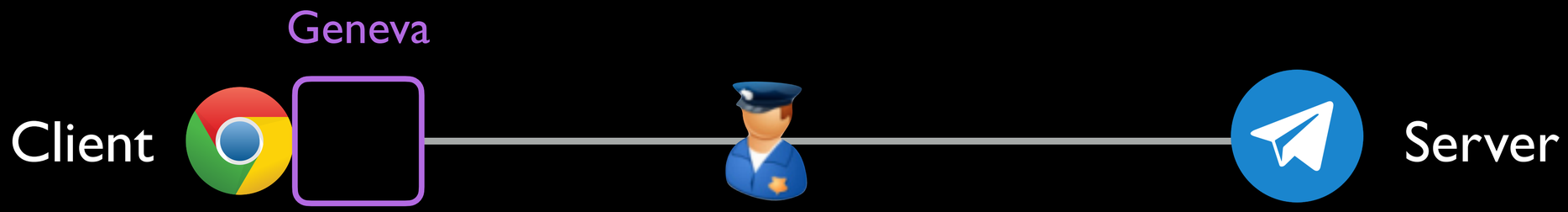


Geneva runs strictly at one side

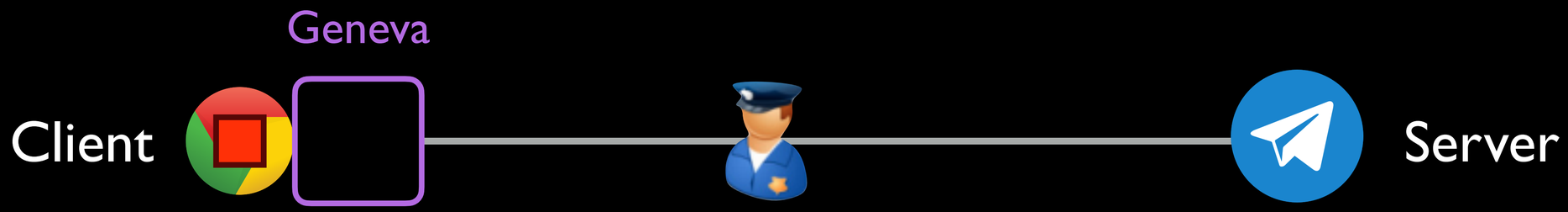
Discovers packets manipulations to evade censorship

Bock et al. CCS'19

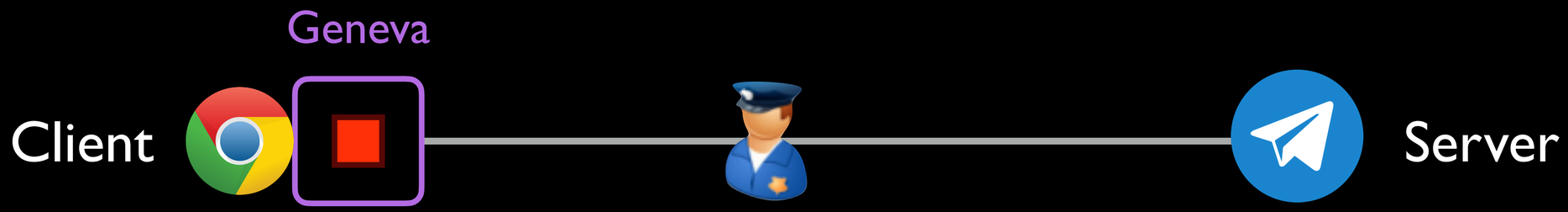
① Inject Fingerprint



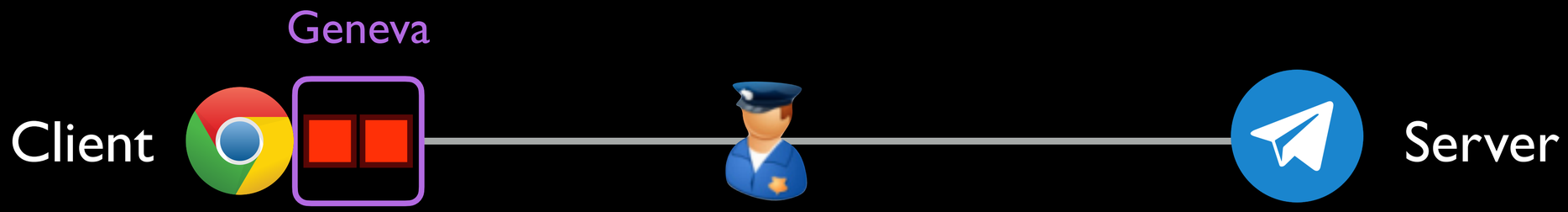
① Inject Fingerprint



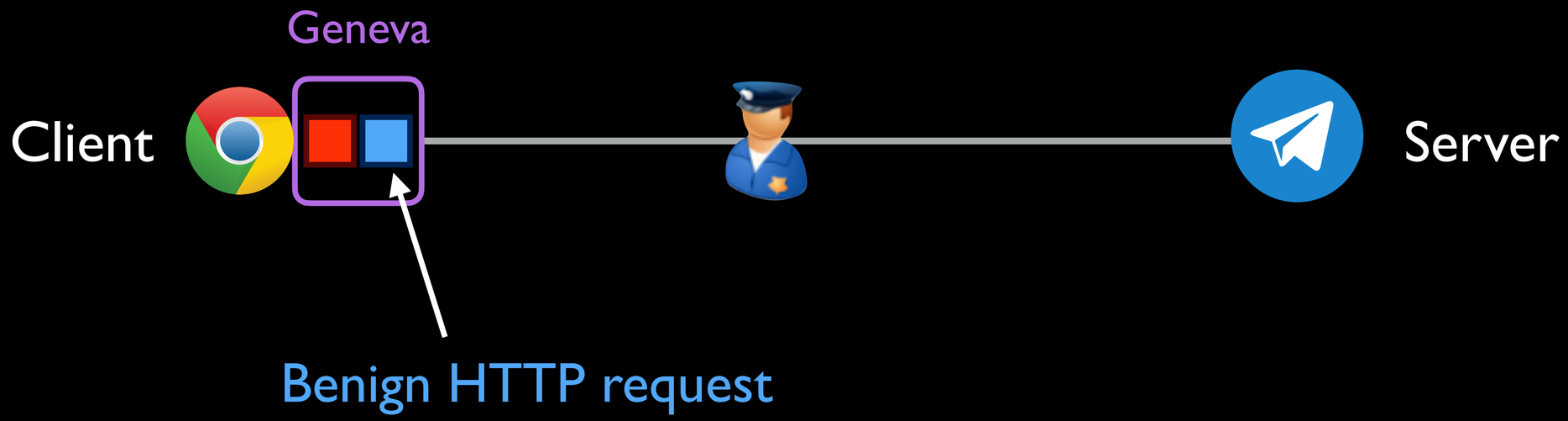
① Inject Fingerprint



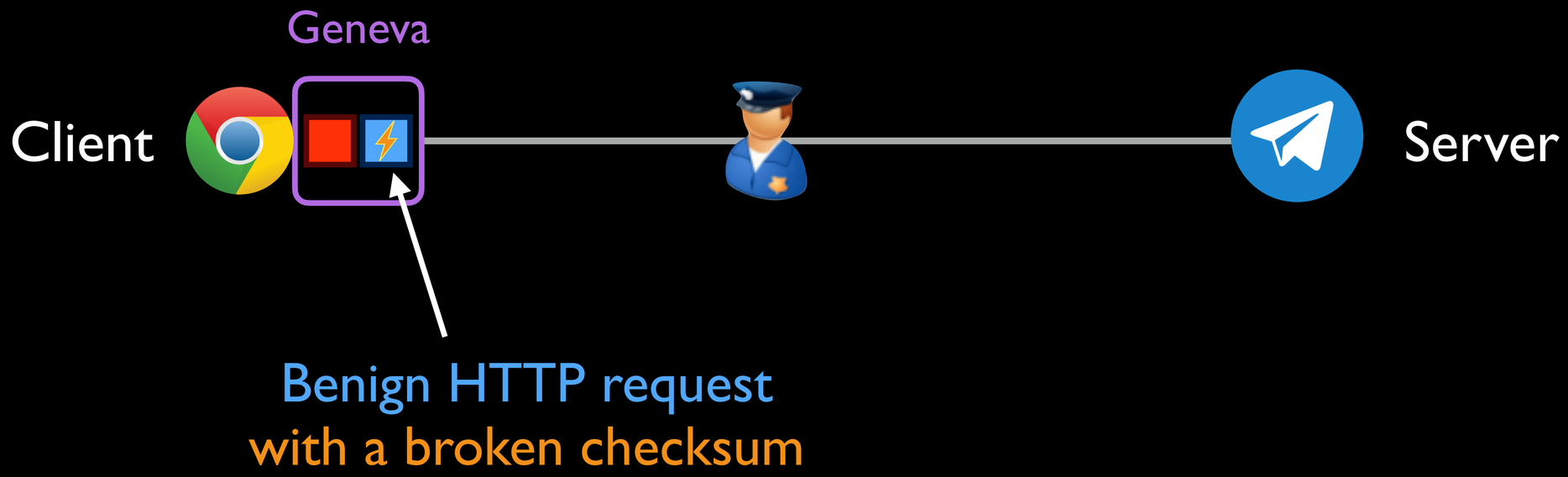
① Inject Fingerprint



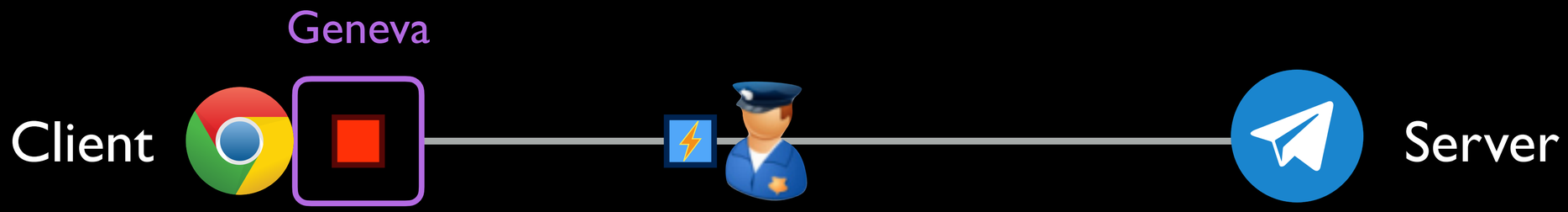
① Inject Fingerprint



① Inject Fingerprint

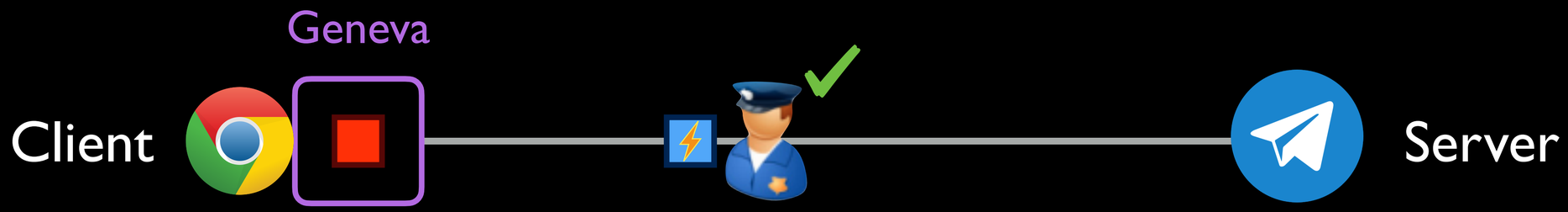


① Inject Fingerprint



Benign HTTP request
with a broken checksum

① Inject Fingerprint



Benign HTTP request
with a broken checksum

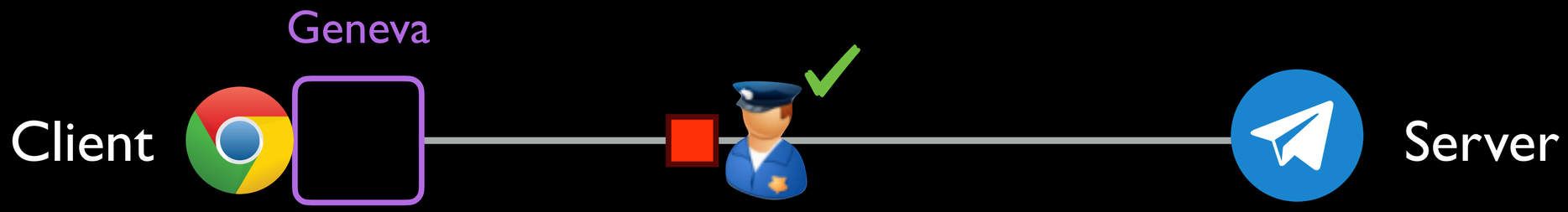
① Inject Fingerprint

Benign HTTP request
with a broken checksum

① Inject Fingerprint

Benign HTTP request
with a broken checksum

① Inject Fingerprint



Benign HTTP request
with a broken checksum

① Inject Fingerprint

Benign HTTP request
with a broken checksum

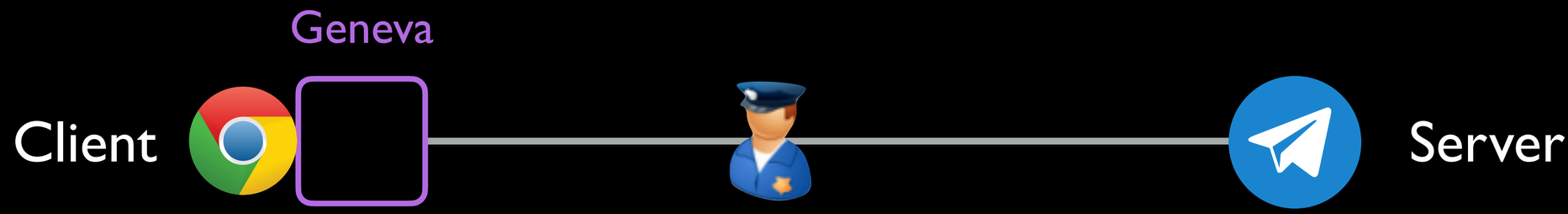
① Inject Fingerprint

Benign HTTP request
with a broken checksum

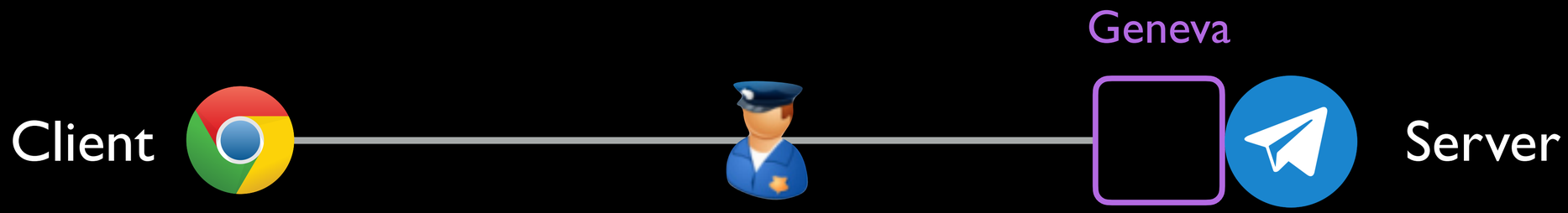
Client-side Evasion Strategies

1. HTTP Request — Send a benign HTTP request with a broken checksum
2. FIN × 2 — Send 2 FIN packets before starting the handshake
3. ACK × 9 — Send 9 ACK packets to complete the handshake

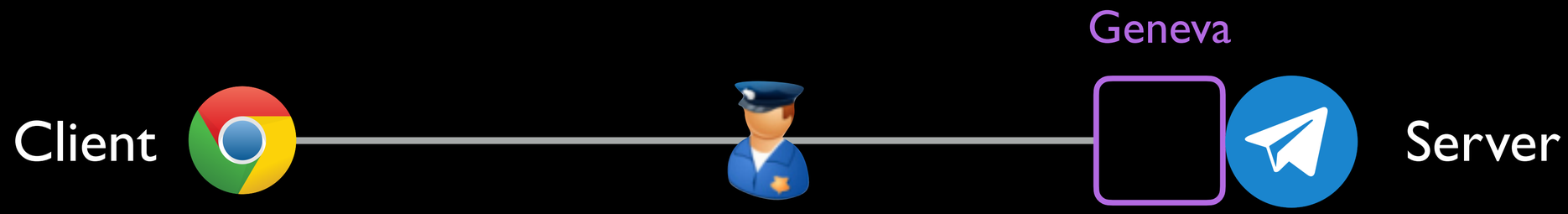
Server-side Evasion Strategies



Server-side Evasion Strategies

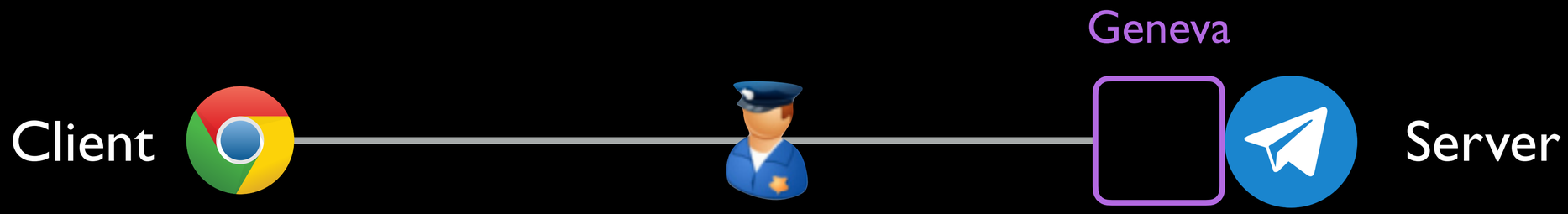


Server-side Evasion Strategies



Requires no client-side modifications

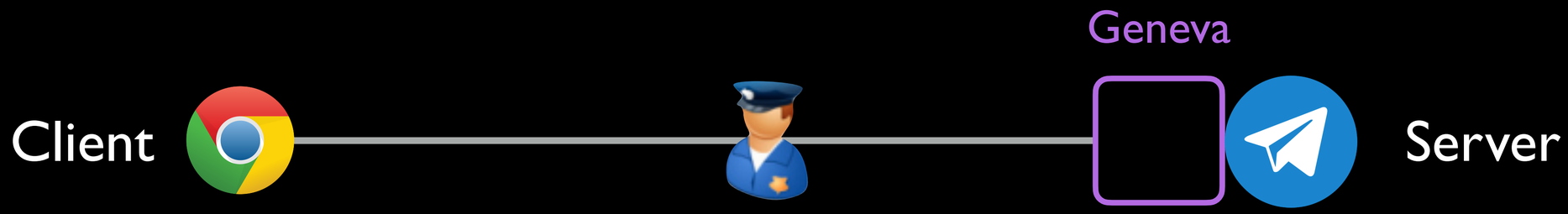
Server-side Evasion Strategies



S+A ×9

Send 9 SYN/ACK packets
with incorrect ack numbers

Server-side Evasion Strategies



S+A ×9

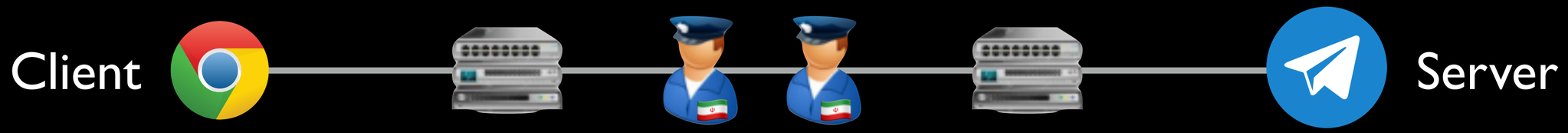
Send 9 SYN/ACK packets
with incorrect ack numbers

Induces 9 RSTs from the client

Evading Iran's Protocol Filter

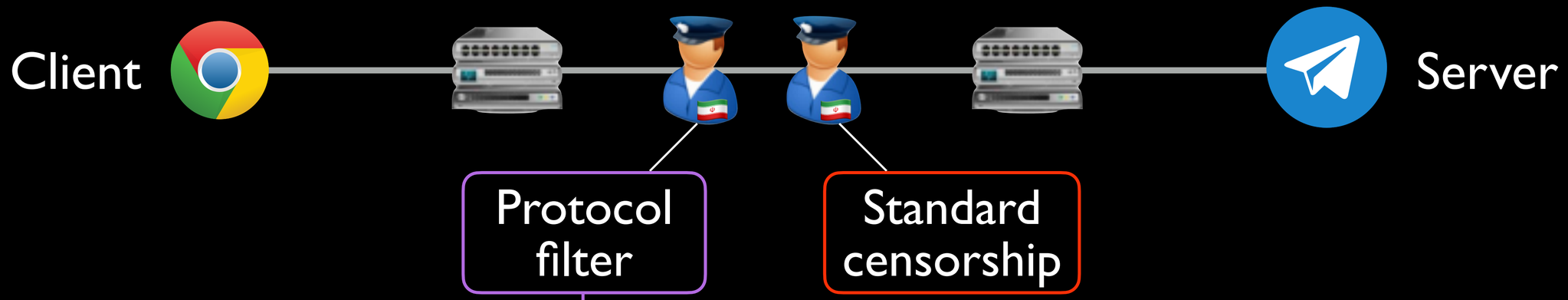
1. HTTP Request — Send a benign HTTP request with a broken checksum
2. FIN × 2 — Send 2 FIN packets before starting the handshake
3. ACK × 9 — Send 9 ACK packets to complete the handshake
4. S+A × 9 — Send 9 SYN/ACK packets with incorrect ack numbers

Iran's Protocol Filter



Geneva code and website geneva.cs.umd.edu

Iran's Protocol Filter



- Matches connections to protocol fingerprints
- Drops connections from unrecognized protocols
- Applies only to *some* destinations

Geneva code and website geneva.cs.umd.edu