TCP Fast Open

draft-cheng-tcpm-fastopen-00.txt

Yuchung Cheng, Jerry Chu, Sivasankar Radhakrishnan, Arvind Jain
{ycheng, hkchu, sivasankar, arvind}@google.com

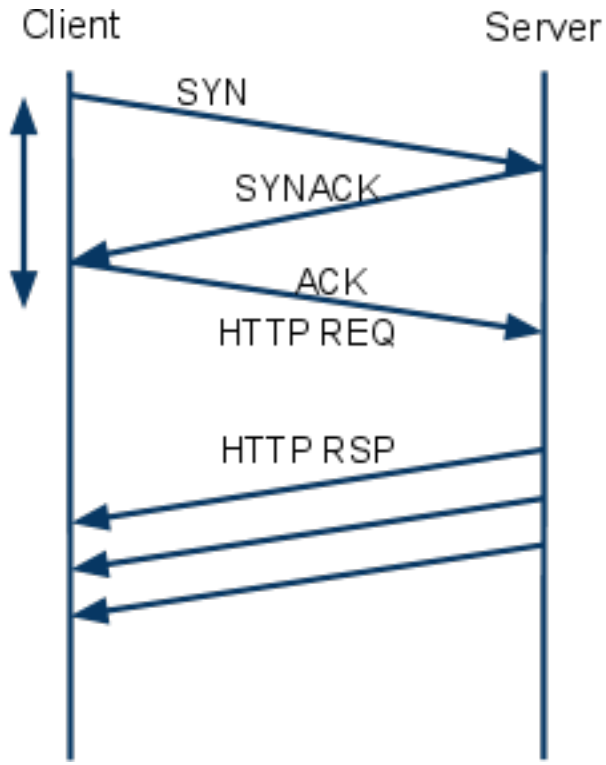
TCP fast open (TFO)

First HTTP request needs to do TCP 3-way handshake (3WHS)

- 1 RTT slowdown
- 35% Chrome HTTP requests
- www.ietf.org/proceedings/80/slides/tsvarea-0.pdf

Goal

- Data exchange (client and server) in 3WHS



Naive data-in-SYN?

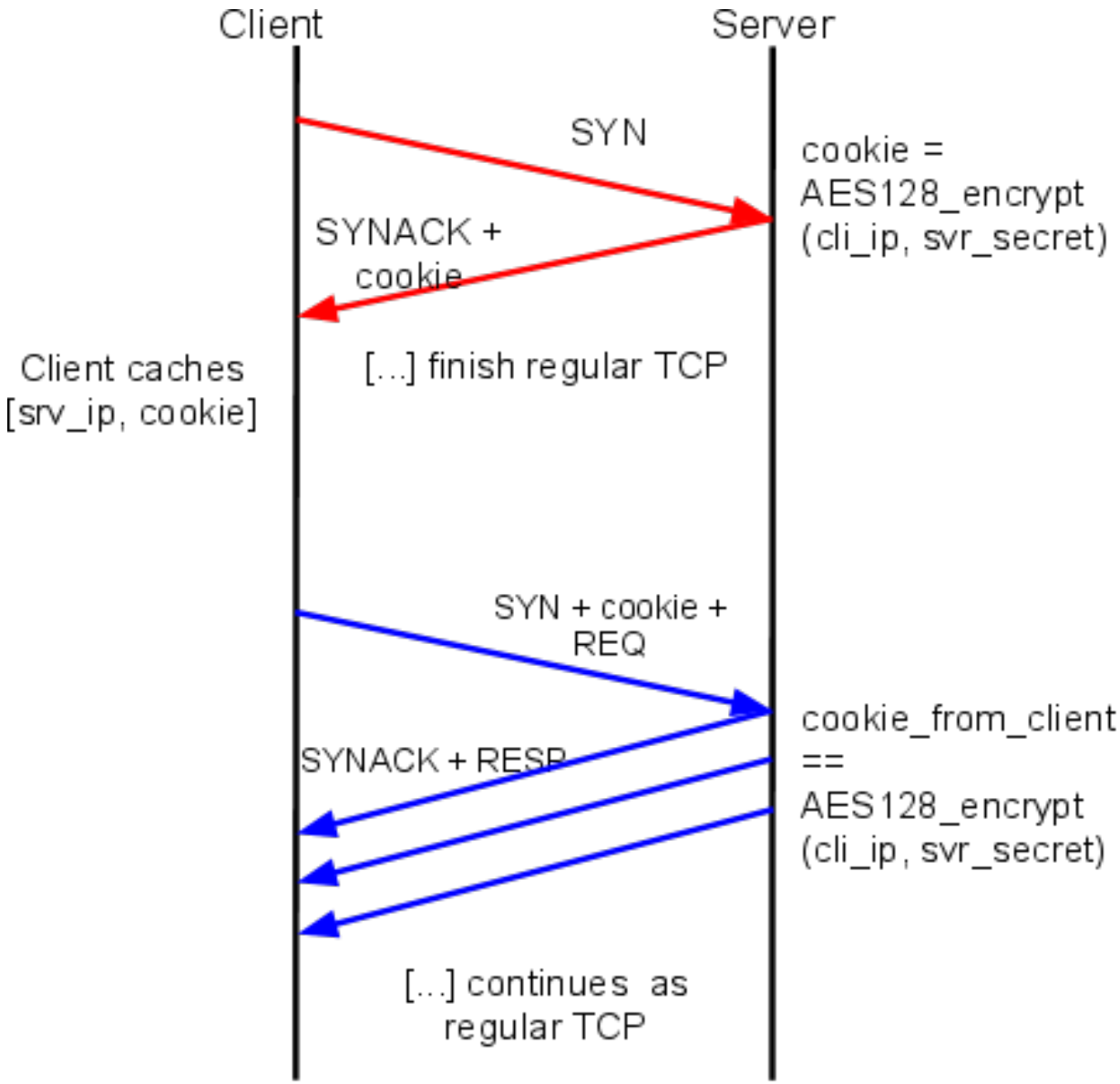
RFC793 TCP 3WHS

- Allows data in SYN
- Forbids processing data until 3WHS completes

Problems with data exchange in 3WHS?

1. Duplicate/old data from prior connections
2. Server resource exhaustion attack
3. Amplified reflection attack

TCP fast open design



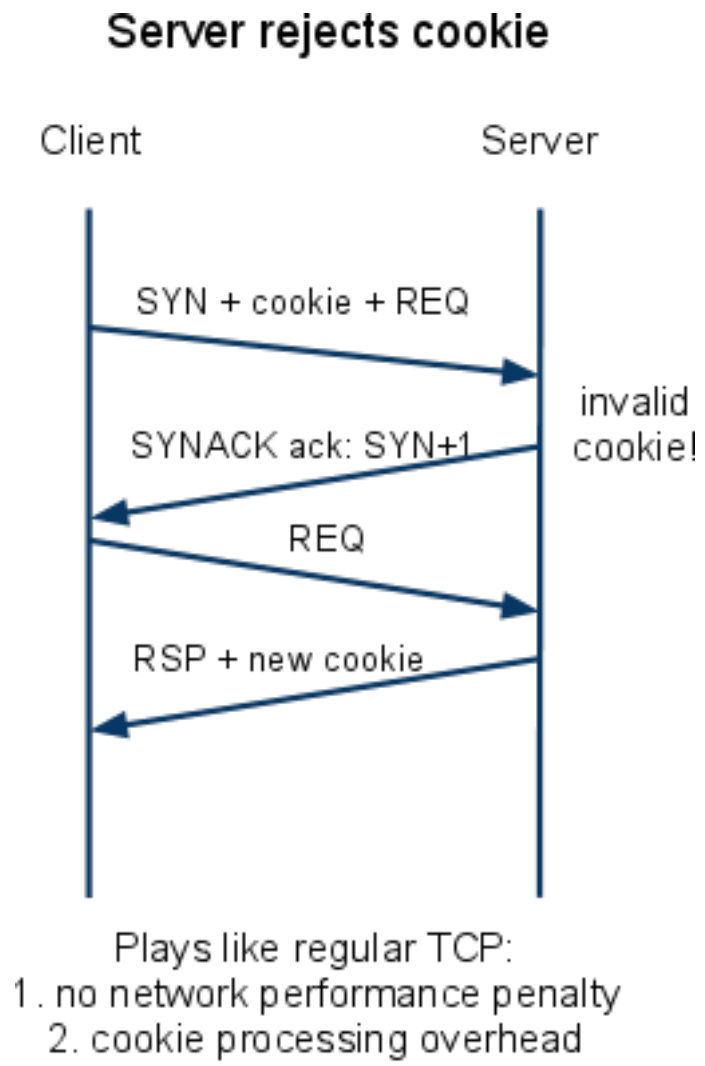
Default: off

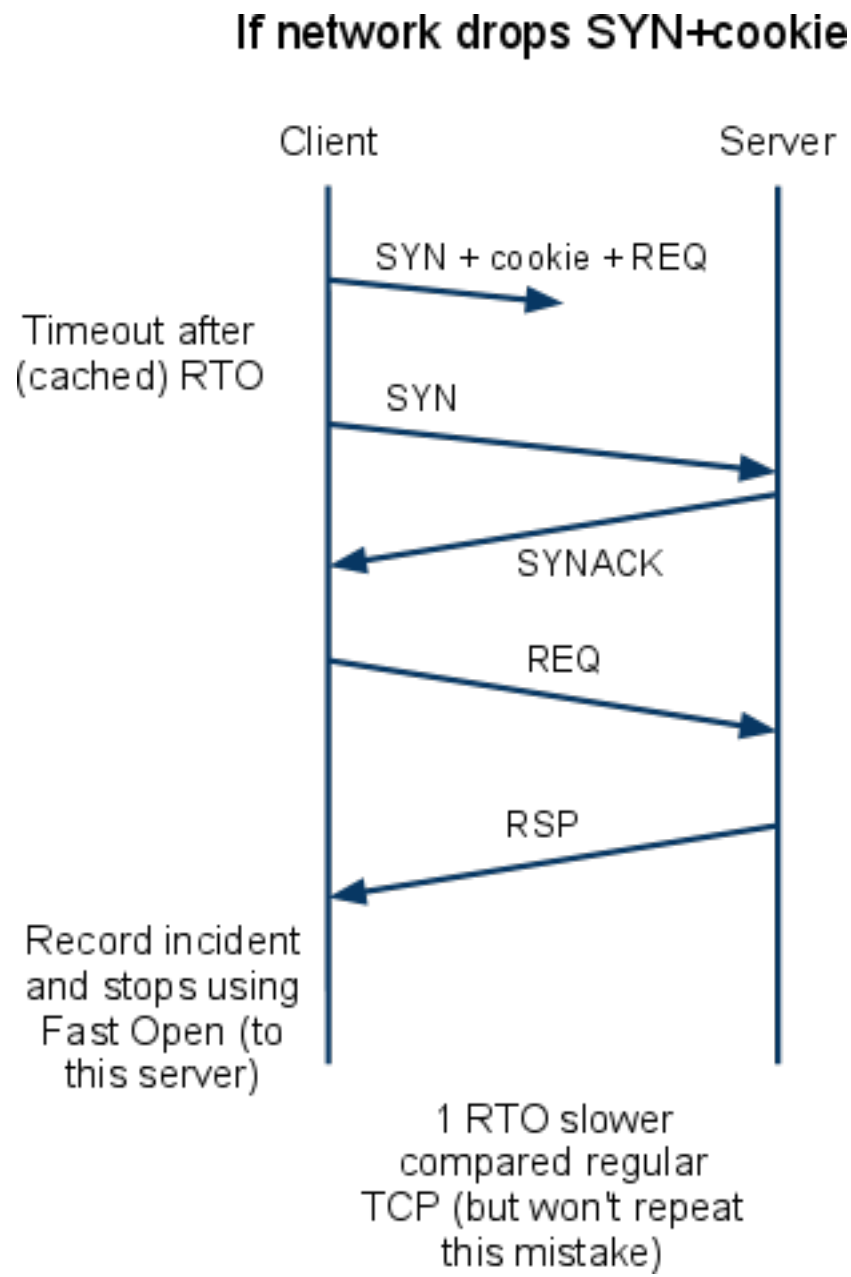
App that tolerates duplicate SYN-data does setsockopt(TFO)

Cookie:
server grants cookie as proof of IP ownership [we exchanged data before]

- *TCP option (64bits)*
- *MAC of client IP and a server secret*

Network dropping SYN with data or new options?





Mitigating security issues

Spoofed TFO SYN flood is still possible

- Obtain some (valid) cookies: DHCP / NAT, Moles
- Flood spoofed SYN w/ data/cookie

Server resource exhaustion attack

	TFO Syn-flood	Traditional Syn-flood
Goal	Exhaust data processing resources	Overflow syn queue
Requirement	1. Vantage point to flood spoofed SYNs 2. Obtain some validate cookies	1. Vantage point to flood spoofed SYNs
Mitigations	1. Limit max TFO connections in SYN_RCVD 2. Update server key every X min	RFC4987 (Syn-cookies)
RST in rsp to SYN-ACK	Fuel the damage	Lower the damage

Mitigating security issues (cont'd)

Amplified reflection attack

- 1 SYN+data to trigger multiple server packets to random victim
  - Disrupt/DOS victim's network
- Mitigations
  - Limit TFO connections in SYN_RCVD
  - Update the server key every X min
- Extra mitigations for server farms for extreme cases
  - Respond only SYN/ACK during 3WHS
  - Server can still process request one RTT earlier

Related work

	TCP Fast Open (TFO)	TCPCT (RFC6013)	T/TCP (RFC1644)
Designed	Cheng et al., 2010	Bill Simpson, 2009	Bob Braden, 1994
Goal	Data exchange in 3WHS	1. Defend any SYN flood 2. Quick conn setup/ tear-down	1. At-most-once semantic 2. Quick conn setup / teardown
Motivating Application	Web	DNSSEC	Transactional one packet RPC
Additional States	client caches server cookies	no	per-IP counter at client/server
Implementation	(private) Linux and Chrome patch	Partially implemented in Linux	?

Conclusion

TCP Fast Open

- Data exchange in TCP handshake
  - 1 RTT savings on 35% of HTTP requests
- Cookie to mitigate security vulnerabilities

Implementation

- Linux (private patch) and Chrome
  - Tested TFO on live Internet connections
    Worked on Comcast, ATT, etc.
  - web server application: only setsockopt(TFO)

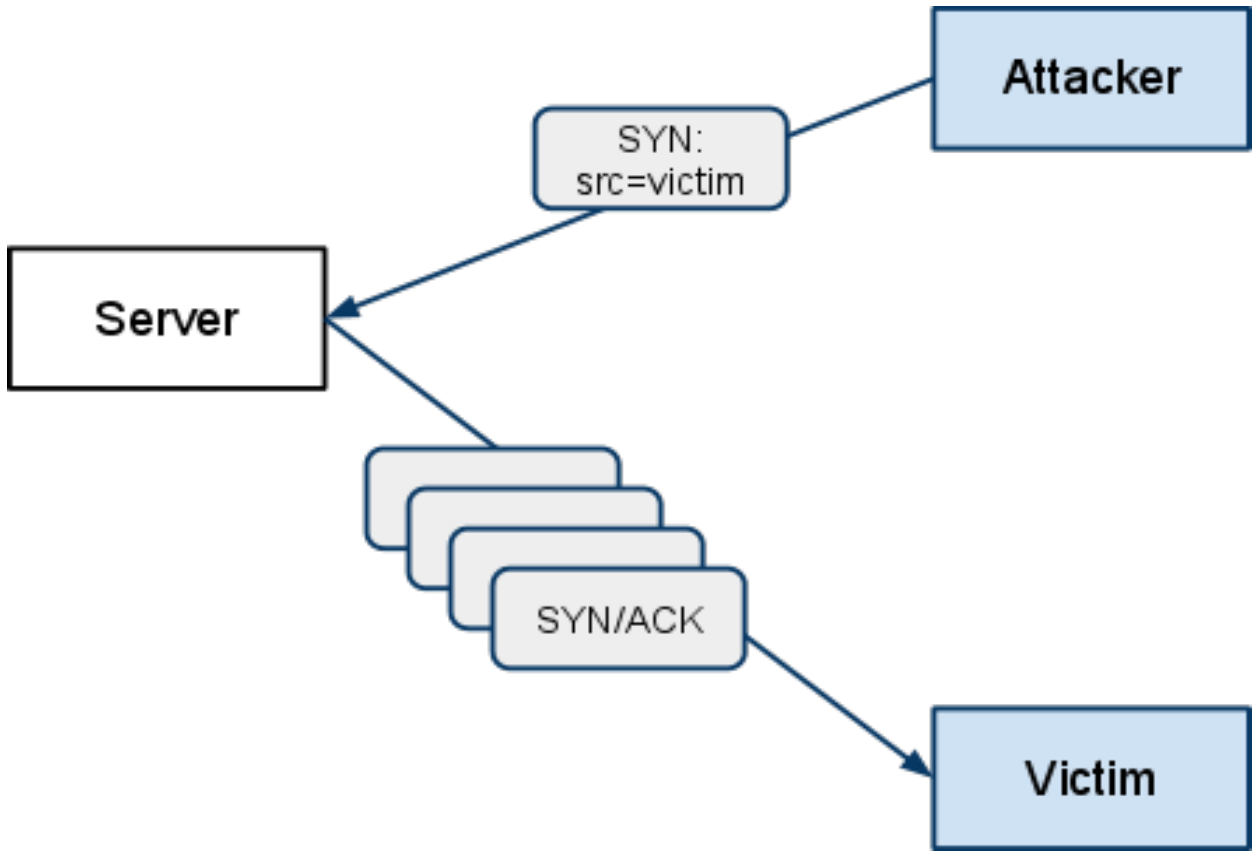
Questions/comments?

Alternate design: one-time cookie

1. Generation:
   cookie = AES_128_encrypt(IP | counter, key)
   counter += 1
2. Validation:
   IP_c | counter_c = AES_128_decrypt(cookie, key)
   IP_c == IP in SYN
3.

	TFO	One-time cookie
Server states	O(1) (key)	O(n): n #cookies small scalar factor
Client states	one cookie per svr IP	one cookie per connection
Cookie Size	32bits - 128bits	128bits

Amplified reflection attack



1 (small) SYN for ${init_cwnd} data packets