

Forward Secrecy in TLS

A Systematic Study

Nick Sullivan (@grittygrease)

Filippo Valsorda (@filosottile)



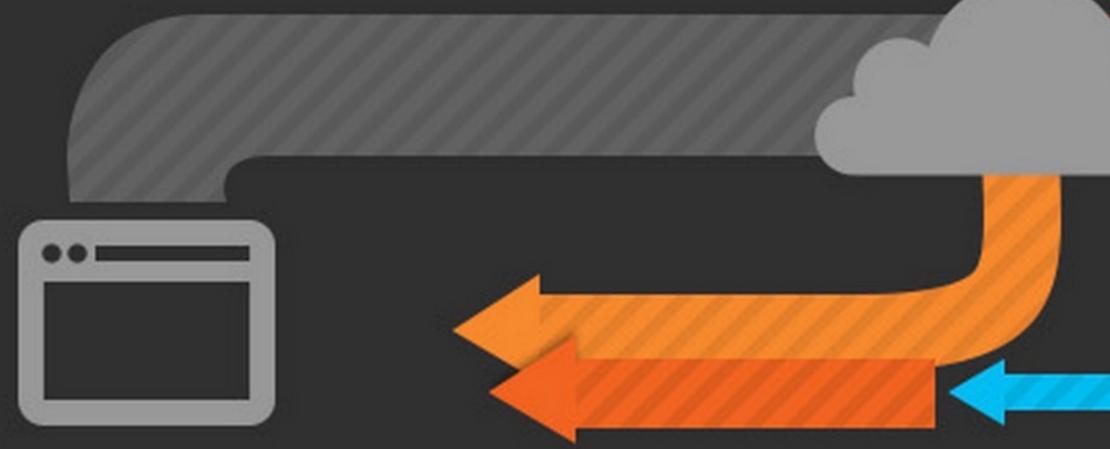
Who am I?

- Cryptography Engineer
- Focused on taking cryptographic concepts and bringing them to the world at scale
- Apple FairPlay: Protecting keys in hostile locations
- Cryptography team at CloudFlare: bringing the security of the Internet giants to everyone



WARNING:

This is a practical talk about the Internet industry.





CloudFlare Reverse Proxy

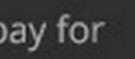
...



×

Bandwidth saved by CloudFlare

Bandwidth you pay for



CloudFlare Network Map

North America

CURRENT

Ashburn, Virginia Atlanta, Georgia Chicago, Illinois Dallas, Texas Los Angeles, California Miami, Florida Minneapolis, Minnesota Montréal, Québec Newark, New Jersey Phoenix, Arizona San Jose, California Seattle, Washington Toronto, Ontario Vancouver, British Columbia

UPCOMING

Boston, Massachusetts Denver, Colorado Honolulu, Hawaii Kansas City, Missouri Las Vegas, Nevada Mexico City, Mexico Panama City, Panama Salt Lake City, Utah



South America

CURRENT

Buenos Aires, Argentina Lima, Peru Medellín, Colombia Valparaíso, Chile São Paulo, Brazil

UPCOMING Rio de Janeiro, Brazil Quito, Ecuador

Willemstad, Curaçao

Europe

CURRENT

Amsterdam, Netherlands Berlin, Germany Bucharest, Romania Copenhagen, Denmark Dublin, Ireland Düsseldorf, Germany Frankfurt, Germany Hamburg, Germany London, England Madrid, Spain Manchester, England Marseille, France Milan, Italy Oslo, Norway Paris, France Prague, Czech Republic Sofia, Bulgaria Stockholm, Sweden Vienna, Austria Warsaw, Poland Zürich, Switzerland

UPCOMING

Athens, Greece Barcelona, Spain Brussels, Belgium Budapest, Hungary Helsinki, Finland Istanbul, Turkey Kiev, Ukraine Lisbon, Portugal Moscow, Russia Munich, Germany



Planned



Asia

CURRENT

Chennai, India Hong Kong, Hong Kong Kuala Lumpur, Malaysia Mumbai, India New Delhi, India Osaka, Japan Seoul, South Korea Singapore, Singapore Tokyo, Japan

UPCOMING

Taipei, Taiwan

Bangkok, Thailand Colombo, Sri Lanka Hanoi, Vietnam Jakarta, Indonesia Manila, Philippines Penang, Malaysia Senai, Malaysia

China

- CURRENT
- Chengdu, China Dongguan, China Foshan, China Fuzhou, China Hangzhou, China Hengyang, China Guangzhou, China Jiaxing, China Langfang, China

Louyang, China Nanning, China Qingdao, China Shenyang, China Shijiazhuang, China Tianjin, China Xi'an, China Zhengzhou, China

Oceania

CURRENT

Auckland, New Zealand Melbourne, Victoria, Austrailia Sydney, New South Wales, Austrailia

UPCOMING

Perth, Australia Brisbane, Australia

Africa CURRENT

C

Cairo, Egypt Johannesburg, South Africa Mombasa, Kenya

UPCOMING

Lagos, Nigeria Luanda, Angola Tunis, Tunisia

CURRENT

Doha, Qatar Dubai, UAE Kuwait City, Kuwait Muscat, Oman

UPCOMING

Riyadh, Saudi Arabia Tel Aviv, Israel

Middle East

Application Layer

- DNS
- HTTP(S)

- 5-7% of web requests go through CloudFlare's network
- We see almost every web user daily
- Low latency is most important feature



Key contributions

- Keyless SSL: Terminating HTTPS without the private key
- Universal DNSSEC: Digital signatures in the DNS with ECDSA
- ChaCha20/Poly1305: djb crypto is not just for Google anymore
- **Deprecation of RC4**: First to drop the creaky cipher
- Origin CA: Free certificates for services behind CloudFlare
- Universal SSL: Free HTTPS for all sites, ECDSA certificates
- Global session resumption: One fewer roundtrip even on new servers



Key contributions

- Keyless SSL: Terminating HTTPS without the private key
- Universal DNSSEC: Digital signatures in the DNS with ECDSA
- ChaCha20/Poly1305: djb crypto is not just for Google anymore
- Deprecation of RC4: First to drop the creaky cipher
- Origin CA: Free certificates for services behind CloudFlare
- Universal SSL: Free HTTPS for all sites, ECDSA certificates
- Global session resumption: One fewer roundtrip even on new servers



Forward Secrecy in TLS 1.2

A weak definition

 Compromising a long-term key does not allow an attacker compromise previous connections.



Threat models

This is the Internet, all wires are tapped

1.Attackers with access to transcript of historical communications

2. Attackers who can place themselves in a MiTM position



We explore what happens when attackers gain access to different keys

TLS 1.2 RSA key exchange Client

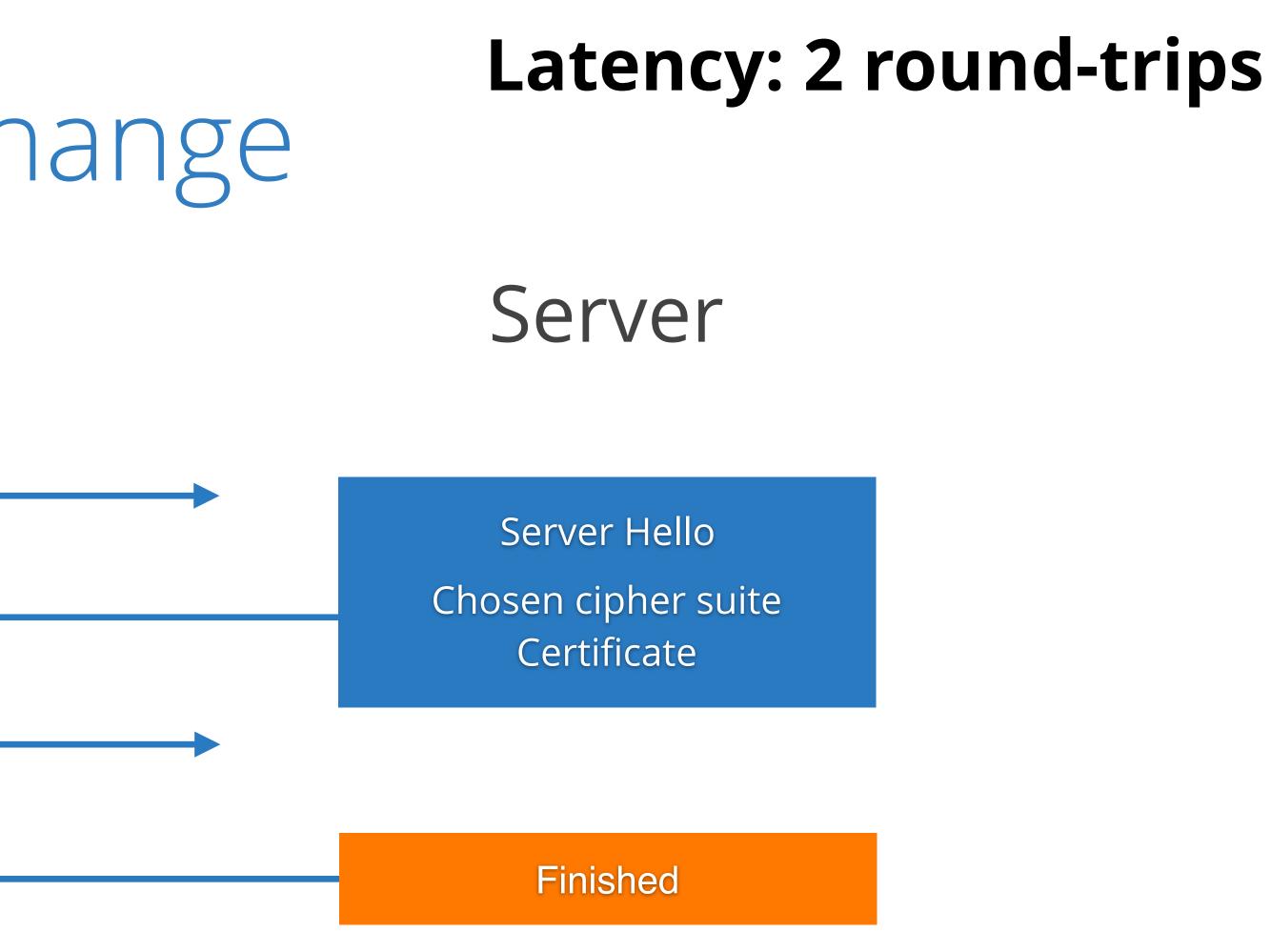
Client Hello Supported cipher suites

ClientKeyExchange

Finished

pre_master_secret > master_secret > traffic keys





ClientKeyExchange = RSA_encrypt(Certificate_pk, pre_master_secret)



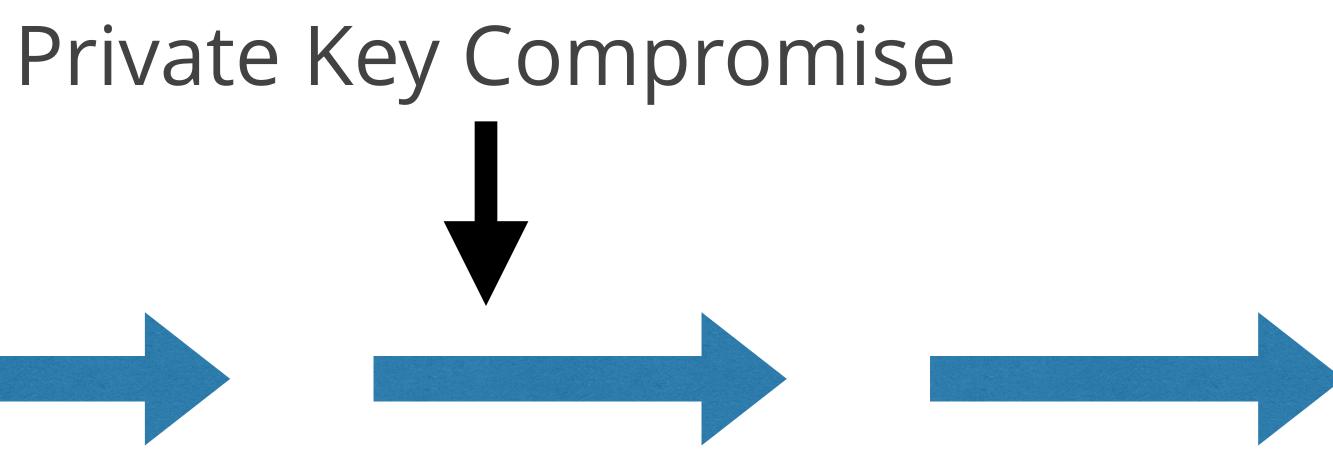


- RSA private key
- Session key



12





Private Key Compromise Modifiable by active attacker

Decryptable by passive attacker

Session Compromise

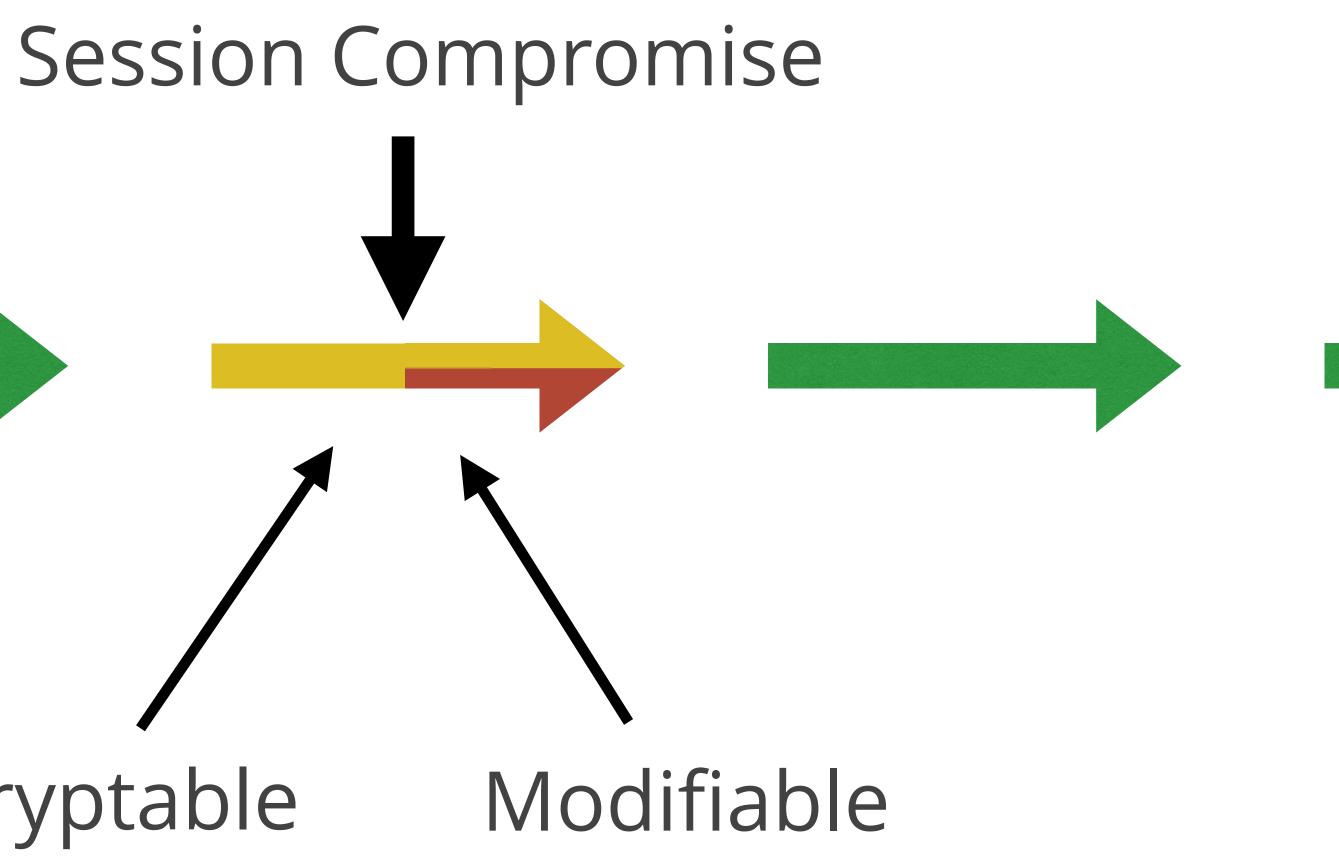








Decryptable



Latency: 2 round-trips TLS 1.2 (EC)DHE key exchange Client Server

Client Hello Supported cipher suites

ClientKeyExchange

Finished

ServerKeyExchange = Sign(Certificate_pk, (EC)DH public key share) ClientKeyExchange = (EC)DH public key share

(EC)DH derivation > pre_master_secret > master_secret > traffic keys



Server Hello Chosen cipher suite ServerKeyExchange Signature Certificate

Finished



(EC)DHE: the E is for ephemeral

- OpenSSL historically did not always do this
- SSL_OP_SINGLE_DH_USE (required as of 2016)
- We assume it is ephemeral





- Certificate private key
- Session key

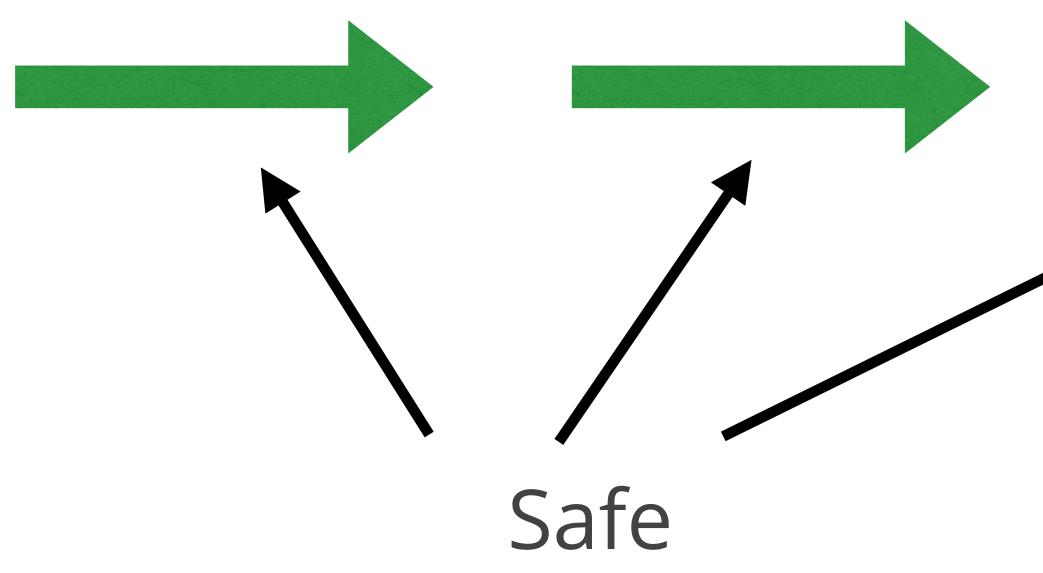


Private Key Compromise





Private Key Compromise



Modifiable Not decryptable by passive attacker



Session Key Compromise

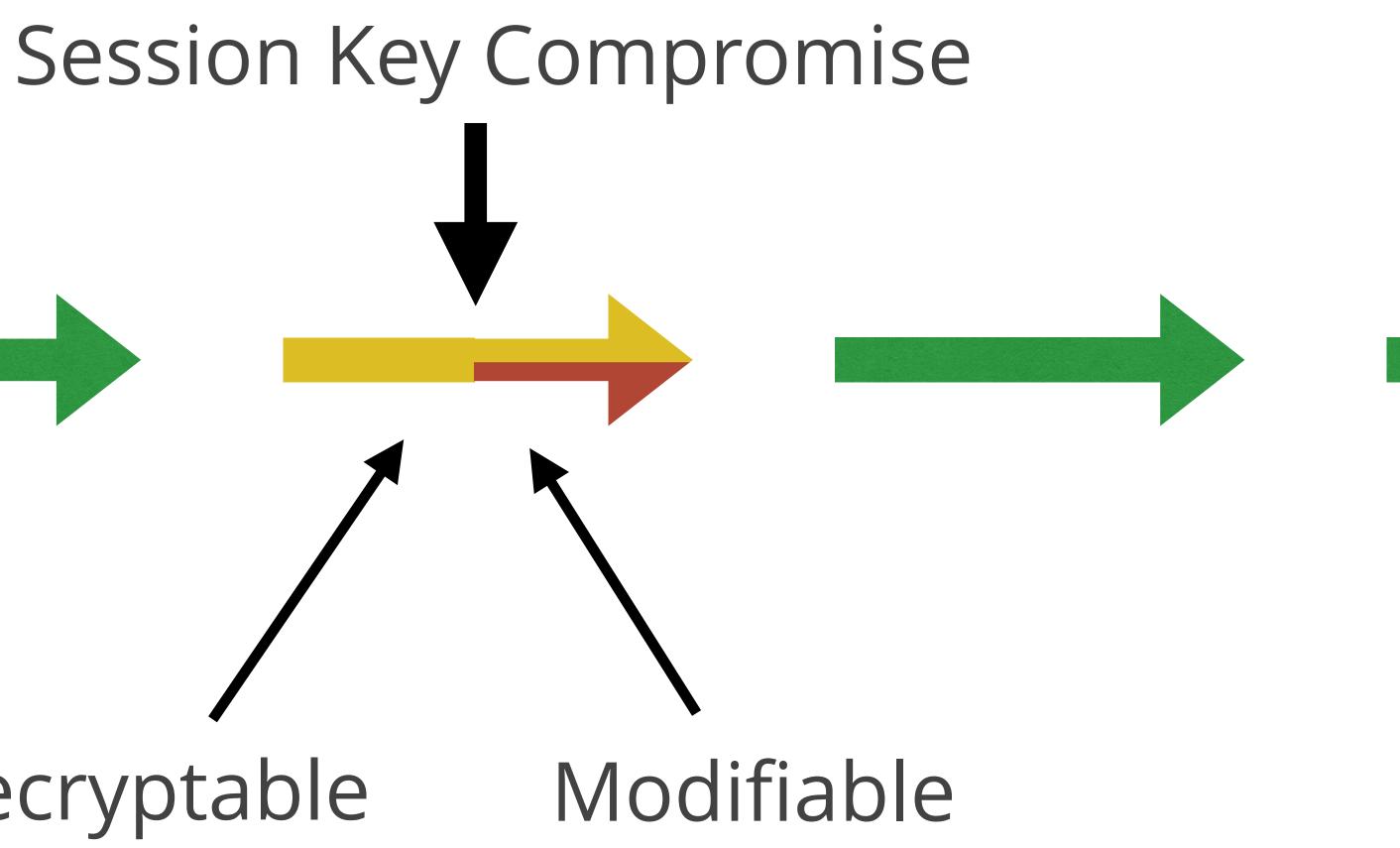








Decryptable



Key contributions

- Keyless SSL: Terminating HTTPS without the private key
- Universal DNSSEC: Digital signatures in the DNS with ECDSA
- ChaCha20/Poly1305: djb crypto is not just for Google anymore
- Deprecation of RC4: First to drop the creaky cipher
- Origin CA: Free certificates for services behind CloudFlare
- Universal SSL: Free HTTPS for all sites, ECDSA certificates \checkmark
- Global session resumption: One fewer roundtrip even on new servers

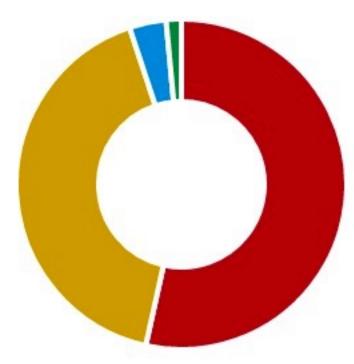


History of forward secrecy support

E)

October 2013

Forward Secrecy



- Not supported 87,982 54.0%
- Some FS suites enabled 68,195 41.8%
- Used with modern browsers 5,916 3.6%
- Used with most browsers 937 0.6%



June 2016

Forward Secrecy



- Not supported
- **27,730** 19.6%
- Some FS suites enabled
- **40,253** 28.5%
- Used with modern browsers
- **41,903** 29.6%
- Used with most browsers

31,501 22.3% + 0.1%

E

But wait...

- What about session resumption?
- Compromise between performance and secrecy



TLS 1.2 Session Resumption Client

Client Hello

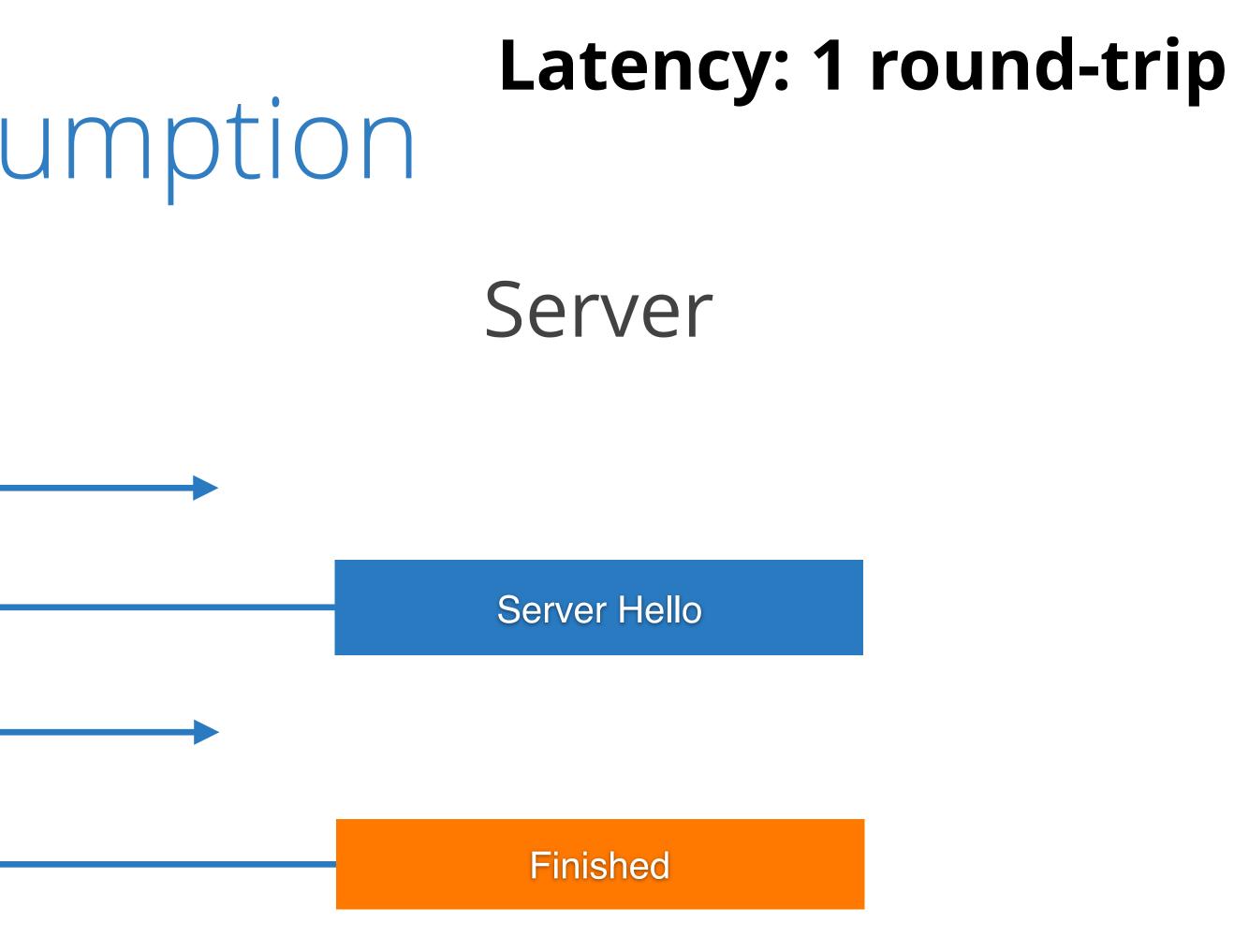
Session ID

Finished

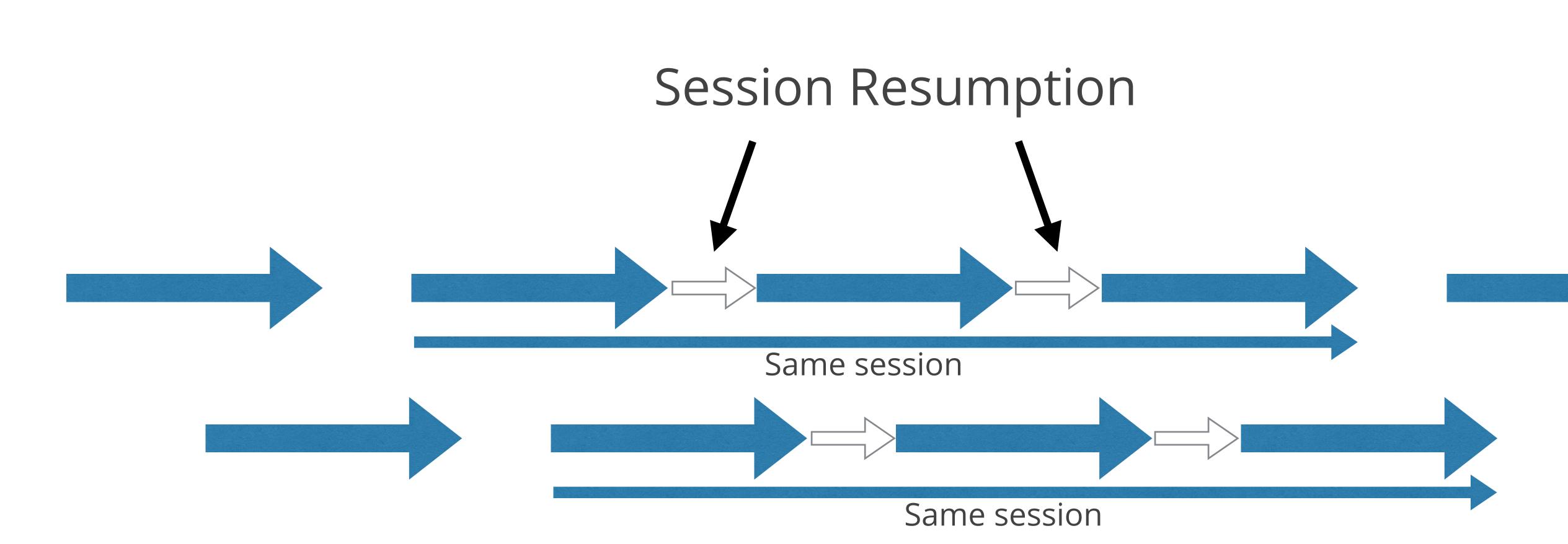
Session = master_secret

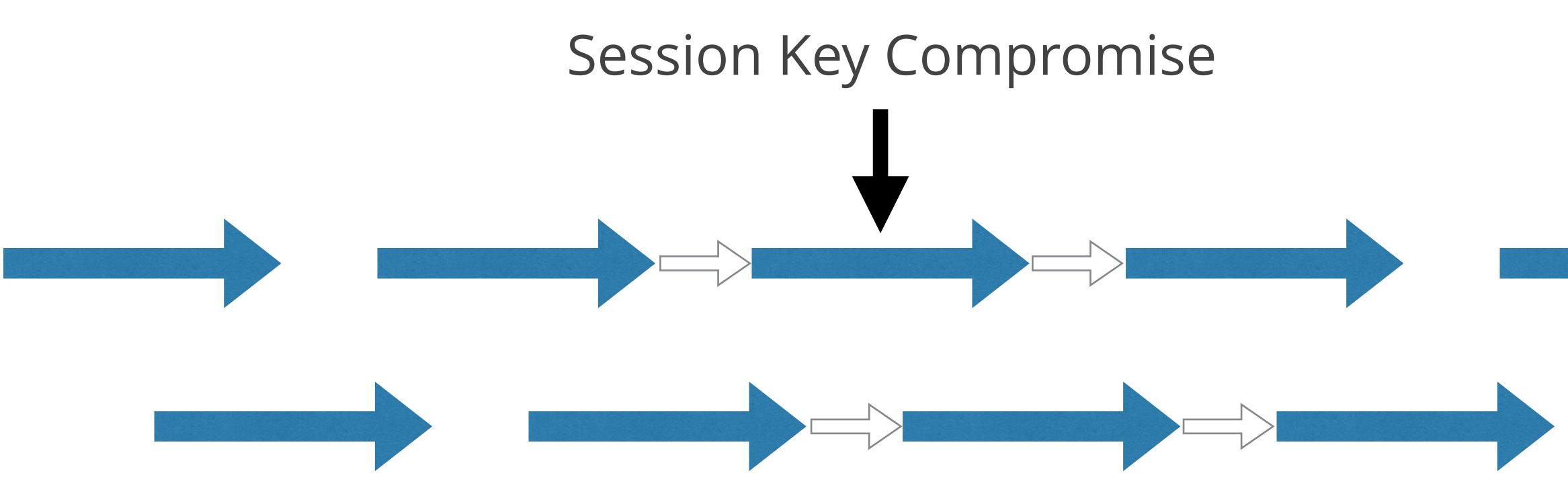
Sessions are saved server-side, indexed by Session ID.

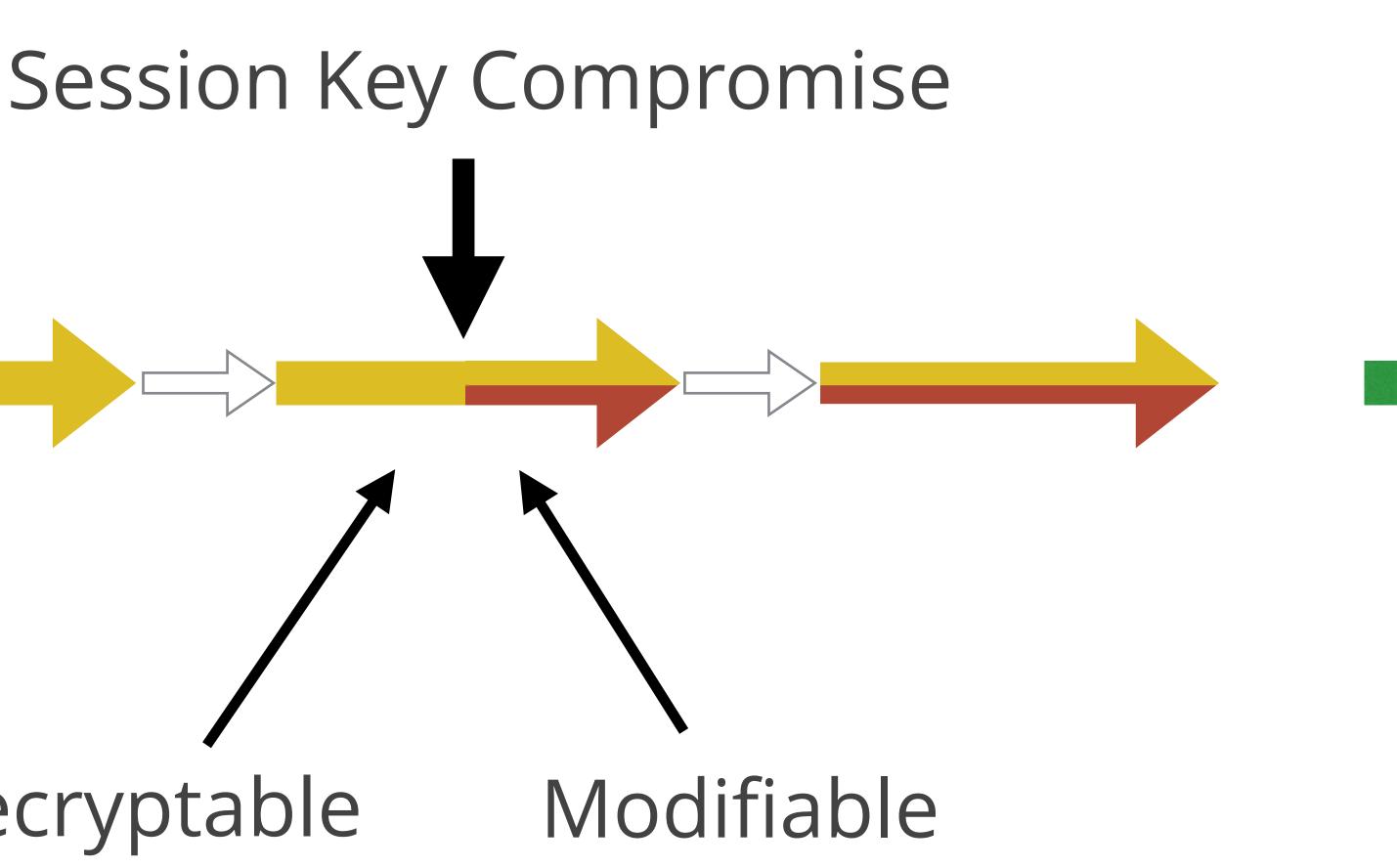




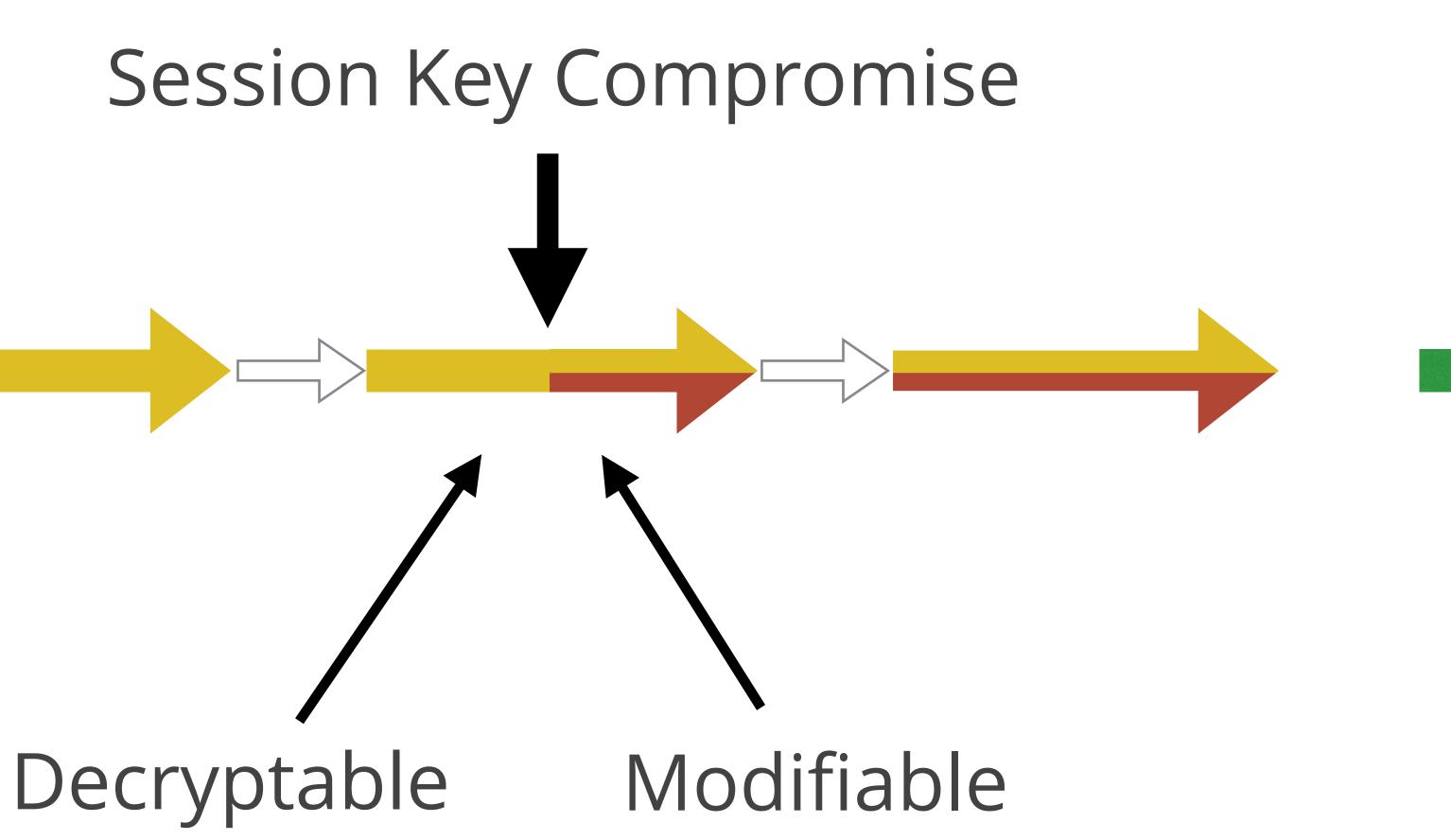


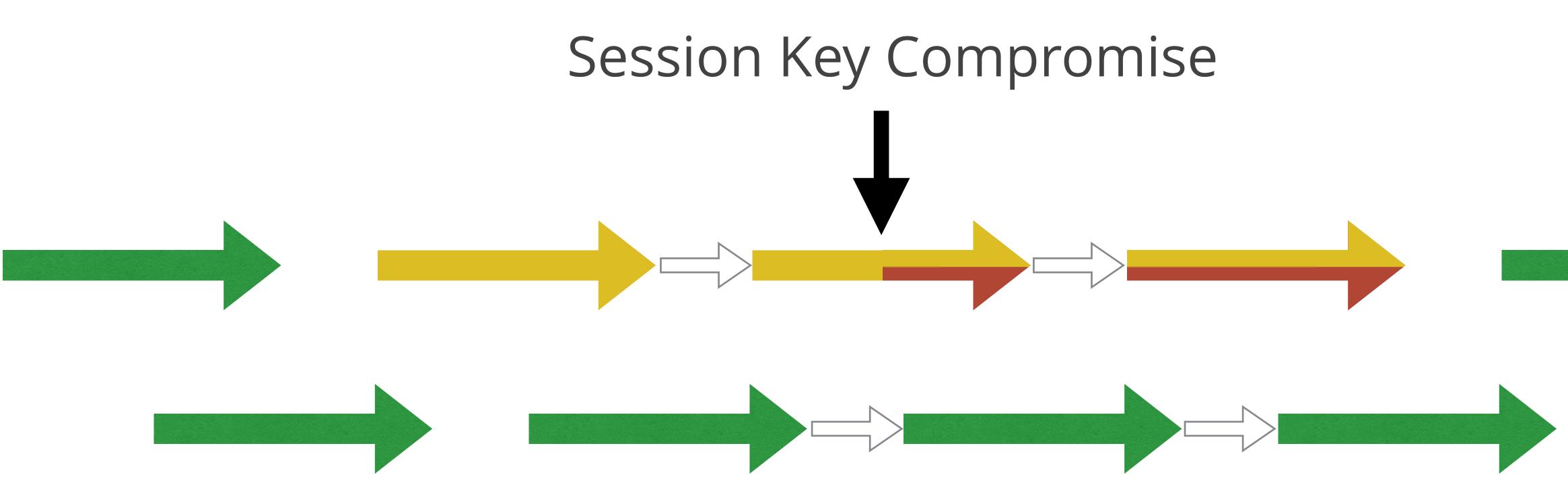












TLS 1.2 Session Tickets RFC5077 Client

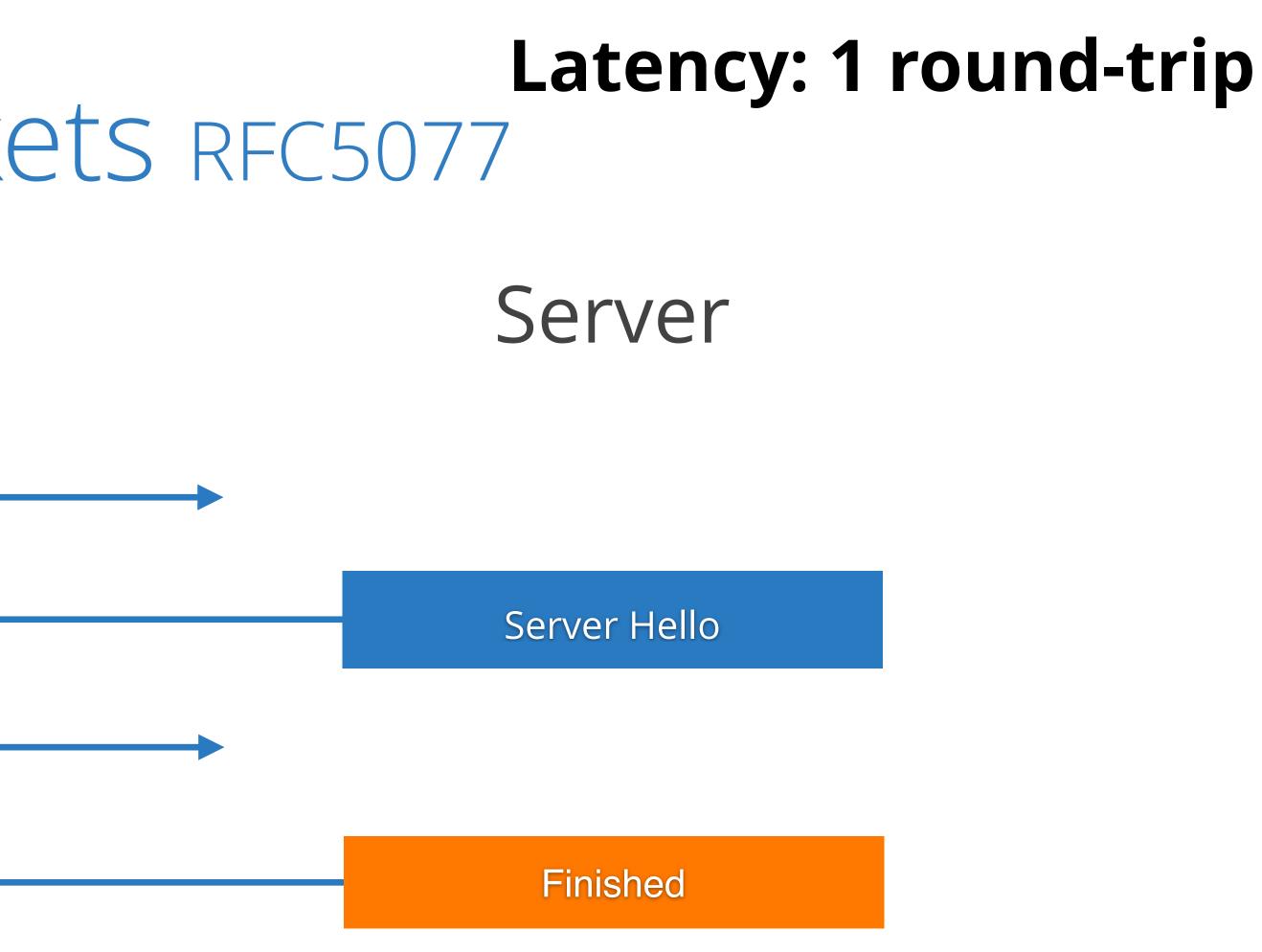
Client Hello Session Ticket

Finished

Session Ticket = encrypt(ticket_key, Session)

The server does not store Sessions, holds the ticket_key.





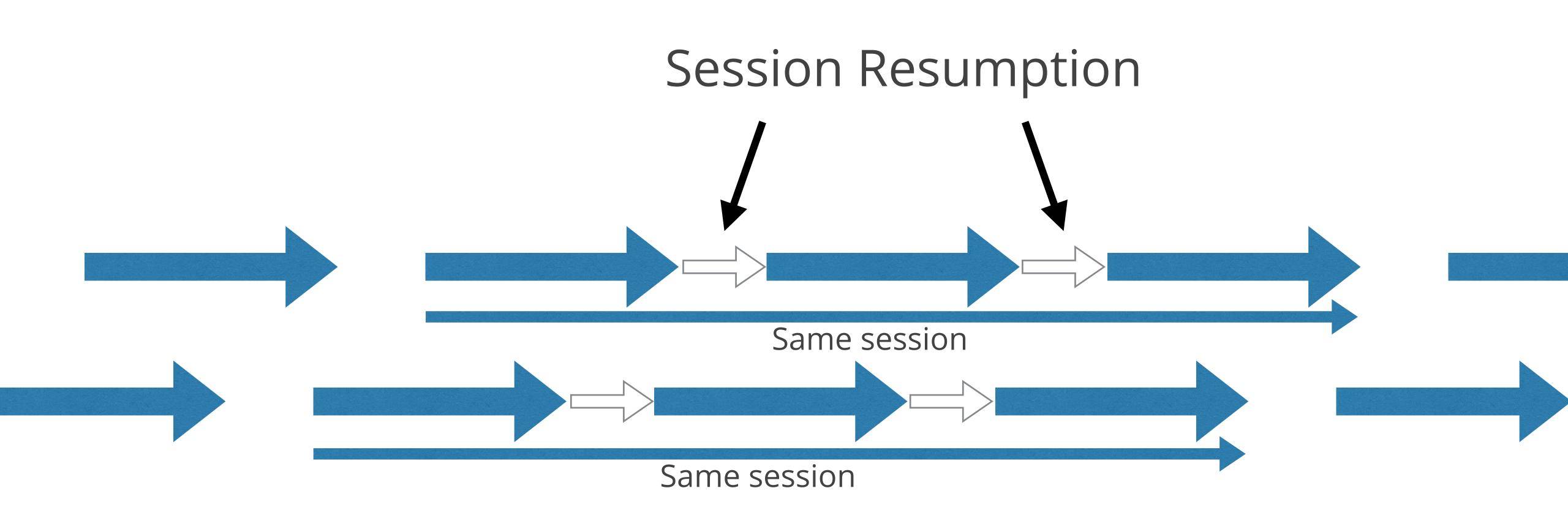


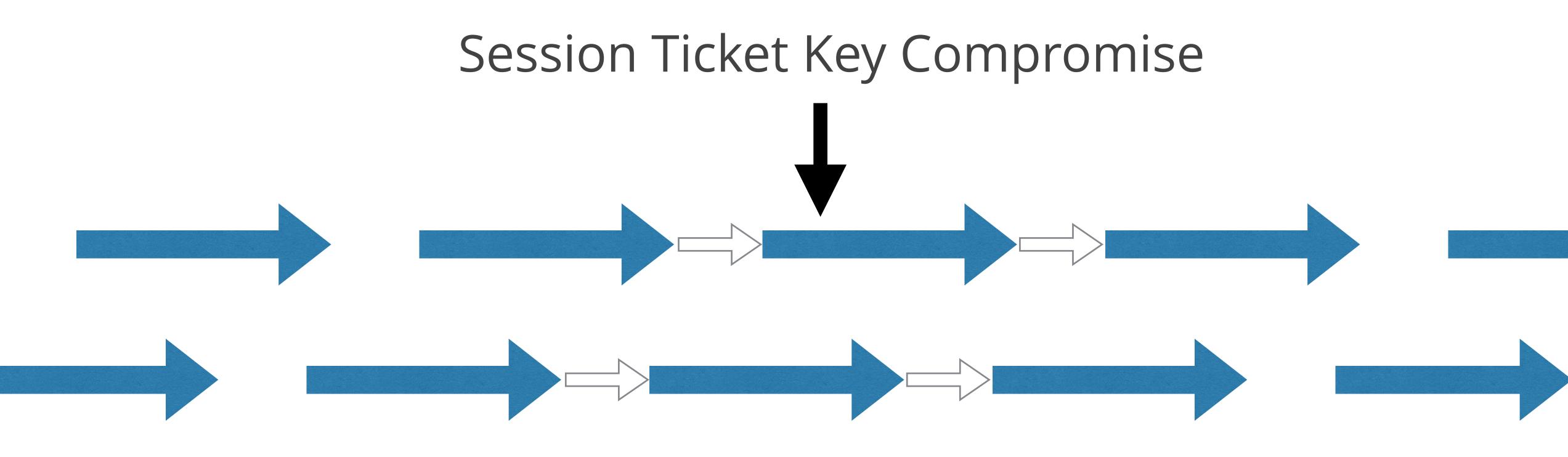
Three keys

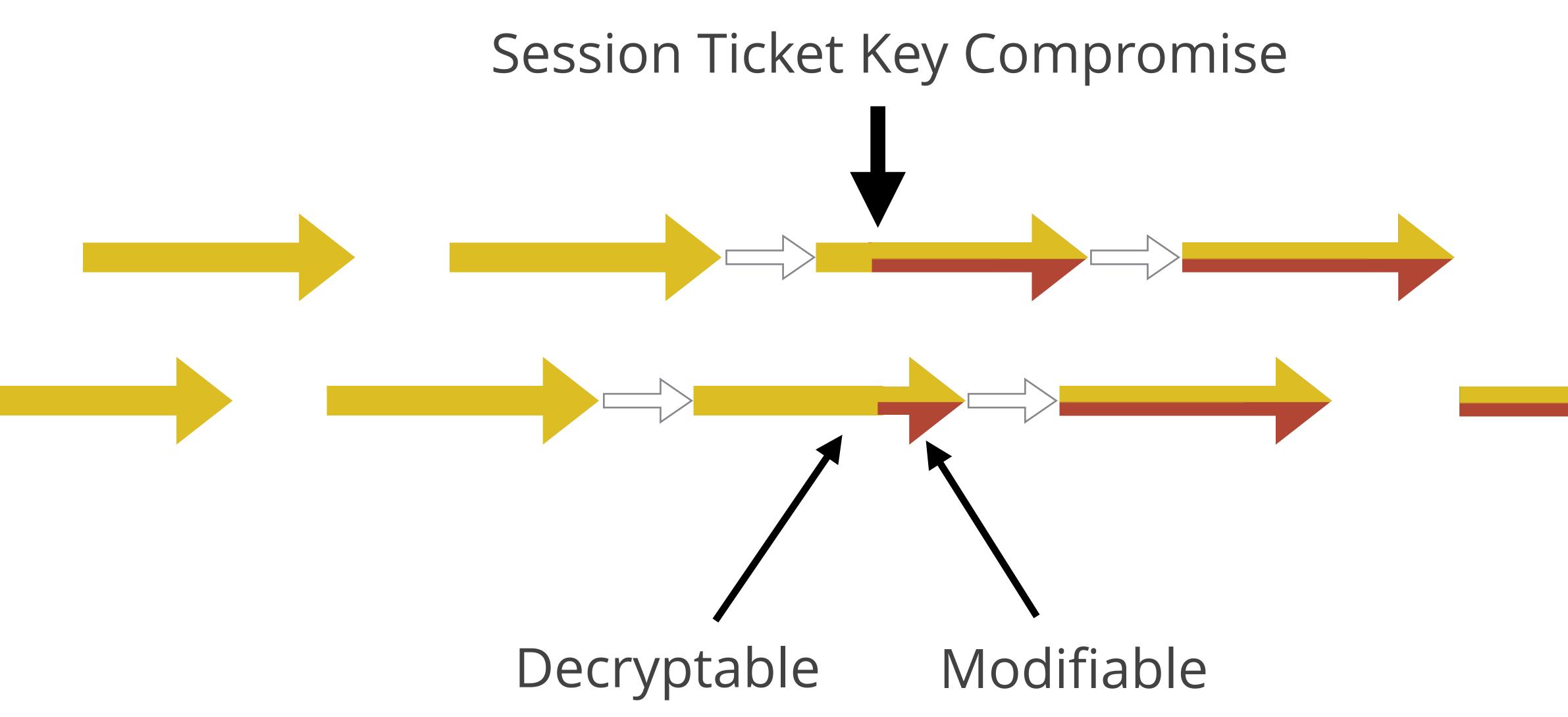
- Certificate private key
- Session key
- Session ticket encryption key



33









Session resumption + load balancing

- OpenSSL: session tickets are generated when the server starts
 - Does not let you resume across servers
 - Some servers run for a long time •

- Session resumption
 - How long are sessions cached? Is there a shared cache between machines?
 - Target for measurement



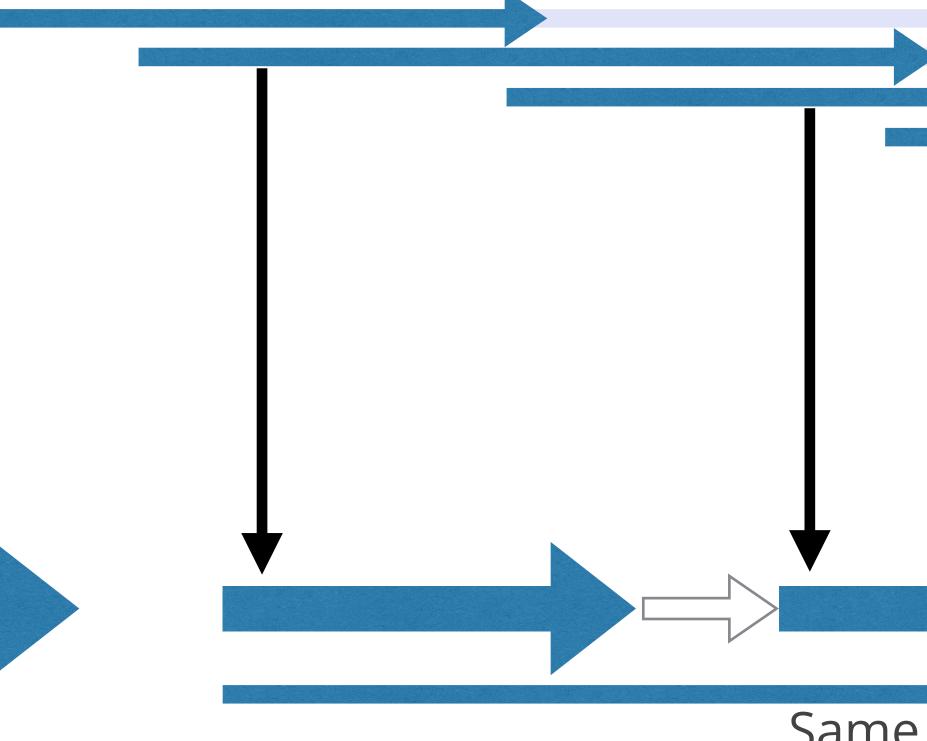
Session resumption + load balancing

- Distributed session ticket keys
 - Distribute session ticket encryption keys globally

 - Target for measurement



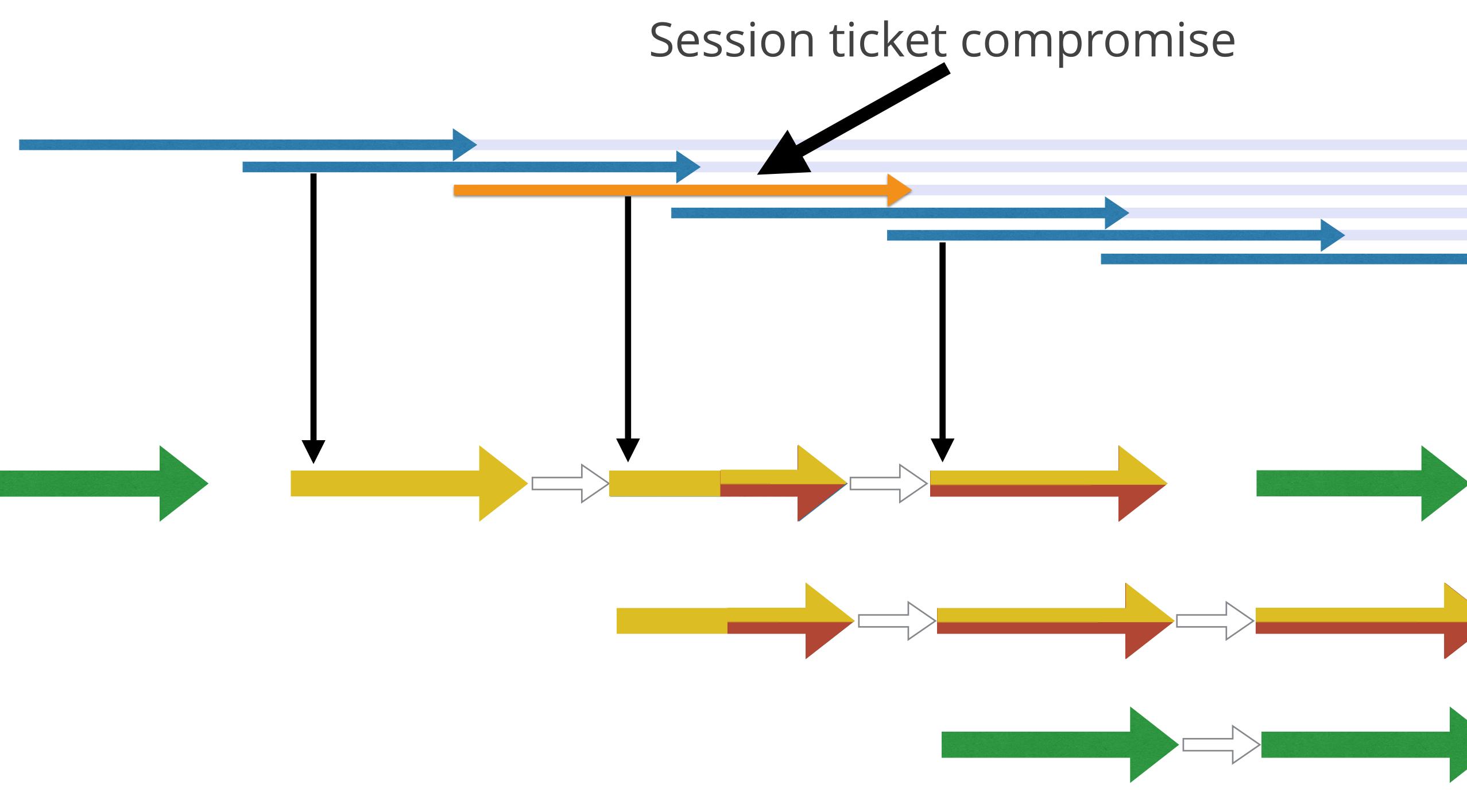
CloudFlare: Rotate current session ticket key hourly, resume with old keys for 18 hours





Session ticket lifetime Same session Same session





Key contributions

- **Keyless SSL**: Terminating HTTPS without the private key
- Universal DNSSEC: Digital signatures in the DNS with ECDSA
- ChaCha20/Poly1305: djb crypto is not just for Google anymore
- **Deprecation of RC4**: First to drop the creaky cipher
- **Origin CA**: Free certificates for services behind CloudFlare
- **Universal SSL**: Free HTTPS for all sites, ECDSA certificates $\sqrt{}$
- Global session resumption: One fewer roundtrip even on new servers ???





Not so "perfect"

- Sessions persist across resumption, compromise of one session compromises all
- the session ticket key



Session ticket compromise compromises all session keys encrypted with

TLS 1.3

Locking down the forward secrecy story



TLS 1.3

Client

Client Hello

Supported cipher suites Key share

Finished



Latency: 1 round-trip

Server

Server Hello

Chosen cipher suite Key share

Certificate & signature

Finished

NewSessionTicket



Session resumption improvements

- Resumed sessions use new traffic keys derived from previous master secret and handshake
- PSK mode: symmetric key ratchet
- PSK-(EC)DHE mode: public key ratchet
- Option to sign resumption handshakes
- Mandatory maximum session ticket lifetime (7 days)

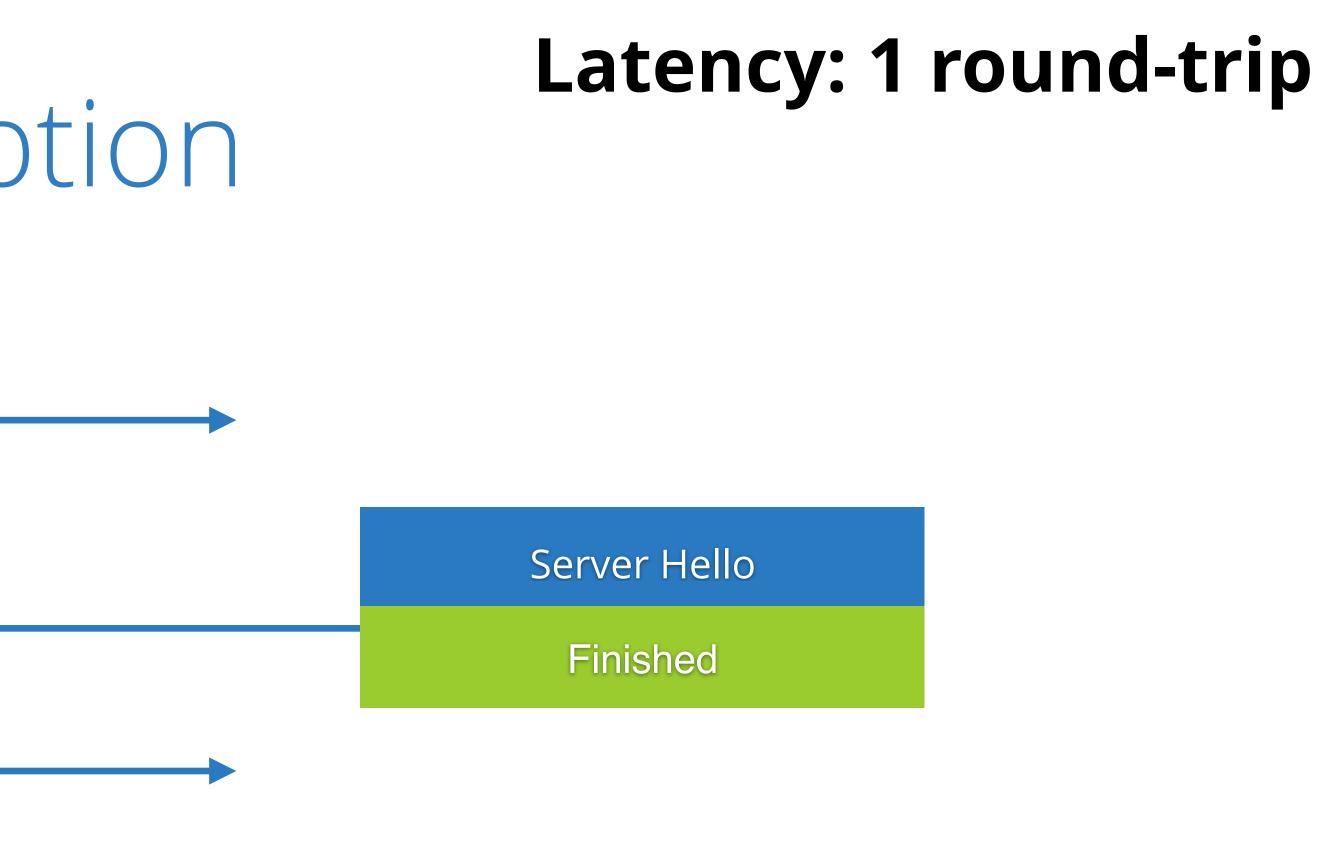


TLS 1.3 PSK Resumption

Client Hello Session Ticket (PSK)

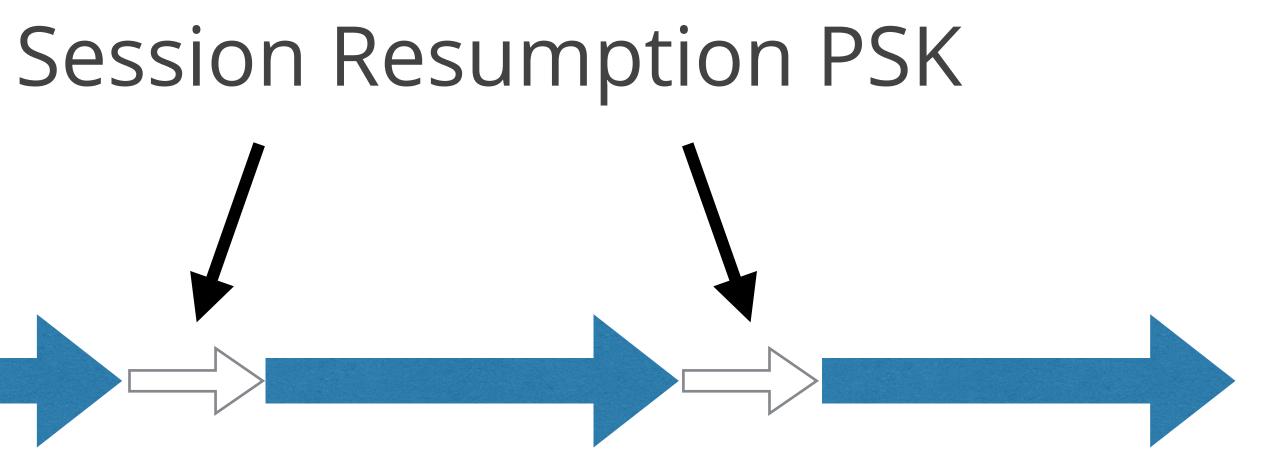
Finished

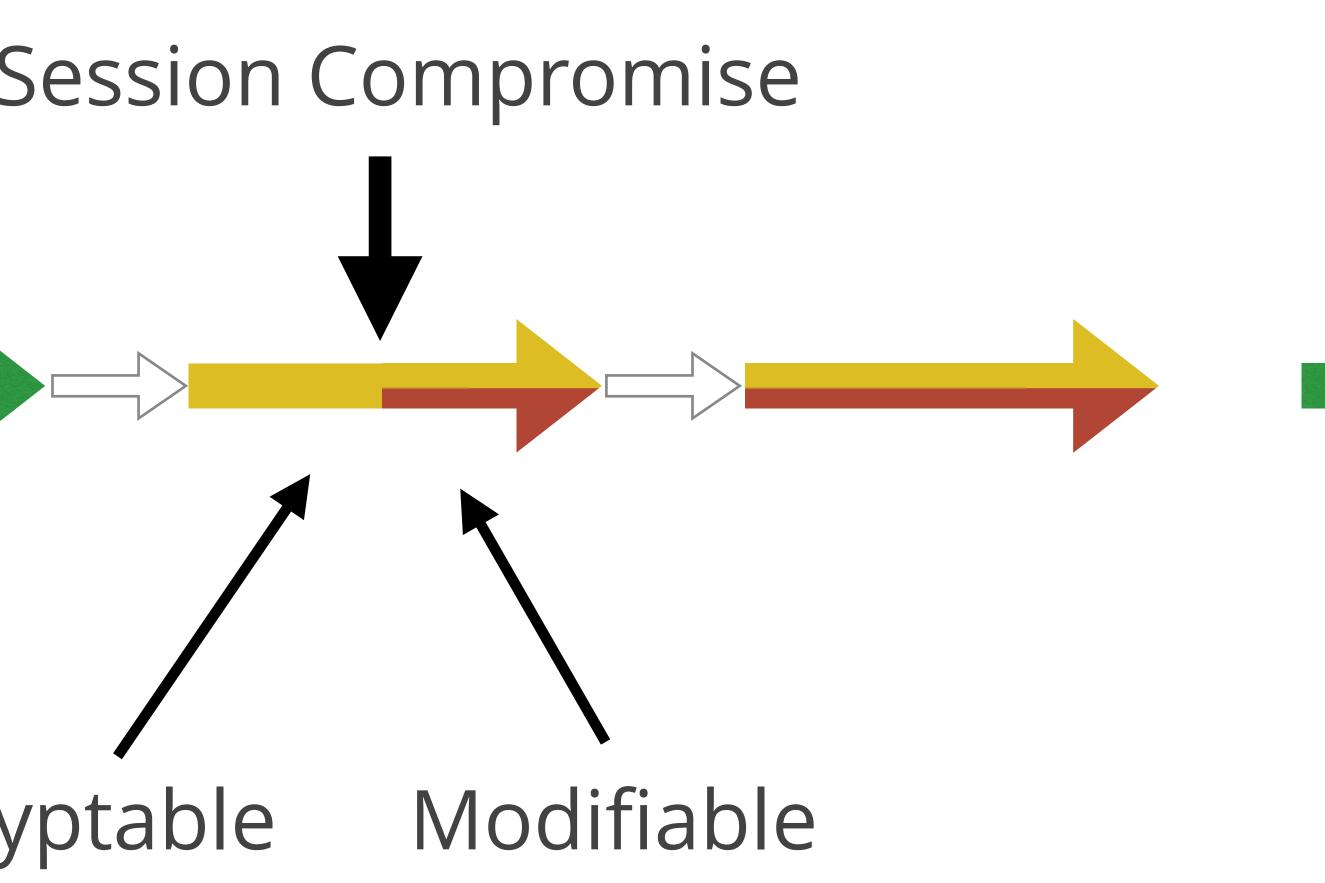


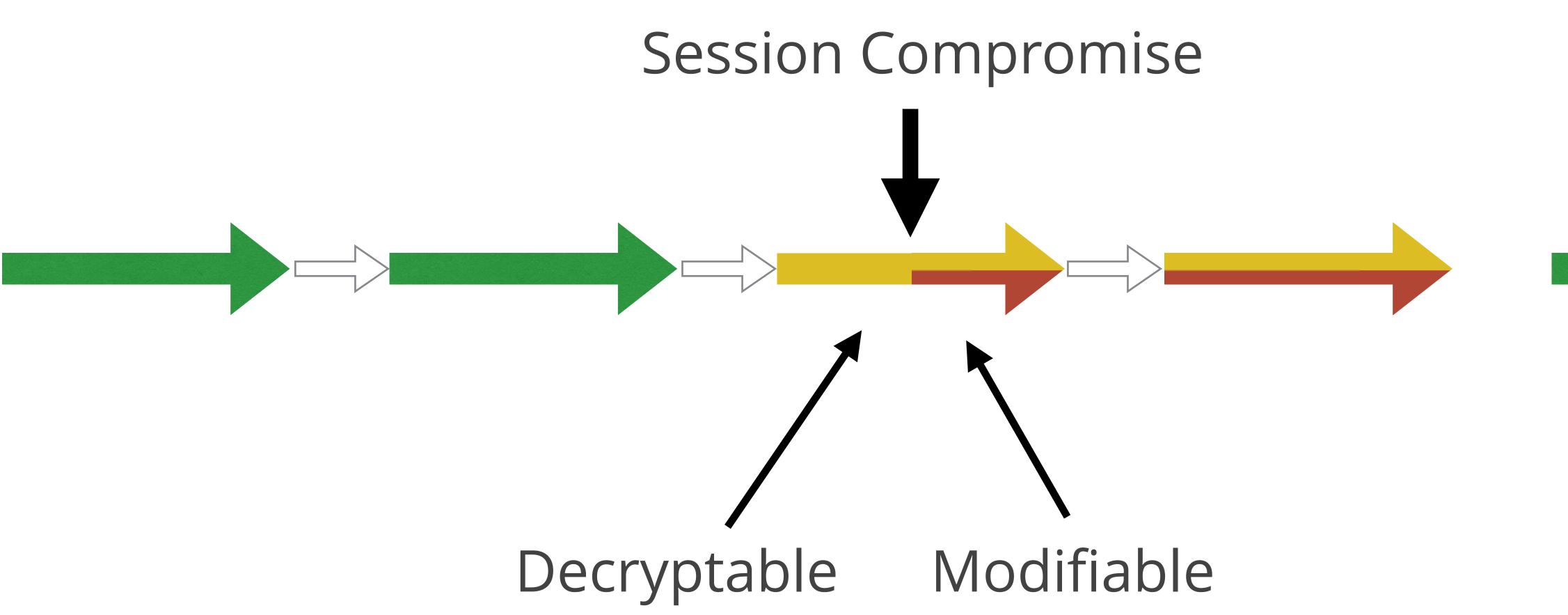


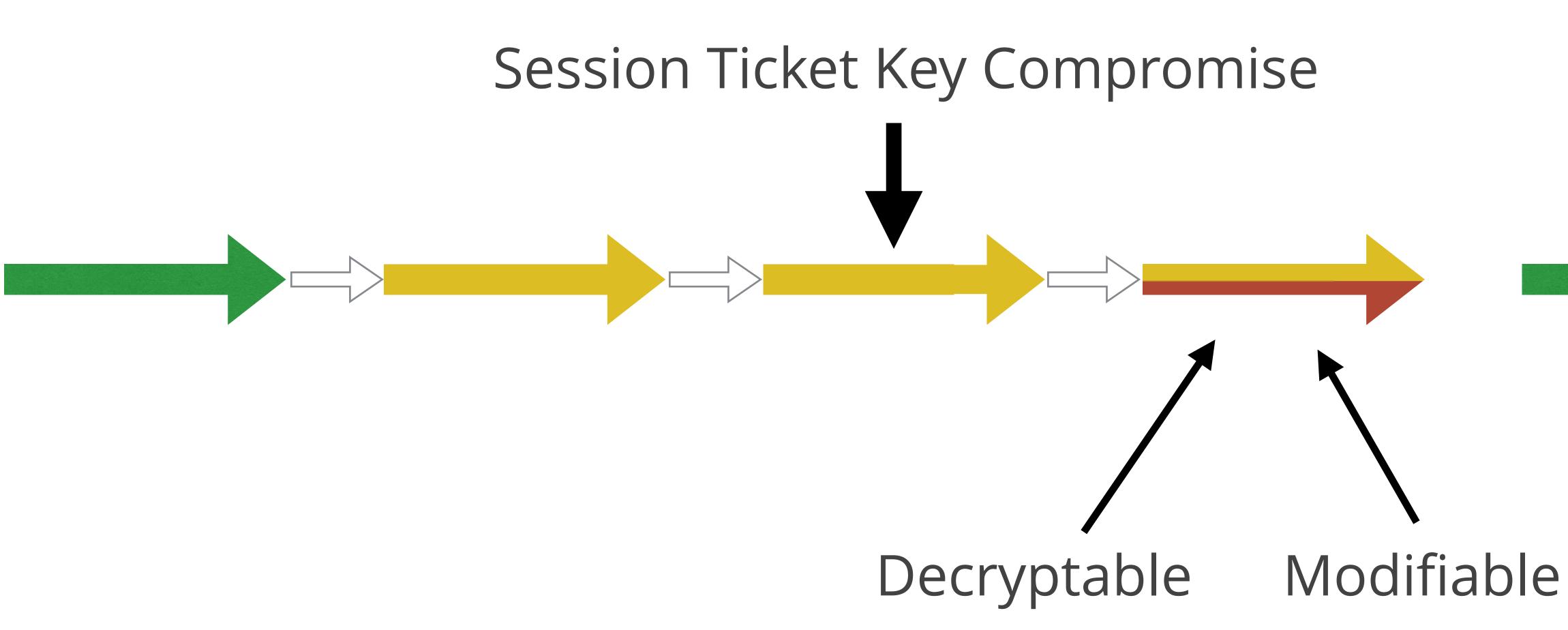












TLS 1.3 PSK-(EC)DHE

Client Hello Session Ticket (PSK) Key share

Finished



Latency: 1 round-trip

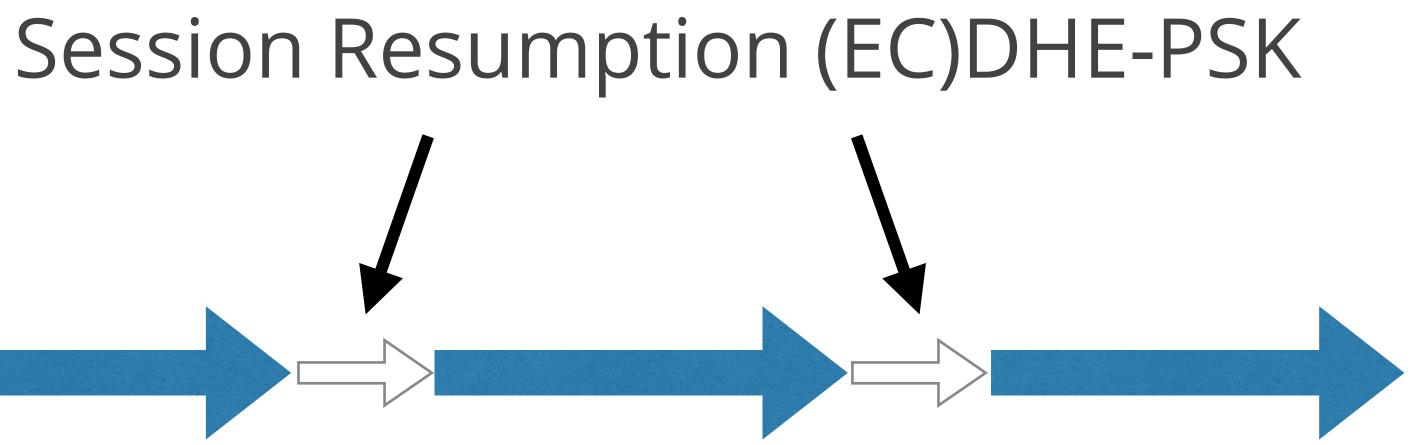


Key share

Finished

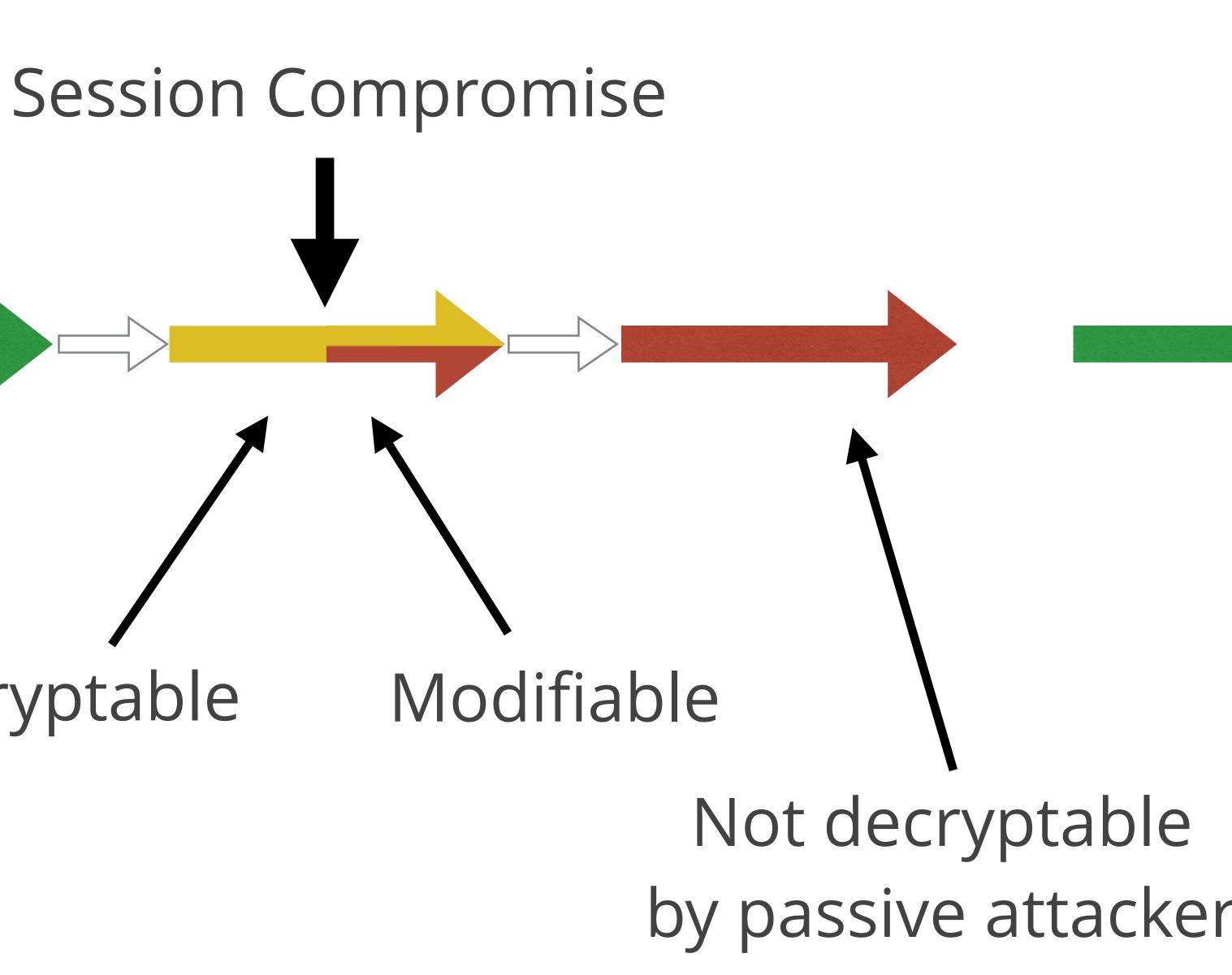


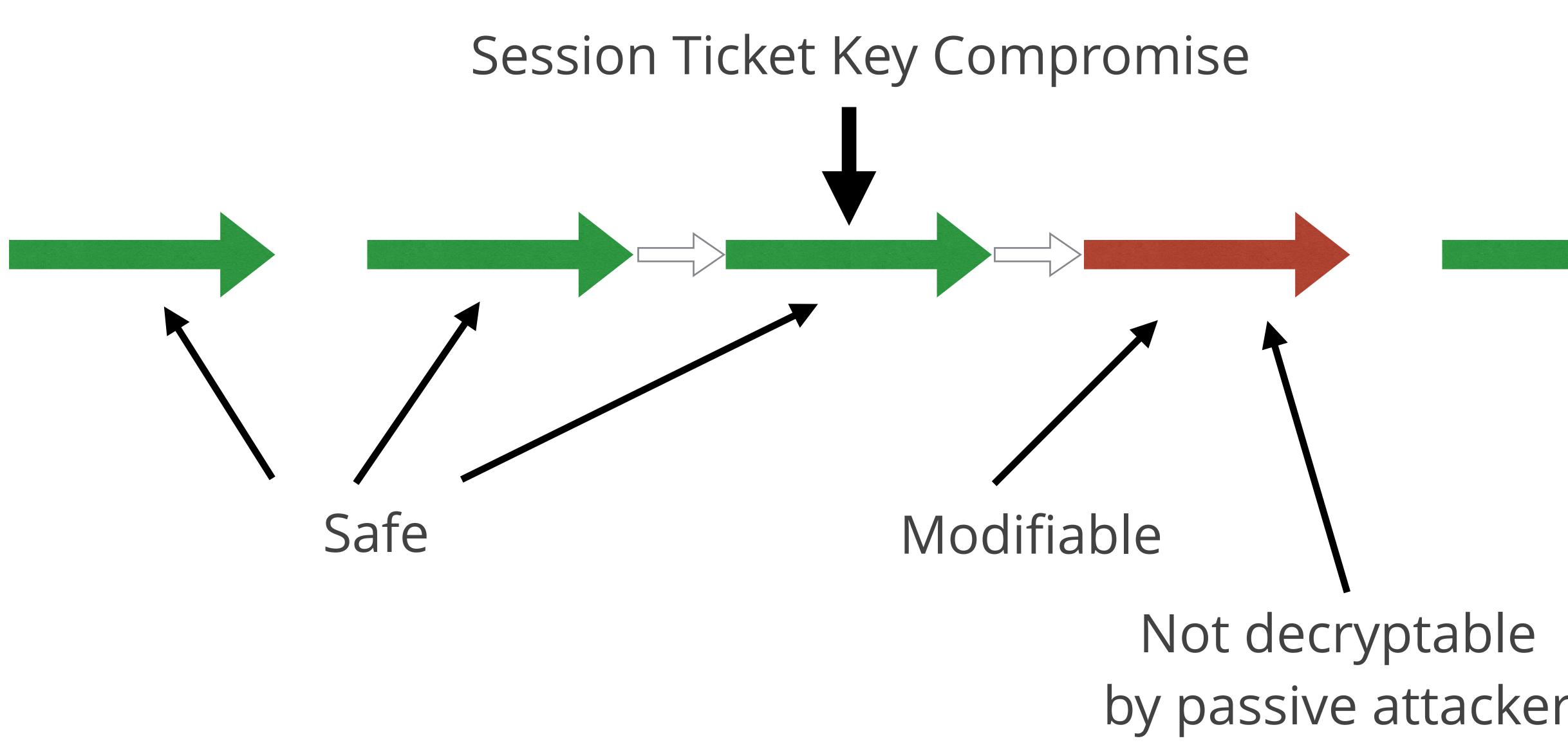






Decryptable





Latency: 1 round-trip TLS 1.3 PSK-(EC)DHE + Signature

Client Hello Session Ticket (PSK) Key share

Finished



Server Hello Key share Signature

Finished

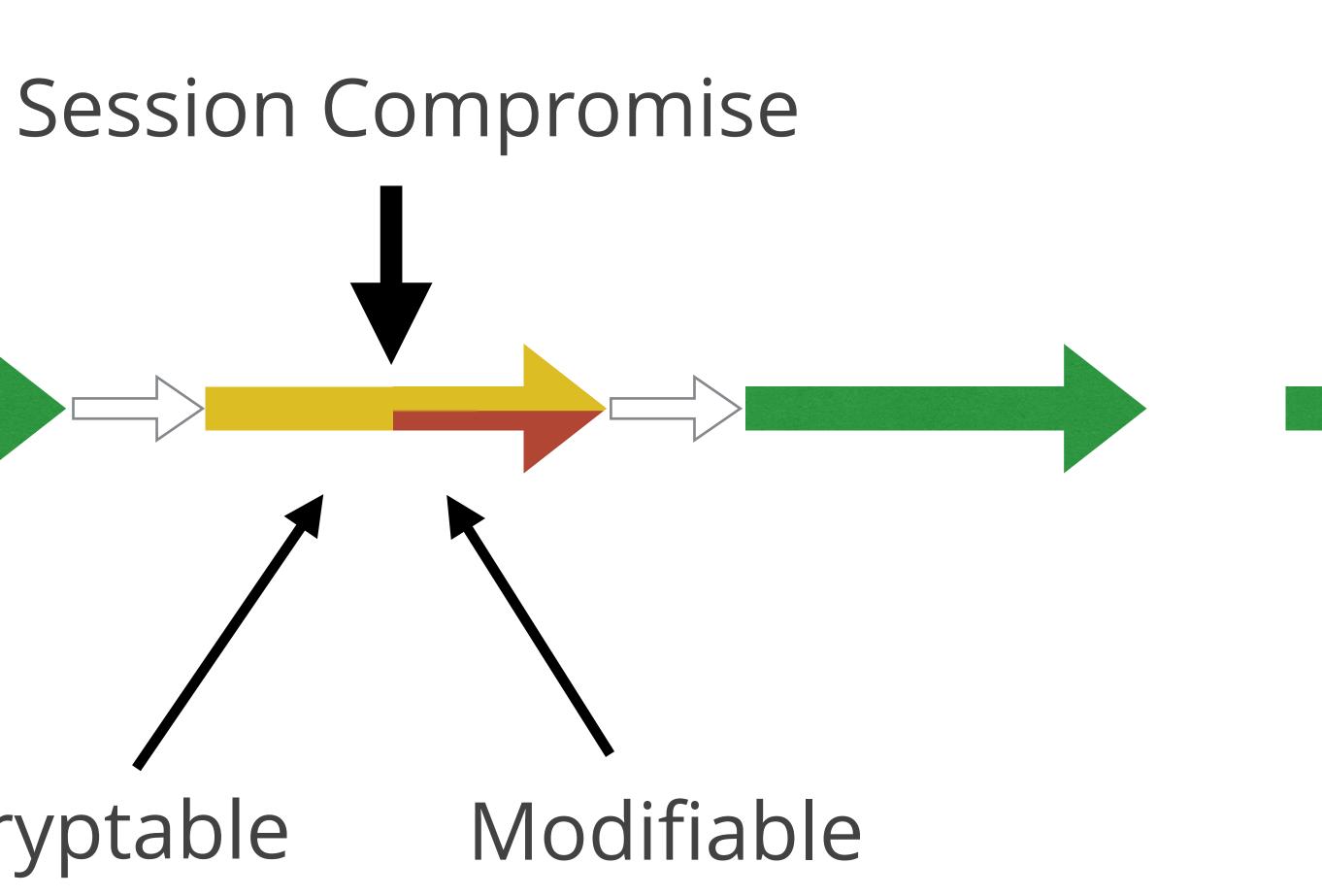


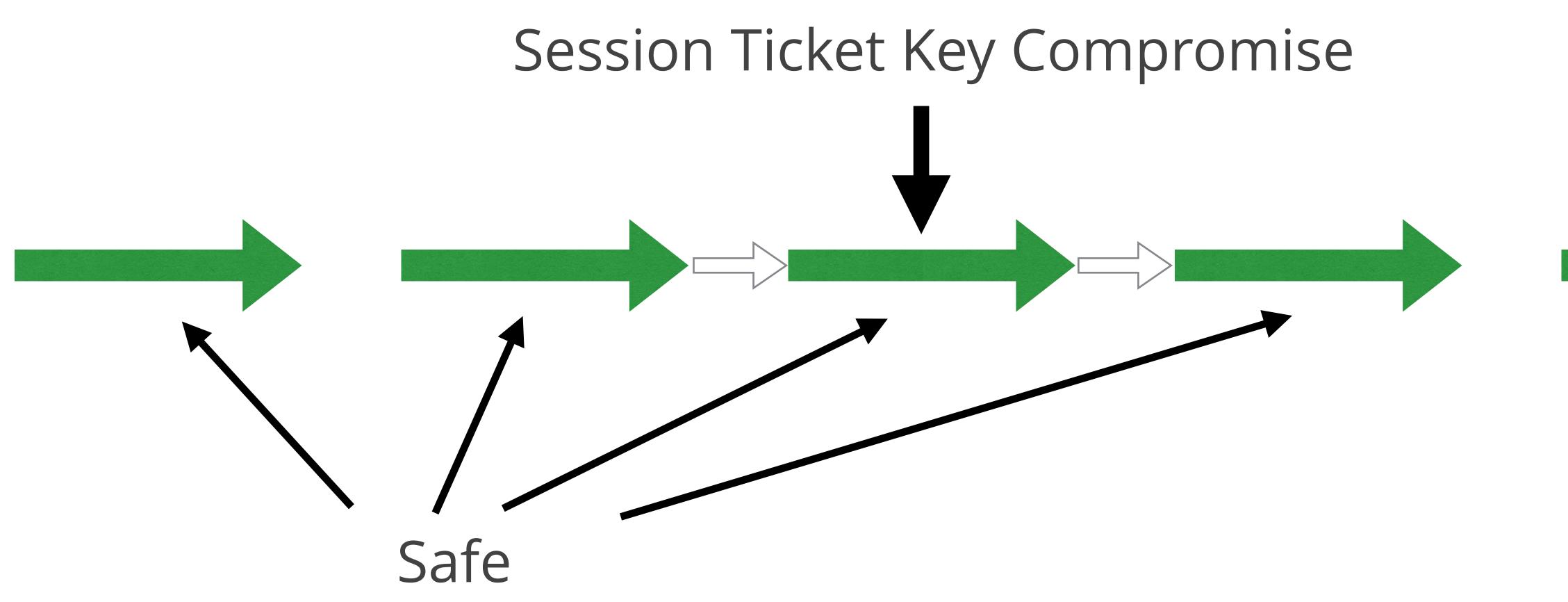
Session Resumption (EC)DHE-PSK + Signature

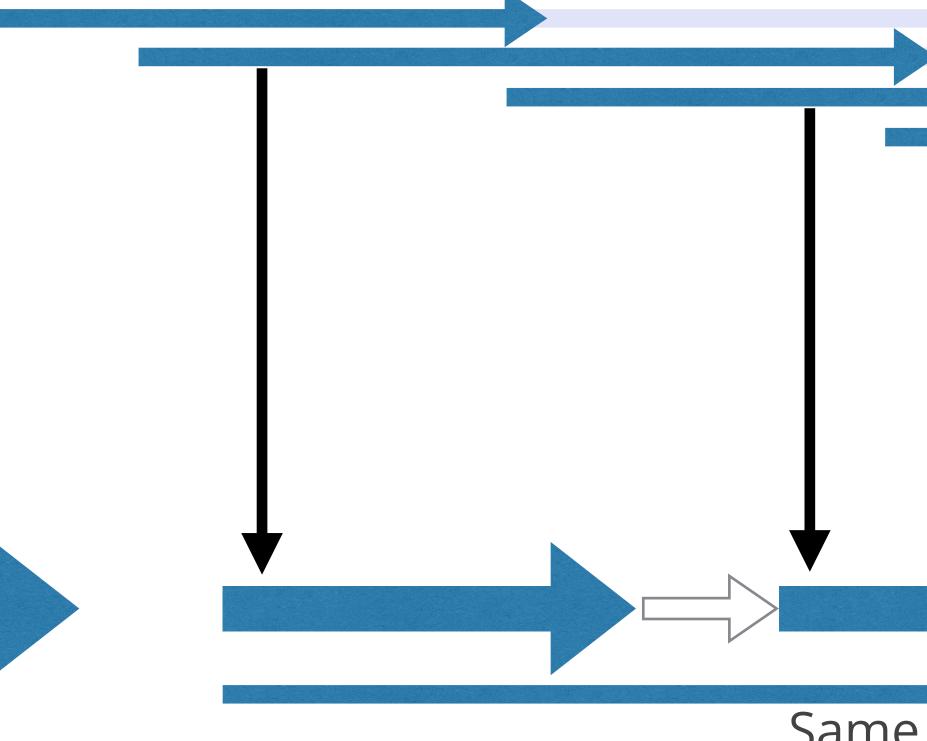




Decryptable



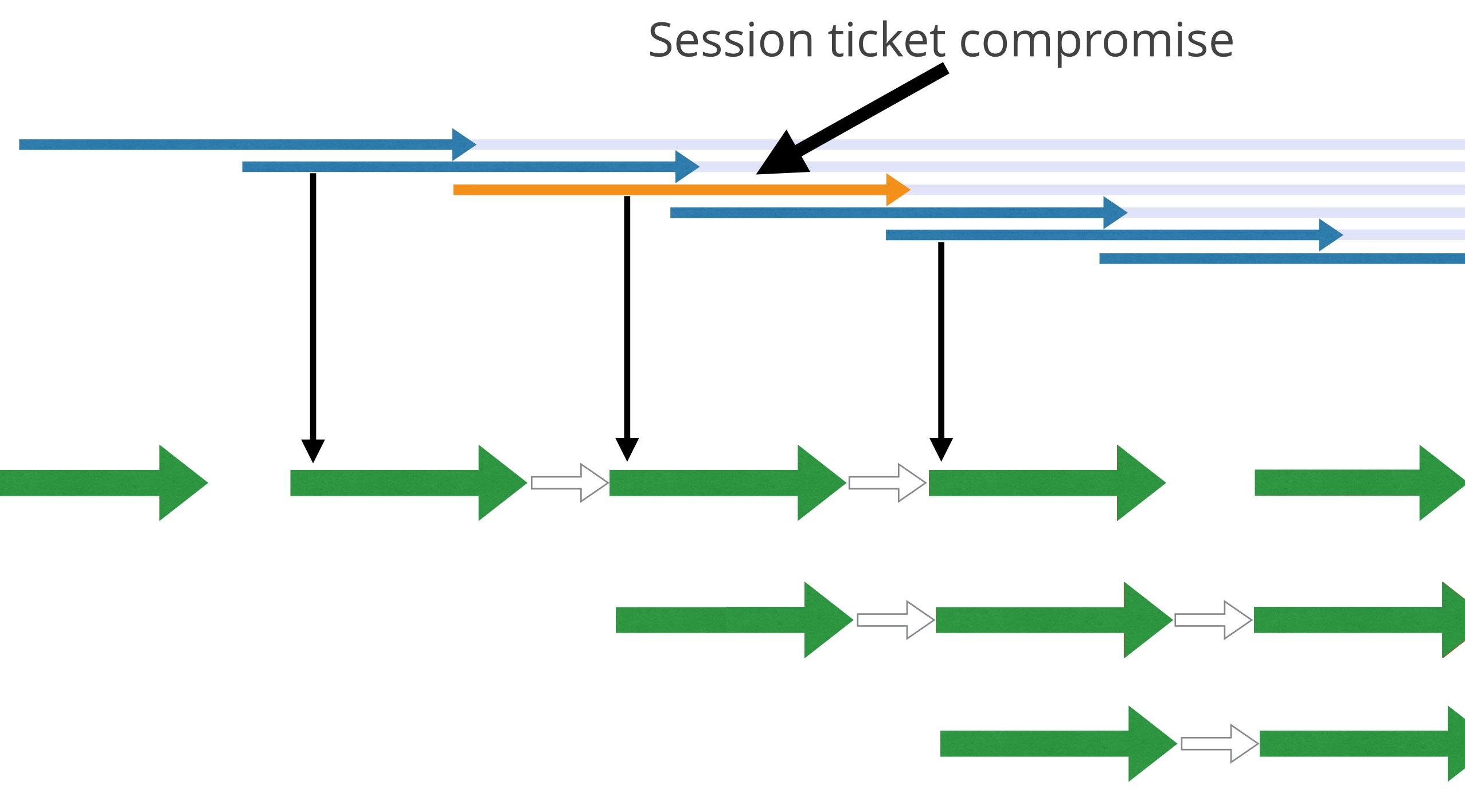






Session ticket lifetime Same session Same session





Enter O-RTT

Performance is king

60

TLS 1.3 0-RTT

Client Hello Session Ticket (PSK) [Key share]

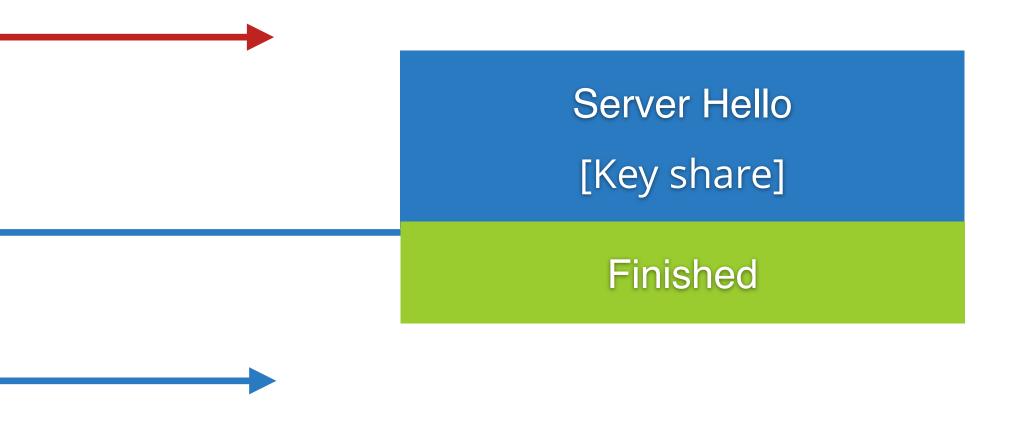
0-RTT Data

Finished

0-RTT data is encrypted with key derived from PSK/Resumption Secret



Latency: 0 round-trips





Session Resumption ORTT



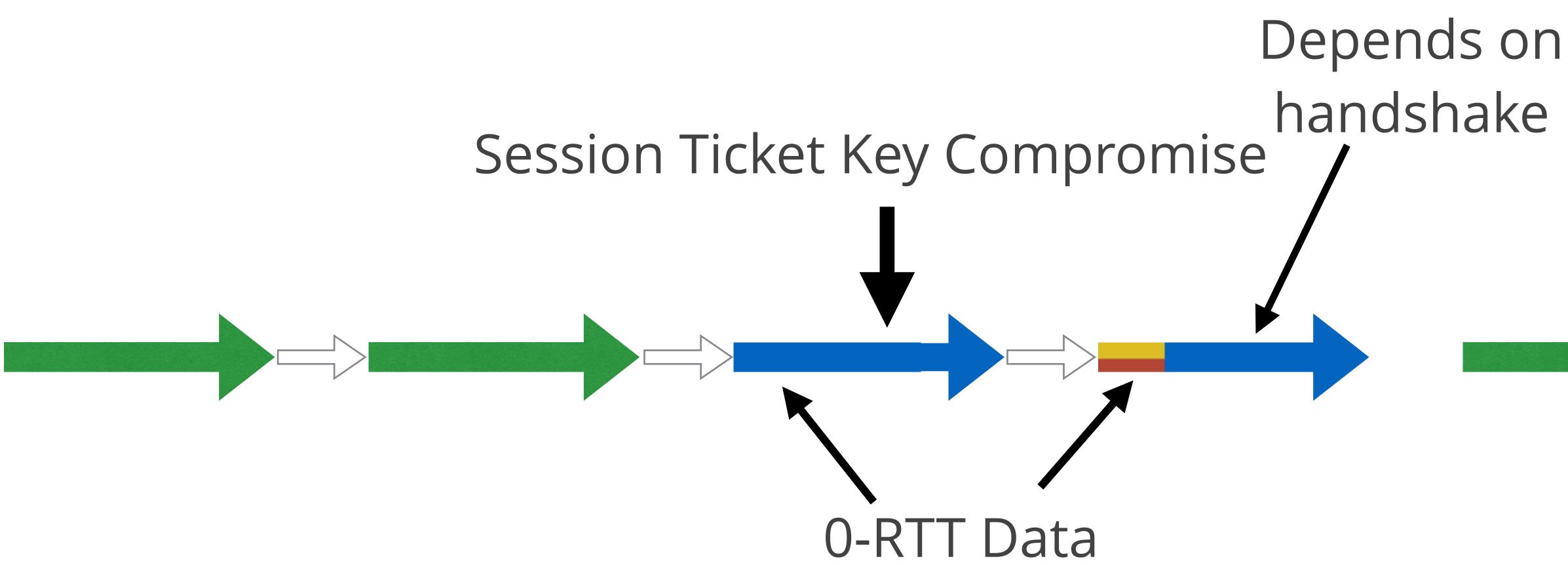




0-RTT Data has same properties as previous connection

Depends on handshake Session Compromise **0-RTT** Data





0-RTT Data is as safe as the application data from a PSK resumption

Key protection

Three keys

- ECDSA private key
- Session
- Session ticket encryption key



• Client compromise of session has same properties as server compromise

Physical/Logical attack

- ECDSA private key -> Can be stored in HSM, or soft HSM (Keyless SSL)
- Session -> In-memory only or server cache for load balancing
- Session ticket encryption key -> Centrally distributed

Only works as long as key is "alive"



Cryptographic advancement

- ECDSA private key -> Quantum computing
- Session -> Quantum computing for breaking DH, AES
- Session ticket encryption key -> Symmetric crypto

• Key can be stolen from cryptanalysis of transcript



In conclusion

69

Summary

- Forward secrecy in TLS is definitely not perfect
- Session resumption provides a complicated tradeoff
- Lifetimes of sessions and session ticket keys are important
 - Measurement needs to be done
- TLS 1.3 provides more protection at the cost of public key operations
- 0-RTT is not "forward secret" it's as secure as the resumption secret





Forward Secrecy in TLS

A Systematic Study

Nick Sullivan (@grittygrease)

Filippo Valsorda (@filosottile)

