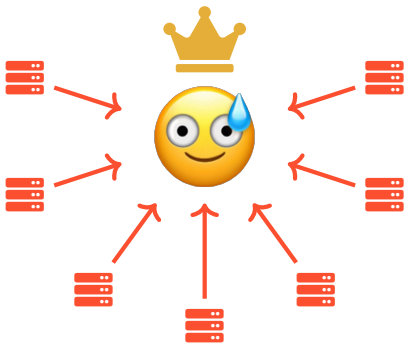


An Empirical Study of Consensus Protocols' DoS Resilience

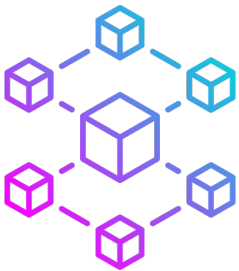
Giacomo Giuliari, Alberto Sonnino, Marc Frei, Fabio Streun, Lefteris Kokoris-Kogias, Adrian Perrig

Friday, July 5th, 2024
AsiaCCS '24

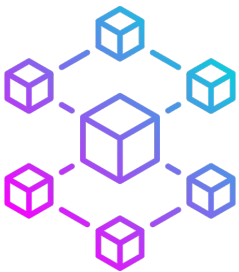
Mysten Labs

ETH zürich

Byzantine fault-tolerant (BFT) **consensus protocols** are becoming critical infrastructure



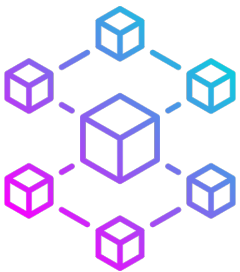
Byzantine fault-tolerant (BFT) **consensus protocols** are becoming critical infrastructure



Digital assets are increasingly exchanged and traded

One mechanism is through open blockchains

Byzantine fault-tolerant (BFT) **consensus protocols** are becoming critical infrastructure



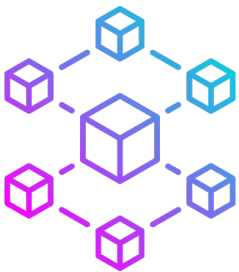
Digital assets are increasingly exchanged and traded

One mechanism is through open blockchains

BFT consensus is at the core of the latest blockchains

Lower latency and higher throughput, less carbon intensive

Byzantine fault-tolerant (BFT) **consensus protocols** are becoming critical infrastructure



Digital assets are increasingly exchanged and traded

One mechanism is through open blockchains

BFT consensus is at the core of the latest blockchains

Lower latency and higher throughput, less carbon intensive

BFT consensus provides compelling theoretical guarantees

for up to 1/3 compromised validators

The essential properties of consensus

The essential properties of consensus

Safety

No double spending, transactions are totally ordered

The essential properties of consensus

Safety

No double spending, transactions are totally ordered

Liveness

The protocol (eventually) makes progress

The essential properties of consensus

Safety

No double spending, transactions are totally ordered

Liveness

The protocol (eventually) makes progress

Even with 1/3 validators compromised

Safety and liveness are conditional on the network model

Safety and liveness are conditional on the network model

SYNC Messages are delivered within a known finite time Δ

Best case scenario, often not realistic

Safety and liveness are conditional on the network model

SYNC Messages are delivered within a known finite time Δ

Best case scenario, often not realistic

ASYNC Messages are delivered "eventually"

but they may arrive out of order in an unbounded time

Safety and liveness are conditional on the network model

SYNC	Messages are delivered within a known finite time Δ Best case scenario, often not realistic
ASYNC	Messages are delivered "eventually" but they may arrive out of order in an unbounded time
Partially SYNC	A mix of SYNC and ASYNC Synchronous after the global stabilization time (GST)

Safety and liveness are conditional on the network model

SYNC	Messages are delivered within a known finite time Δ Best case scenario, often not realistic
ASYNC	Messages are delivered "eventually" but they may arrive out of order in an unbounded time
Partially SYNC	A mix of SYNC and ASYNC Synchronous after the global stabilization time (GST)

Model violations are problematic

Safety and liveness are conditional on the network model

loss of safety and liveness

SYNC — Messages are delivered within a known finite time Δ
Best case scenario, often not realistic

ASYNC — Messages are delivered "eventually"
but they may arrive out of order in an unbounded time

Partially SYNC — A mix of SYNC and ASYNC
Synchronous after the global stabilization time (GST)

Model violations are problematic

Safety and liveness are conditional on the network model

loss of safety and liveness

SYNC — Messages are delivered within a known finite time Δ

Best case scenario, often not realistic

safety assured

ASYNC — Messages are delivered "eventually"

but they may arrive out of order in an unbounded time

Partially SYNC — A mix of SYNC and ASYNC

Synchronous after the global stabilization time (GST)

Model violations are problematic

Safety and liveness are conditional on the network model

loss of safety and liveness

SYNC — Messages are delivered within a known finite time Δ

Best case scenario, often not realistic

safety assured

ASYNC — Messages are delivered "eventually" — live in asynchrony

but they may arrive out of order in an unbounded time

Partially SYNC — A mix of SYNC and ASYNC

Synchronous after the global stabilization time (GST)

Model violations are problematic

Safety and liveness are conditional on the network model

loss of safety and liveness

SYNC — Messages are delivered within a known finite time Δ

Best case scenario, often not realistic

safety assured

ASYNC — Messages are delivered "eventually" — live in asynchrony

but they may arrive out of order in an unbounded time

Partially SYNC — A mix of SYNC and ASYNC — not live in aynchrony

Synchronous after the global stabilization time (GST)

Model violations are problematic

Can we enforce these theoretical assumptions in real deployments?

Can we enforce these theoretical assumptions in real deployments?

"Eventual delivery" of messages

Can we enforce these theoretical assumptions in real deployments?

"Eventual delivery" of messages

- Could take a long time

Can we enforce these theoretical assumptions in real deployments?

"Eventual delivery" of messages

- Could take a long time
- May require infinite buffers

Can we enforce these theoretical assumptions in real deployments?

"Eventual delivery" of messages

- Could take a long time
- May require infinite buffers

Can we enforce these theoretical assumptions in real deployments?

"Eventual delivery" of messages

- Could take a long time
- May require infinite buffers

Protocol *leader* for optimal communication complexity

Can we enforce these theoretical assumptions in real deployments?

"Eventual delivery" of messages

- Could take a long time
- May require infinite buffers

Protocol *leader* for optimal communication complexity

- Single point of failure

Can we enforce these theoretical assumptions in real deployments?

"Eventual delivery" of messages

- Could take a long time
- May require infinite buffers

Protocol *leader* for optimal communication complexity

- Single point of failure

Open distributed system + signatures for authentication

Can we enforce these theoretical assumptions in real deployments?

"Eventual delivery" of messages

- Could take a long time
- May require infinite buffers

Protocol *leader* for optimal communication complexity

- Single point of failure

Open distributed system + signatures for authentication

- Signature floods possible

We study attacks & defenses on state-of-the-art protocols

We study attacks & defenses on state-of-the-art protocols

HotStuff: partially-synchronous leader-based protocol

Optimal communication complexity (linear)

We study attacks & defenses on state-of-the-art protocols

HotStuff: partially-synchronous leader-based protocol

Optimal communication complexity (linear)

Tusk: aynchronous protocol

DAG based, no leader to target

We study attacks & defenses on state-of-the-art protocols

HotStuff: partially-synchronous leader-based protocol

Optimal communication complexity (linear)

Tusk: aynchronous protocol

DAG based, no leader to target

Global distributed testbed on AWS

4 regions

We study attacks & defenses on state-of-the-art protocols

HotStuff: partially-synchronous leader-based protocol

Optimal communication complexity (linear)

Tusk: aynchronous protocol

DAG based, no leader to target

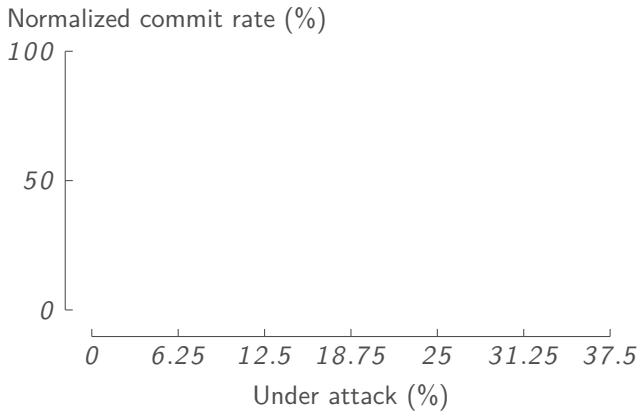
Global distributed testbed on AWS

4 regions

Up to 64 validators and 64 adversary bots

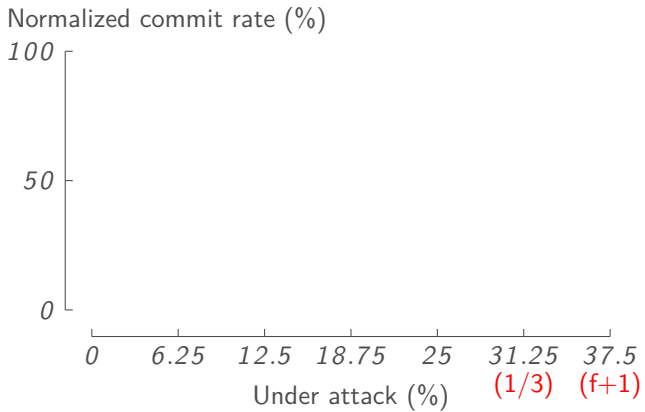
Multiple runs per scenario, hundreds of runs

Result 1: Attacks on a fixed subset of validators



$$\frac{\text{blocks per sec. attack}}{\text{blocks per sec. no attack}}$$

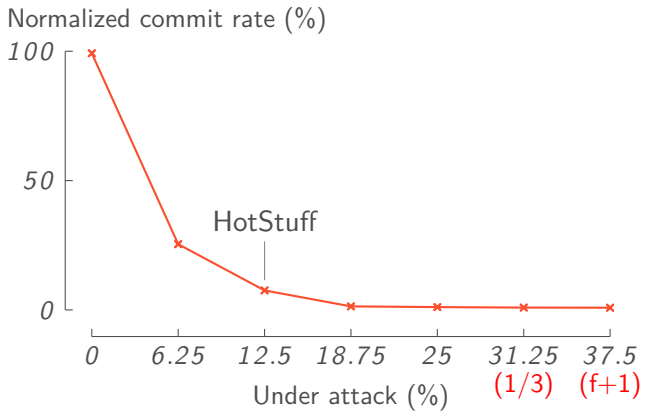
Result 1: Attacks on a fixed subset of validators



$$\frac{\text{blocks per sec. attack}}{\text{blocks per sec. no attack}}$$

Theory: live with less than $1/3$ (f+1) crashed

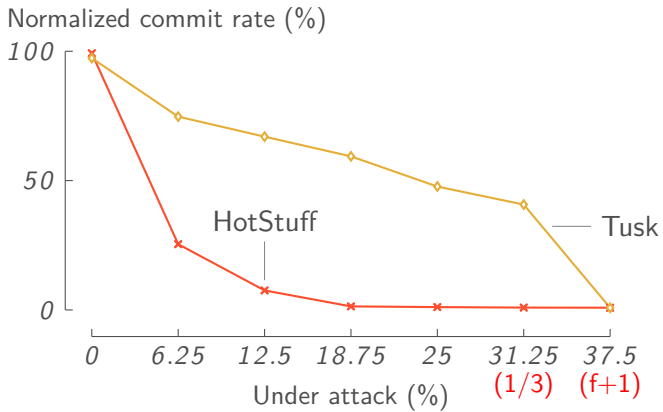
Result 1: Attacks on a fixed subset of validators



$$\frac{\text{blocks per sec. attack}}{\text{blocks per sec. no attack}}$$

Theory: live with less than 1/3 (f+1) crashed

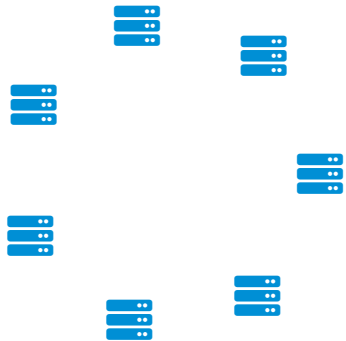
Result 1: Attacks on a fixed subset of validators



$$\frac{\text{blocks per sec. attack}}{\text{blocks per sec. no attack}}$$

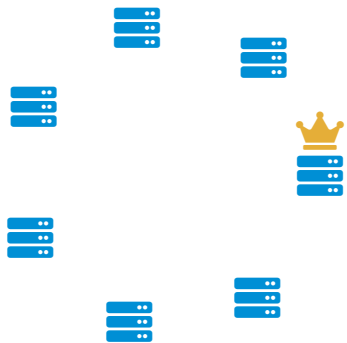
Theory: live with less than 1/3 (f+1) crashed

Result 2: Attacks against leader-based consensus protocols



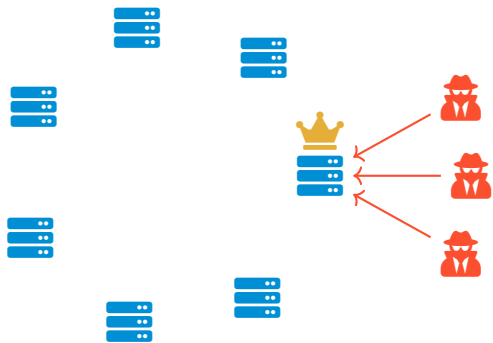
Result 2: Attacks against leader-based consensus protocols

For each round, the protocol has a leader



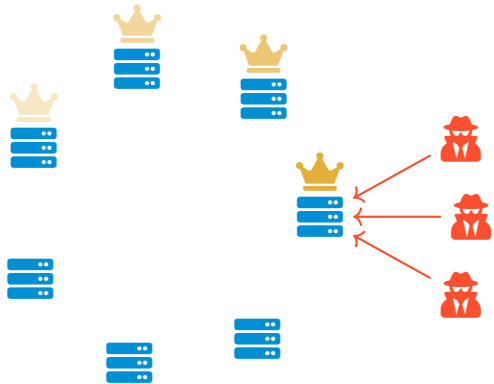
Result 2: Attacks against leader-based consensus protocols

For each round, the protocol has a leader



Result 2: Attacks against leader-based consensus protocols

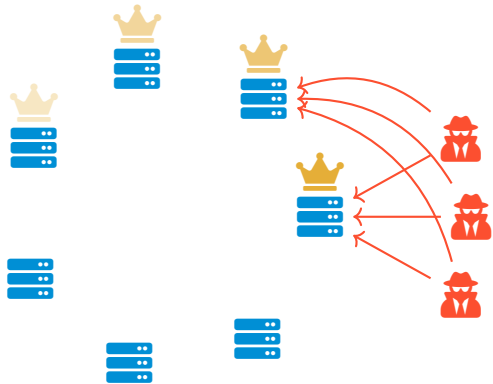
For each round, the protocol has a leader



Result 2: Attacks against leader-based consensus protocols

For each round, the protocol has a leader

The adversary knows the election sequence

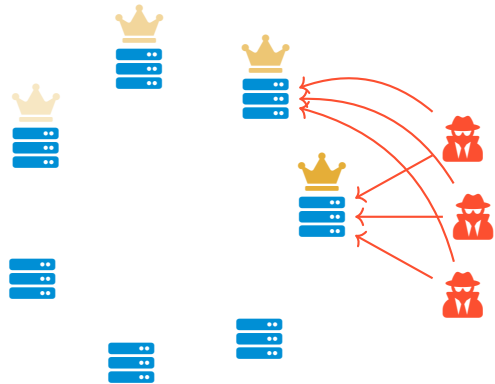


Result 2: Attacks against leader-based consensus protocols

For each round, the protocol has a leader

The adversary knows the election sequence

Consensus is **halted in seconds, with $<1/3$ crashes**



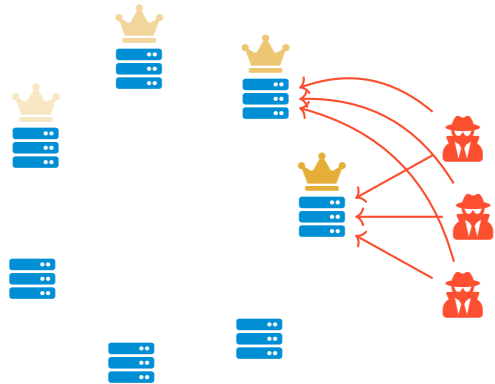
Result 2: Attacks against leader-based consensus protocols

For each round, the protocol has a leader

The adversary knows the election sequence

Consensus is **halted in seconds, with $<1/3$ crashes**

See the paper for unpredictable leader election



Defending consensus with symmetric-key authentication & rate limiting

Defending consensus with symmetric-key authentication & rate limiting

MACs: Line-rate auth.

Defending consensus with symmetric-key authentication & rate limiting

MACs: Line-rate auth.

DRKey: No attacks on connection establishment

Defending consensus with symmetric-key authentication & rate limiting

MACs: Line-rate auth.

DRKey: No attacks on connection establishment

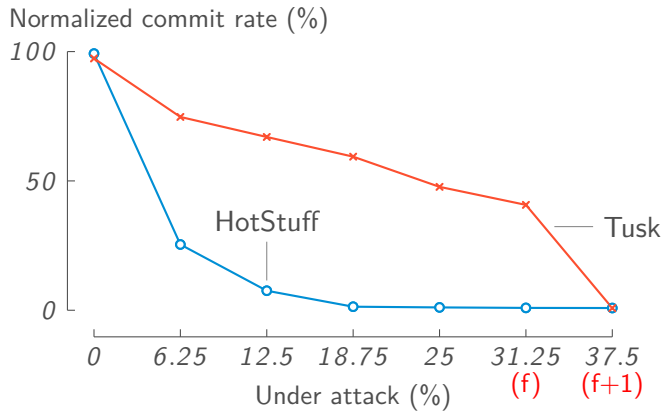
Rate limits: protect from internal adversaries

Defending consensus with symmetric-key authentication & rate limiting

MACs: Line-rate auth.

DRKey: No attacks on connection establishment

Rate limits: protect from internal adversaries

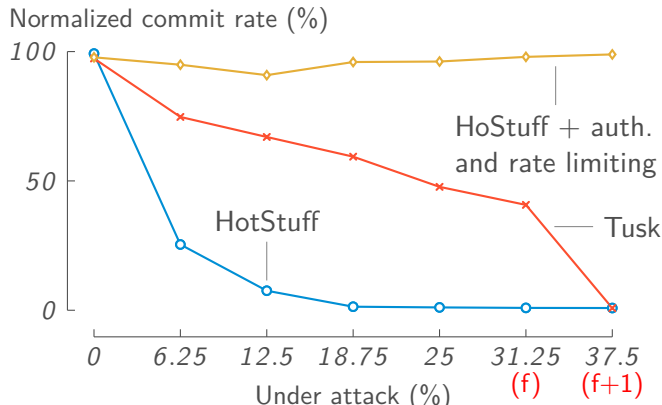


Defending consensus with symmetric-key authentication & rate limiting

MACs: Line-rate auth.

DRKey: No attacks on connection establishment

Rate limits: protect from internal adversaries



Future work

Future work

Analyze more consensus protocols

Protocols keep evolving

Future work

Analyze more consensus protocols

Protocols keep evolving

Expand attack vectors & mitigations

Consider clients and full nodes

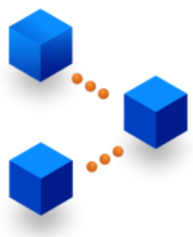
Conclusion: **Enforcing theoretical requirements of BFT consensus protocols is hard in real deployments**

Conclusion: **Enforcing theoretical requirements of BFT consensus protocols is hard in real deployments**

Decentralization constraints ▶ validator defense is hard

Many attack vectors, at multiple layers of the stack

Conclusion: **Enforcing theoretical requirements of BFT consensus protocols is hard in real deployments**



Decentralization constraints ▶ validator defense is hard

Many attack vectors, at multiple layers of the stack

Asymmetric cryptography is too slow

to be the only layer of defense

Conclusion: **Enforcing theoretical requirements of BFT consensus protocols is hard in real deployments**

Decentralization constraints ▶ validator defense is hard

Many attack vectors, at multiple layers of the stack

Asymmetric cryptography is too slow

to be the only layer of defense

Tusk performs better than HotStuff under attack

- ▶ Asynchronous operation
- ▶ No explicit leader

Conclusion: **Enforcing theoretical requirements of BFT consensus protocols is hard in real deployments**

Decentralization constraints ▶ validator defense is hard

Many attack vectors, at multiple layers of the stack

Asymmetric cryptography is too slow

to be the only layer of defense

Tusk performs better than HotStuff under attack

- ▶ Asynchronous operation
- ▶ No explicit leader

Thank you!

giacomo@mystenlabs.com

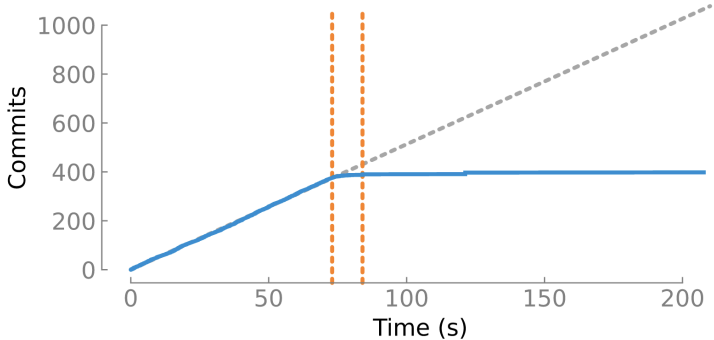
Appendix

Evaluation metrics: Time-to-last commit (TLC)

How much time before the adversary completely halts consensus?

Lower values ▶ more powerful attack

Ignore transient reconnections



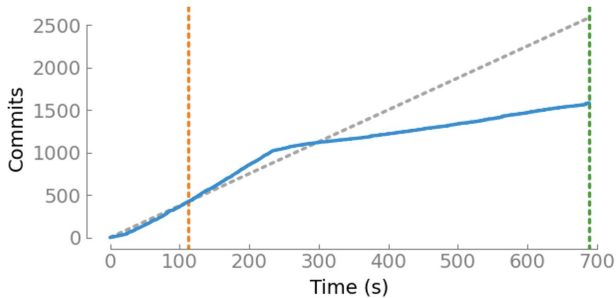
Evaluation metrics: Normalized commit rate (after attack start)

How much does the attack influence the consensus throughput?

$$\frac{\text{commit/s under attack}}{\text{commit/s normal operation}}$$

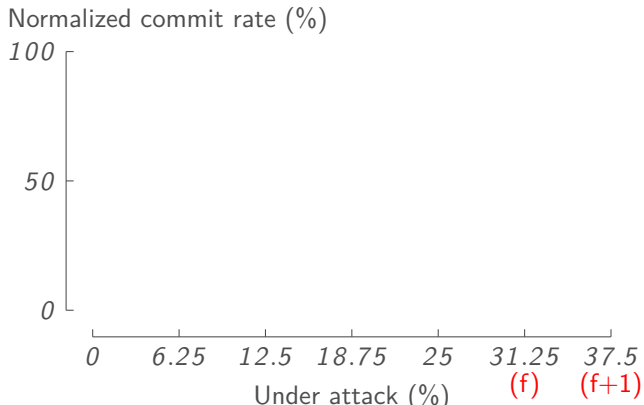
Lower values ▶ more powerful attack

Attack halts consensus ▶ Commit rate $\to 0$



Fixed subset attack results:
HotStuff's liveness threshold is lower than f+1

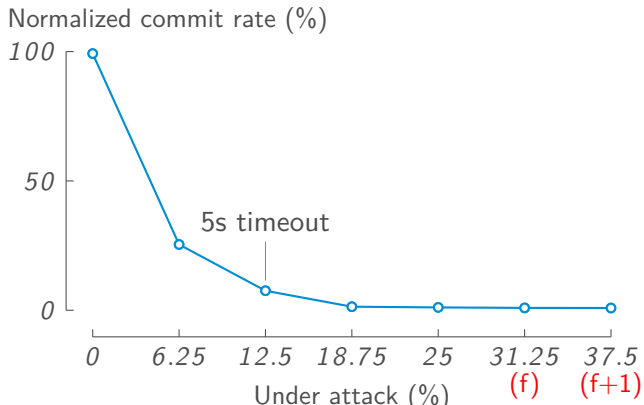
Attack a 16-validator HotStuff deployment



Fixed subset attack results:
HotStuff's liveness threshold is lower than f+1

Attack a 16-validator HotStuff deployment

Liveness almost completely lost with 2/3

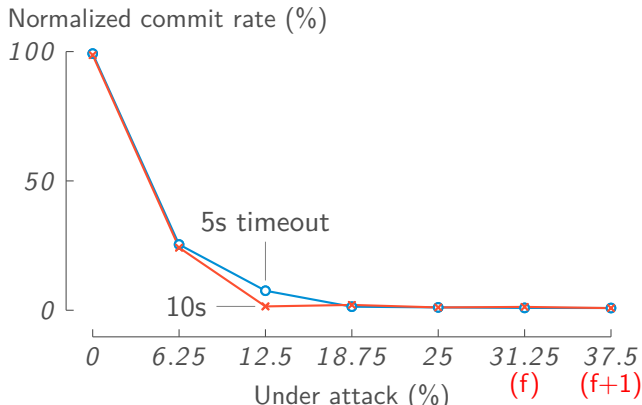


Fixed subset attack results:
HotStuff's liveness threshold is lower than f+1

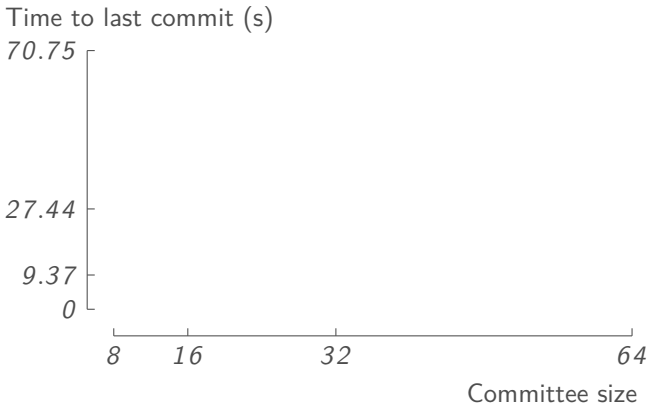
Attack a 16-validator HotStuff deployment

Liveness almost completely lost with 2/3

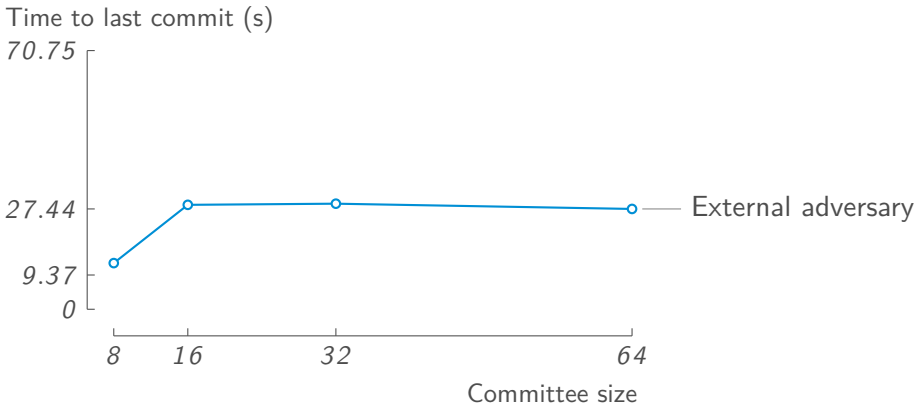
Longer timeouts may be more disruptive in practice



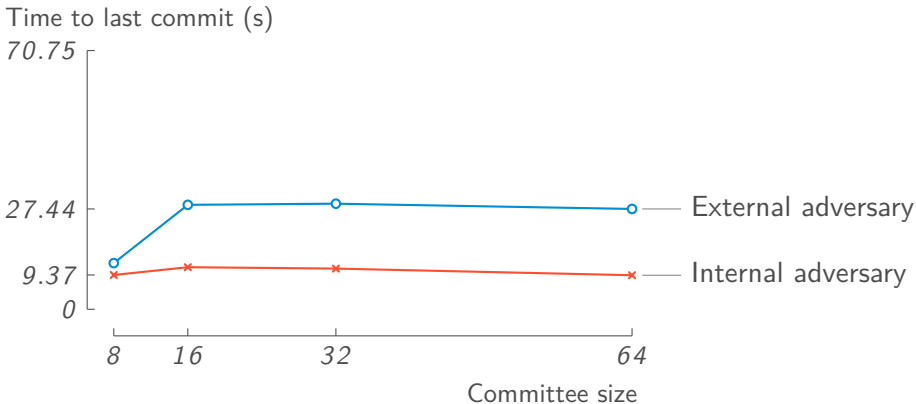
Leader-tracking attack results: HotStuff's progress is halted in seconds



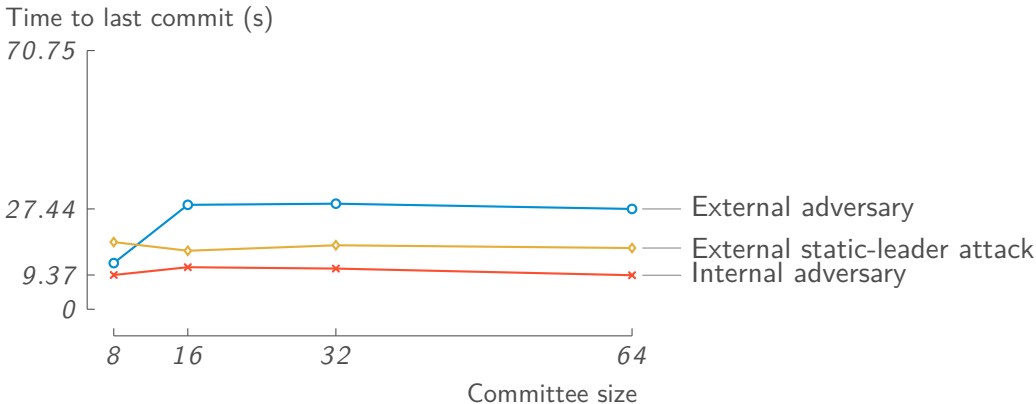
Leader-tracking attack results: HotStuff's progress is halted in seconds



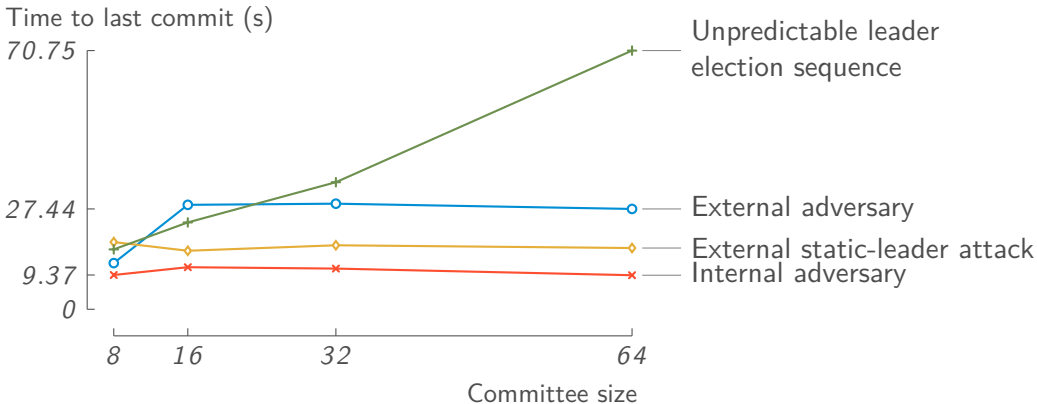
Leader-tracking attack results: HotStuff's progress is halted in seconds



Leader-tracking attack results: HotStuff's progress is halted in seconds

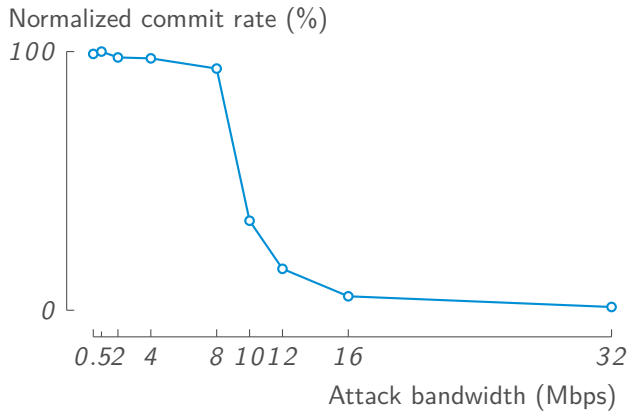


Leader-tracking attack results: HotStuff's progress is halted in seconds



Leader-tracking attack results:
Only a few Mbps of attack traffic per validator suffice

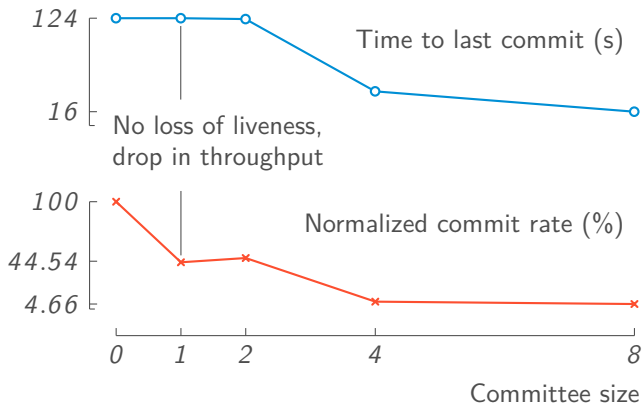
8–10 Mbps to each validator under target



Leader-tracking attack results: The adversary requires minimal resources

What is the minimum adversary size?

Attack a commitee of 64 validators



Why are these attacks possible?

The adversary exploits distributed consensus goals

Decentralized operation, trust only in signatures

Why are these attacks possible?

The adversary exploits distributed consensus goals

Decentralized operation, trust only in signatures

Prototype design only uses signatures for authentication

Why are these attacks possible?

The adversary exploits distributed consensus goals

Decentralized operation, trust only in signatures

Prototype design only uses signatures for authentication

Can we do better?

Traditional countermeasures are insufficient

Traditional countermeasures are insufficient

"Just use TLS"

Open to attacks on the handshake (asymmetric crypto) or on TCP state machine

Traditional countermeasures are insufficient

"Just use TLS"

Open to attacks on the handshake (asymmetric crypto) or on TCP state machine

"Just use Wireguard"

Open to attacks on the handshake/rekeying [1]

[1] F. Streun et. al, "Evaluating Susceptibility of VPN Implementations to DoS Attacks Using Adversarial Testing", NDSS '22.

Traditional countermeasures are insufficient

"Just use TLS"

Open to attacks on the handshake (asymmetric crypto) or on TCP state machine

"Just use Wireguard"

Open to attacks on the handshake/rekeying [1]

"Just use * with pre-shared keys"

WG uses asymmetric crypto with PSK. TLS PSK-only uses TCP. Limits use to known committee members.

[1] F. Streun et. al, "Evaluating Susceptibility of VPN Implementations to DoS Attacks Using Adversarial Testing", NDSS '22.

Traditional countermeasures are insufficient

"Just use TLS"

Open to attacks on the handshake (asymmetric crypto) or on TCP state machine

"Just use Wireguard"

Open to attacks on the handshake/rekeying [1]

"Just use * with pre-shared keys"

WG uses asymmetric crypto with PSK. TLS PSK-only uses TCP. Limits use to known committee members.

"Just add rate-limiting to *"

[1] F. Streun et. al, "Evaluating Susceptibility of VPN Implementations to DoS Attacks Using Adversarial Testing", NDSS '22.

Reduce the attack surface identifying the minimal set of properties

Property	TLS	WG	WG+PSK	TLS+PSK	Ideal
Confidentiality					
Authentication					
PFS					
No asymm crypto					
Rate limit					
PSK					
PSK with anyone					

Reduce the attack surface identifying the minimal set of properties

Property	TLS	WG	WG+PSK	TLS+PSK	Ideal
Confidentiality	✔				
Authentication					
PFS					
No asymm crypto					
Rate limit					
PSK					
PSK with anyone					

Reduce the attack surface identifying the minimal set of properties

Property	TLS	WG	WG+PSK	TLS+PSK	Ideal
Confidentiality	✔				
Authentication	✔				
PFS					
No asymm crypto					
Rate limit					
PSK					
PSK with anyone					

Reduce the attack surface identifying the minimal set of properties

Property	TLS	WG	WG+PSK	TLS+PSK	Ideal
Confidentiality	✔				
Authentication	✔				
PFS	✔				
No asymm crypto					
Rate limit					
PSK					
PSK with anyone					

Reduce the attack surface identifying the minimal set of properties

Property	TLS	WG	WG+PSK	TLS+PSK	Ideal
Confidentiality	✓				
Authentication	✓				
PFS	✓				
No asymm crypto	✗				
Rate limit					
PSK					
PSK with anyone					

Reduce the attack surface identifying the minimal set of properties

Property	TLS	WG	WG+PSK	TLS+PSK	Ideal
Confidentiality	✓				
Authentication	✓				
PFS	✓				
No asymm crypto	✗				
Rate limit	✗				
PSK					
PSK with anyone					

Reduce the attack surface identifying the minimal set of properties

Property	TLS	WG	WG+PSK	TLS+PSK	Ideal
Confidentiality	✓				
Authentication	✓				
PFS	✓				
No asymm crypto	✗				
Rate limit	✗				
PSK	✗				
PSK with anyone					

Reduce the attack surface identifying the minimal set of properties

Property	TLS	WG	WG+PSK	TLS+PSK	Ideal
Confidentiality	✓				
Authentication	✓				
PFS	✓				
No asymm crypto	✗				
Rate limit	✗				
PSK	✗				
PSK with anyone	✗				

Reduce the attack surface identifying the minimal set of properties

Property	TLS	WG	WG+PSK	TLS+PSK	Ideal
Confidentiality	✓	✓			
Authentication	✓	✓			
PFS	✓	✓			
No asymm crypto	✗	✗			
Rate limit	✗	✗			
PSK	✗	✗			
PSK with anyone	✗	✗			

Reduce the attack surface identifying the minimal set of properties

Property	TLS	WG	WG+PSK	TLS+PSK	Ideal
Confidentiality	✓	✓	✓		
Authentication	✓	✓	✓		
PFS	✓	✓	✓		
No asymm crypto	✗	✗	✗		
Rate limit	✗	✗	✗		
PSK	✗	✗	✓		
PSK with anyone	✗	✗	✗		

Reduce the attack surface identifying the minimal set of properties

Property	TLS	WG	WG+PSK	TLS+PSK	Ideal
Confidentiality	✓	✓	✓	✓	
Authentication	✓	✓	✓	✓	
PFS	✓	✓	✓	✗	
No asymm crypto	✗	✗	✗	✓	
Rate limit	✗	✗	✗	✗	
PSK	✗	✗	✓	✓	
PSK with anyone	✗	✗	✗	✗	

Reduce the attack surface identifying the minimal set of properties

Property	TLS	WG	WG+PSK	TLS+PSK	Ideal
Confidentiality	✓	✓	✓	✓	✗
Authentication	✓	✓	✓	✓	
PFS	✓	✓	✓	✗	
No asymm crypto	✗	✗	✗	✓	
Rate limit	✗	✗	✗	✗	
PSK	✗	✗	✓	✓	
PSK with anyone	✗	✗	✗	✗	

Reduce the attack surface identifying the minimal set of properties

Property	TLS	WG	WG+PSK	TLS+PSK	Ideal
Confidentiality	✓	✓	✓	✓	✗
Authentication	✓	✓	✓	✓	✓
PFS	✓	✓	✓	✗	
No asymm crypto	✗	✗	✗	✓	
Rate limit	✗	✗	✗	✗	
PSK	✗	✗	✓	✓	
PSK with anyone	✗	✗	✗	✗	

Reduce the attack surface identifying the minimal set of properties

Property	TLS	WG	WG+PSK	TLS+PSK	Ideal
Confidentiality	✓	✓	✓	✓	✗
Authentication	✓	✓	✓	✓	✓
PFS	✓	✓	✓	✗	✗
No asymm crypto	✗	✗	✗	✓	
Rate limit	✗	✗	✗	✗	
PSK	✗	✗	✓	✓	
PSK with anyone	✗	✗	✗	✗	

Reduce the attack surface identifying the minimal set of properties

Property	TLS	WG	WG+PSK	TLS+PSK	Ideal
Confidentiality	✓	✓	✓	✓	✗
Authentication	✓	✓	✓	✓	✓
PFS	✓	✓	✓	✗	✗
No asymm crypto	✗	✗	✗	✓	✓
Rate limit	✗	✗	✗	✗	
PSK	✗	✗	✓	✓	
PSK with anyone	✗	✗	✗	✗	

Reduce the attack surface identifying the minimal set of properties

Property	TLS	WG	WG+PSK	TLS+PSK	Ideal
Confidentiality	✓	✓	✓	✓	✗
Authentication	✓	✓	✓	✓	✓
PFS	✓	✓	✓	✗	✗
No asymm crypto	✗	✗	✗	✓	✓
Rate limit	✗	✗	✗	✗	✓
PSK	✗	✗	✓	✓	
PSK with anyone	✗	✗	✗	✗	

Reduce the attack surface identifying the minimal set of properties

Property	TLS	WG	WG+PSK	TLS+PSK	Ideal
Confidentiality	✓	✓	✓	✓	✗
Authentication	✓	✓	✓	✓	✓
PFS	✓	✓	✓	✗	✗
No asymm crypto	✗	✗	✗	✓	✓
Rate limit	✗	✗	✗	✗	✓
PSK	✗	✗	✓	✓	✓
PSK with anyone	✗	✗	✗	✗	

Reduce the attack surface identifying the minimal set of properties

Property	TLS	WG	WG+PSK	TLS+PSK	Ideal
Confidentiality	✓	✓	✓	✓	✗
Authentication	✓	✓	✓	✓	✓
PFS	✓	✓	✓	✗	✗
No asymm crypto	✗	✗	✗	✓	✓
Rate limit	✗	✗	✗	✗	✓
PSK	✗	✗	✓	✓	✓
PSK with anyone	✗	✗	✗	✗	✓

DRKey leverages the control-plane PKI for fast and efficient key distribution

Framework for authentication and key establishment
for secure network operations

Dynamically
Recreatable
Keys

DRKey leverages the control-plane PKI for fast and efficient key distribution

Dynamically Recreatable Keys

Framework for authentication and key establishment
for secure network operations

Integrated in the control-plane PKI
Sovereign operation within ISDs

DRKey leverages the control-plane PKI for fast and efficient key distribution

Dynamically Recreatable Keys

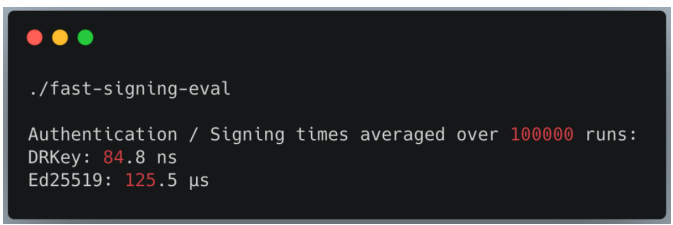
Framework for authentication and key establishment
for secure network operations

Integrated in the control-plane PKI
Sovereign operation within ISDs

A router can derive a key in ~ 20 ns
Enables source authentication and DDoS defence

Symmetric crypto key derivation at data-plane speed

Factor $\sim 1450x$



Idea: use a per-AS secret value to derive keys with a Pseudo-Random Function (PRF)

$K_{X \to Y} = PRF_{SV_X}(Y)$

Example: AS X creates a key for AS Y using secret value SV_X

Idea: use a per-AS secret value to derive keys with a Pseudo-Random Function (PRF)

$K_{X \to Y} = PRF_{SV_X}(Y)$

Example: AS X creates a key for AS Y
using secret value SV_X

Intel AES-NI instructions ▶ PRF within 30 cycles
$\sim 7x$ faster than a DRAM lookup

Idea: use a per-AS secret value to derive keys with a Pseudo-Random Function (PRF)

$K_{X \to Y} = PRF_{SV_X}(Y)$

Direction of derivation not communication

Example: AS X creates a key for AS Y
using secret value SV_X

Intel AES-NI instructions ▶ PRF within 30 cycles
$\sim 7x$ faster than a DRAM lookup

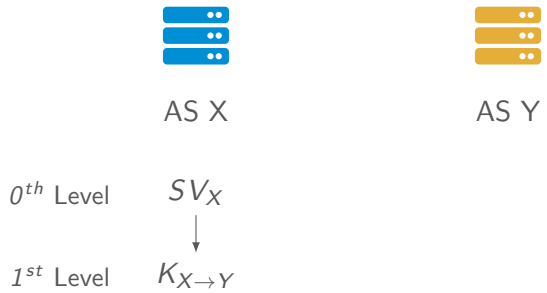
Key derviation is asymmetric
Know SV_X ▶ can derive $K_{X \to Y}$ fast

The key-server infrastructure is essential for DRkey's operation

AS X

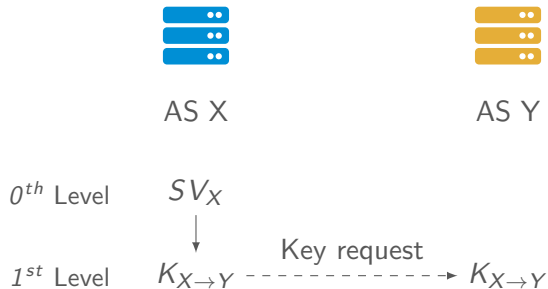
AS Y

The key-server infrastructure is essential for DRkey's operation



The key-server infrastructure is essential for DRkey's operation

Key requests are authenticated using the control-plane PKI

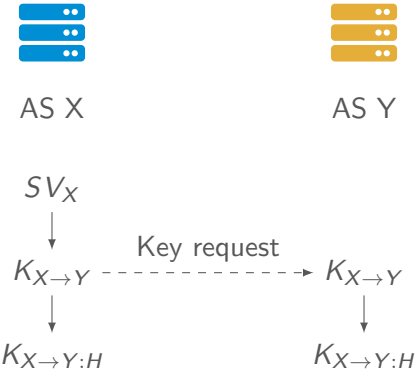


The key-server infrastructure is essential for DRkey's operation

Key requests are authenticated using the control-plane PKI

Only AS-level keys are shared across servers

$K_{X \to Y:H} = PRF_{K_{X \to Y}}(H)$



Lightning Filter: Scalable line-rate authentication

Use DRKey to authenticate data plane packets

Lightning Filter: Scalable line-rate authentication

Use DRKey to authenticate data plane packets

The server can re-derive the keys on the fly

The side under load is protected

Lightning Filter: Scalable line-rate authentication

Use DRKey to authenticate data plane packets

The server can re-derive the keys on the fly

The side under load is protected

Authenticate any host in the Internet

without asymmetric crypto

DRKey + Lightning Filter in one slide

SRC AS 

 DST AS

DRKey + Lightning Filter in one slide

DRKey + Lightning Filter in one slide

DRKey + Lightning Filter in one slide

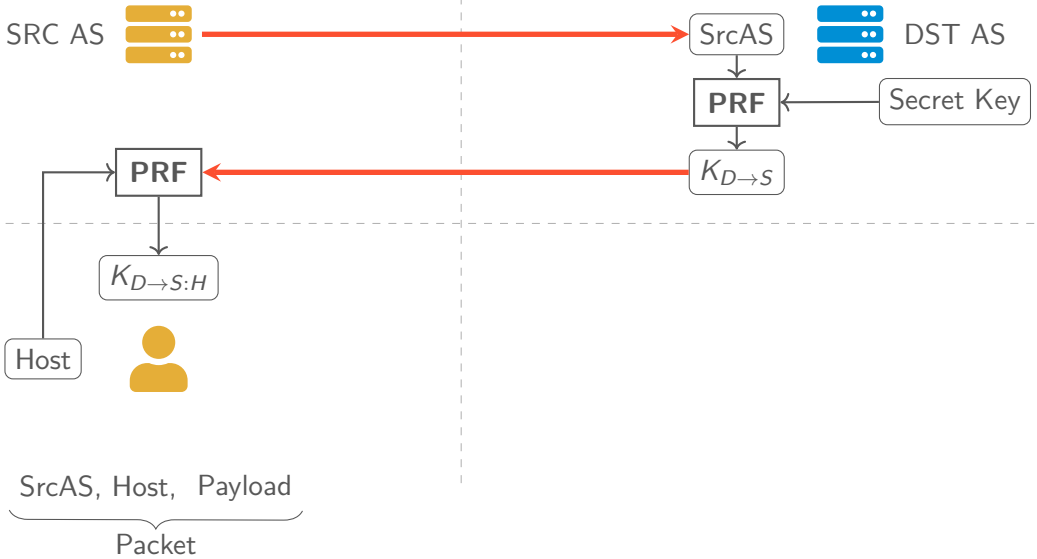
DRKey + Lightning Filter in one slide

Host

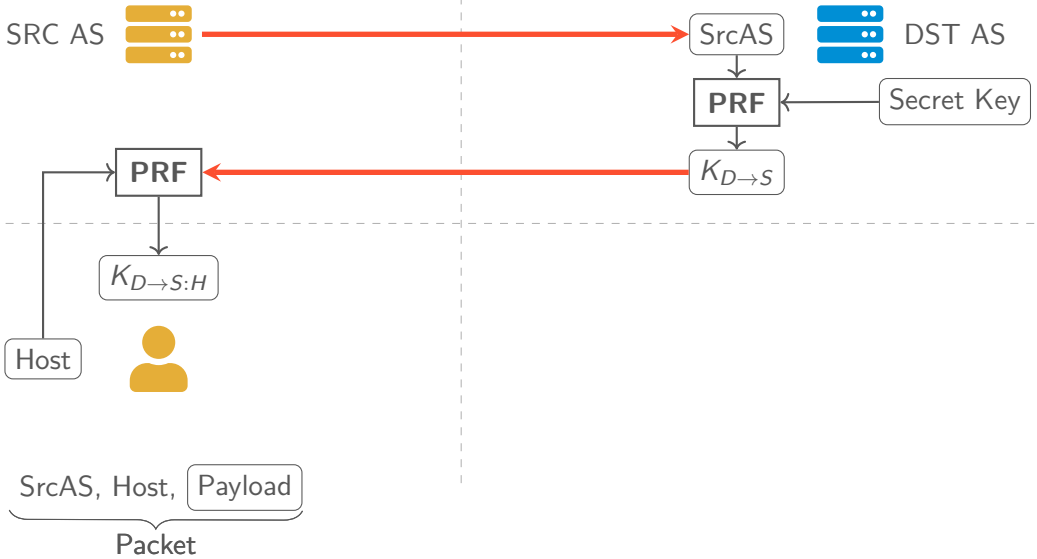
DRKey + Lightning Filter in one slide

DRKey + Lightning Filter in one slide

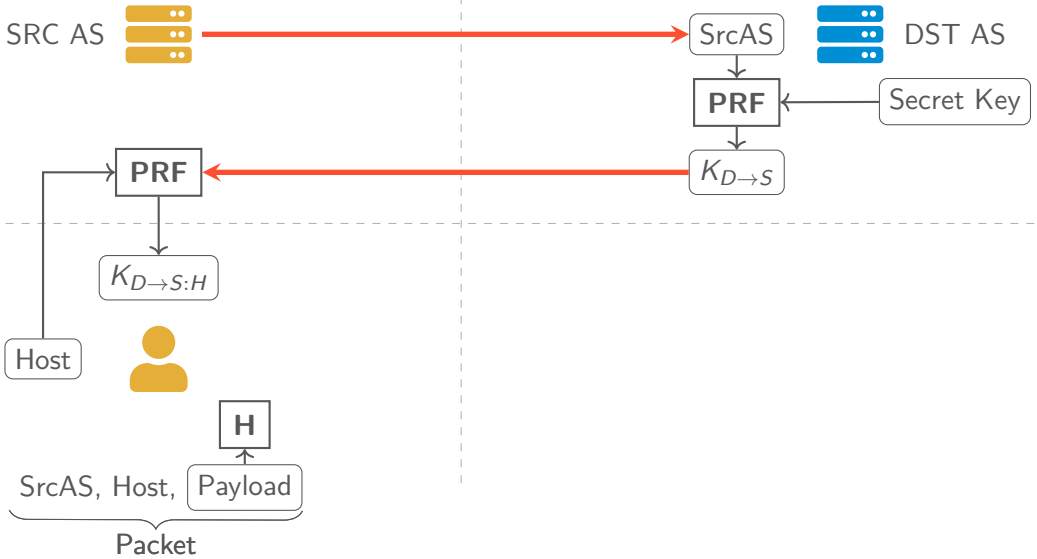
DRKey + Lightning Filter in one slide



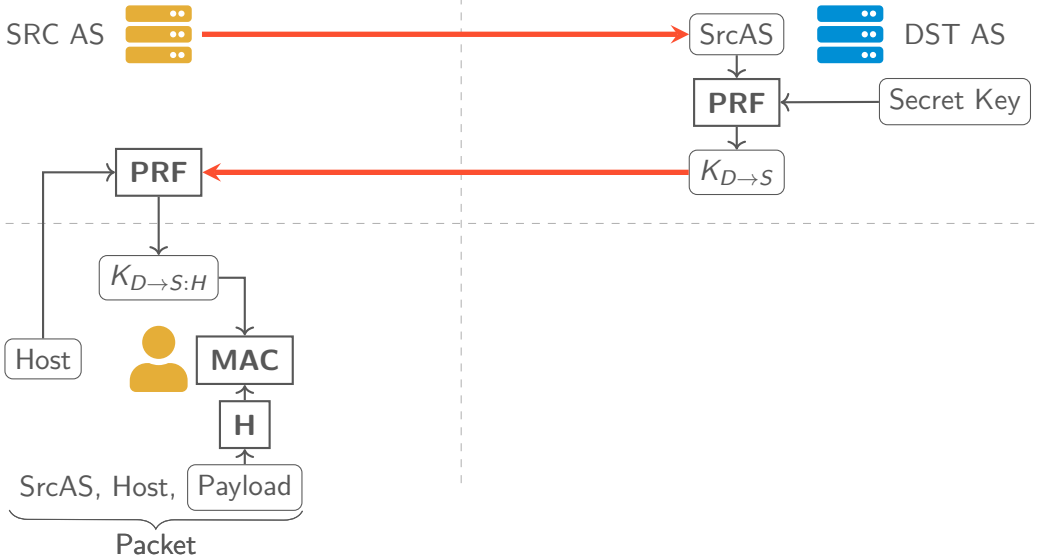
DRKey + Lightning Filter in one slide



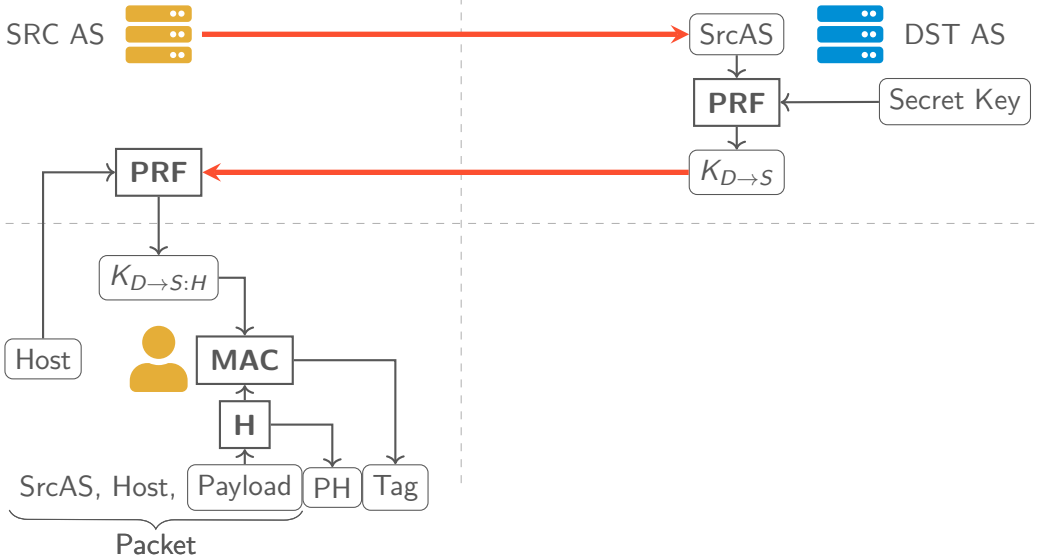
DRKey + Lightning Filter in one slide



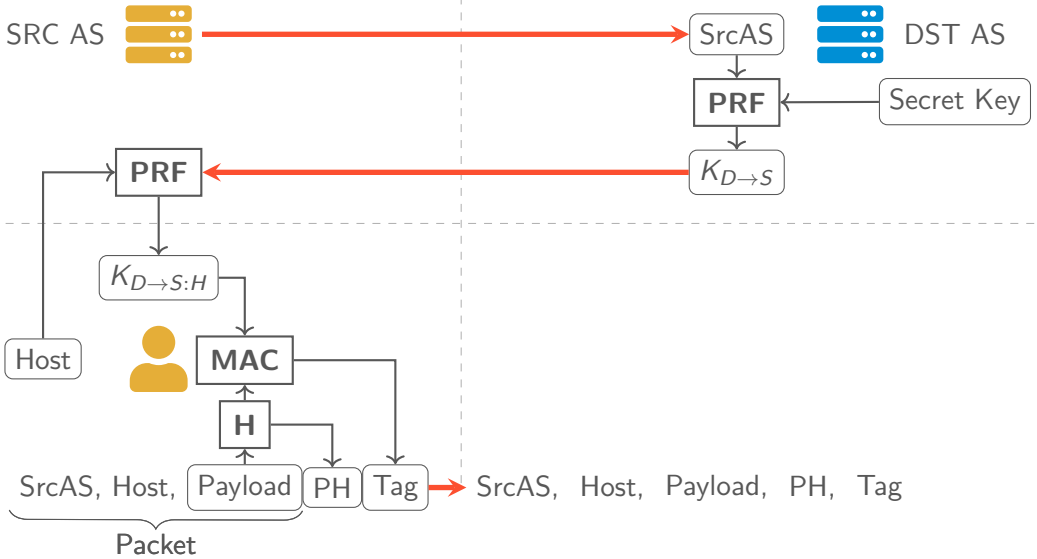
DRKey + Lightning Filter in one slide



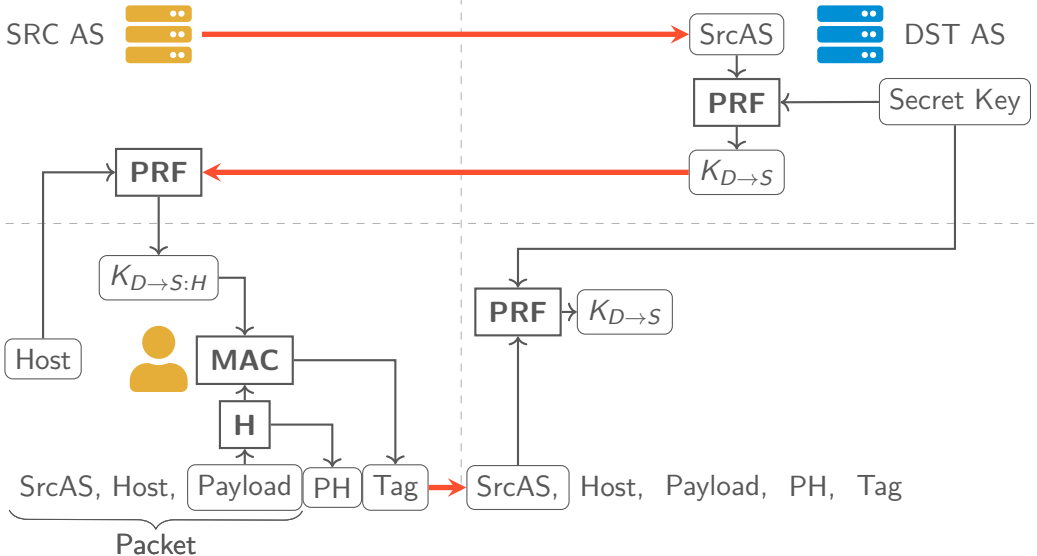
DRKey + Lightning Filter in one slide



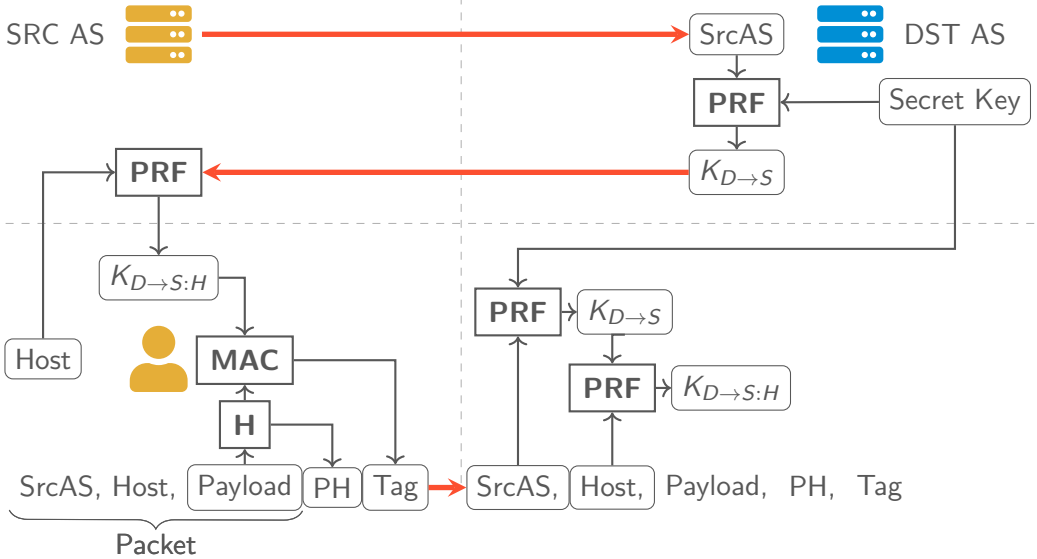
DRKey + Lightning Filter in one slide



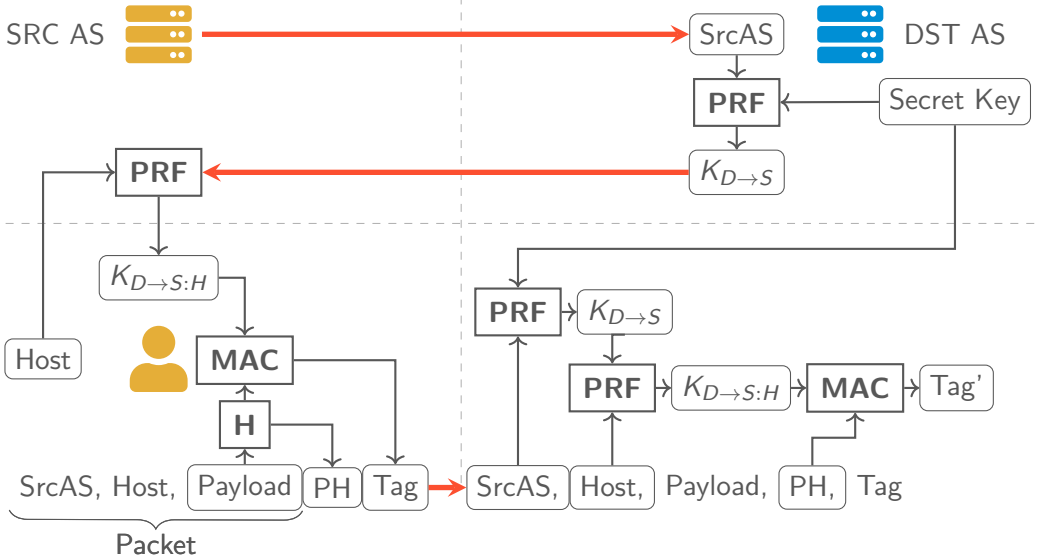
DRKey + Lightning Filter in one slide



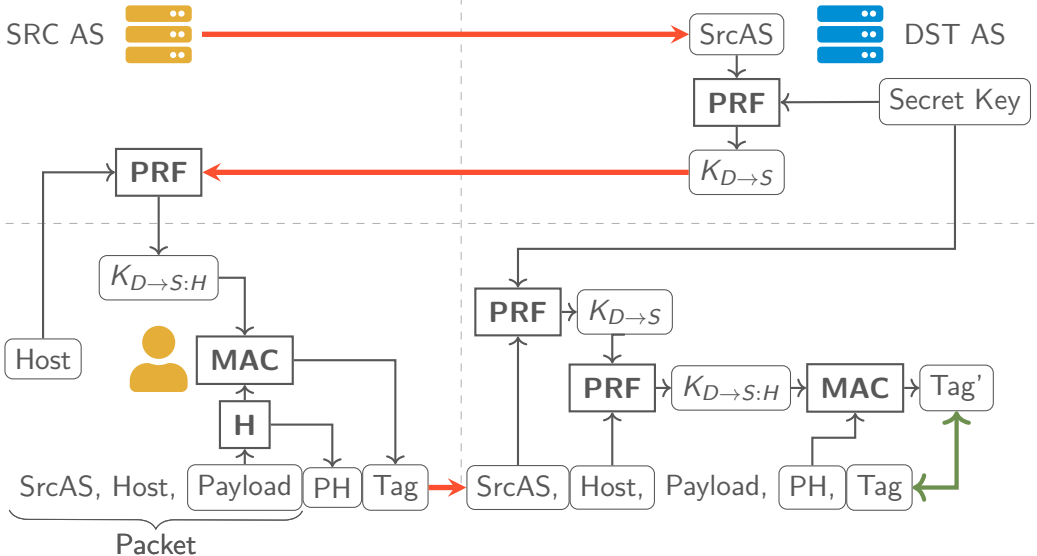
DRKey + Lightning Filter in one slide



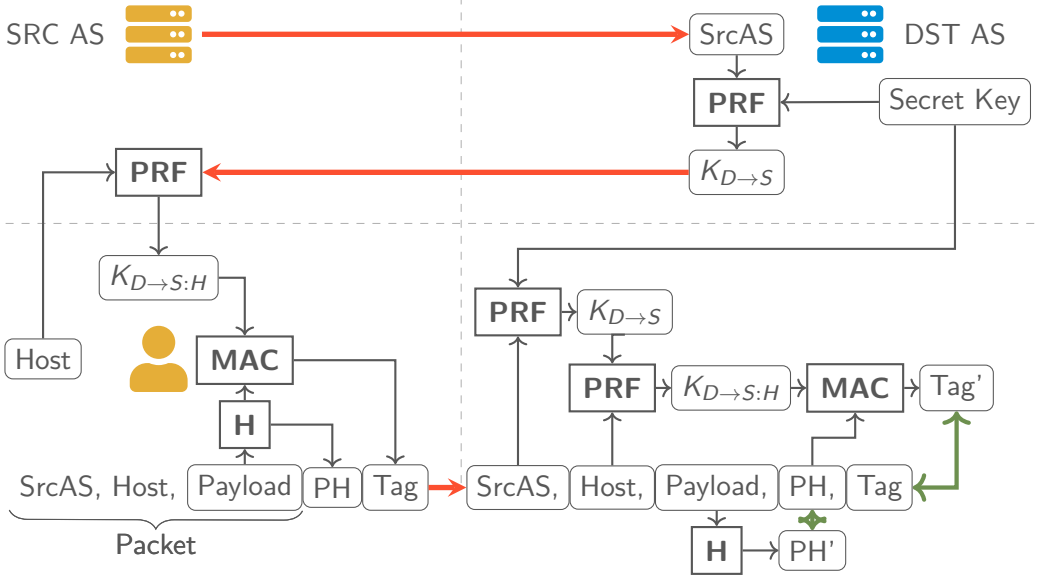
DRKey + Lightning Filter in one slide



DRKey + Lightning Filter in one slide



DRKey + Lightning Filter in one slide



Lightning Filter Recap

A server can authenticate any client in the Internet
By re-deriving DRKeys

Minimal feature set offers small attack surface
Optimized to authenticate traffic at line rate

Rate-limiting is built in

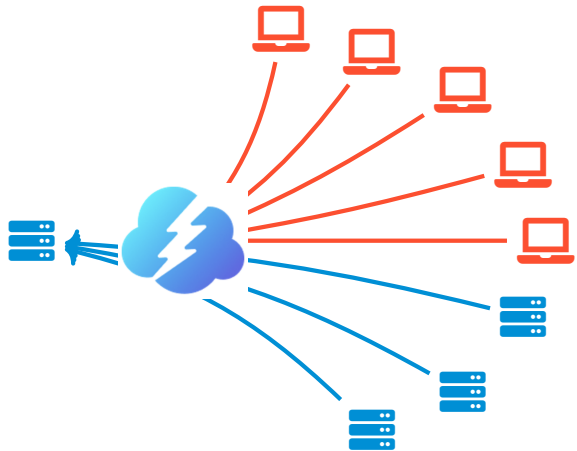
Lightning Filter protects validators from external and internal adversaries

Lightning Filter protects validators from external and internal adversaries

Lightning Filter protects validators from external and internal adversaries

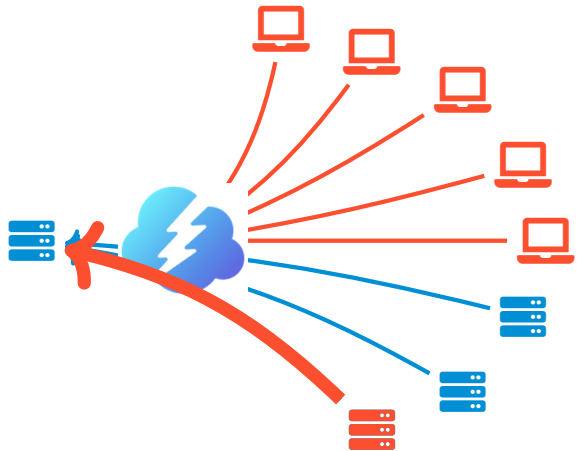
Lightning Filter protects validators from external and internal adversaries

Source authentication prevents attacks from outside the committee



Lightning Filter protects validators from external and internal adversaries

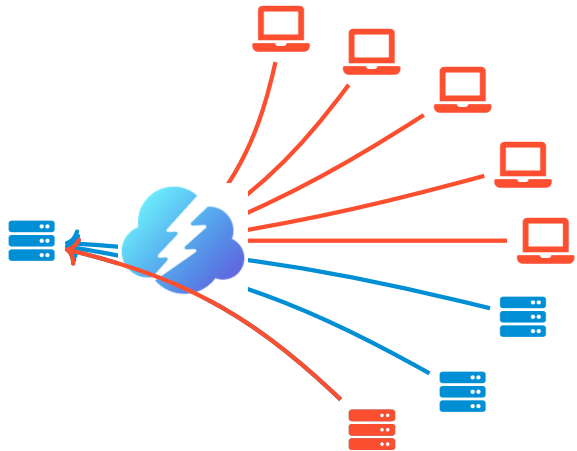
Source authentication prevents attacks from outside the committee



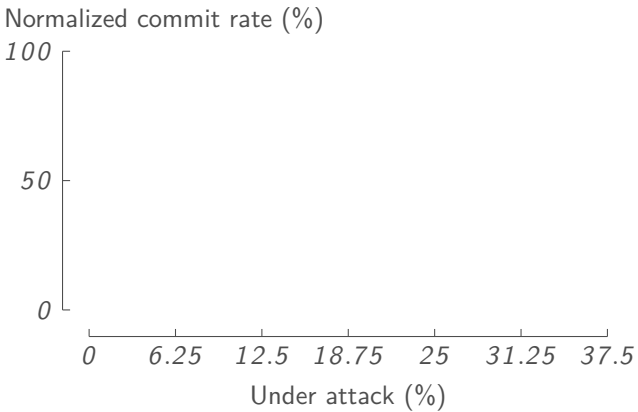
Lightning Filter protects validators from external and internal adversaries

Source authentication prevents attacks from outside the committee

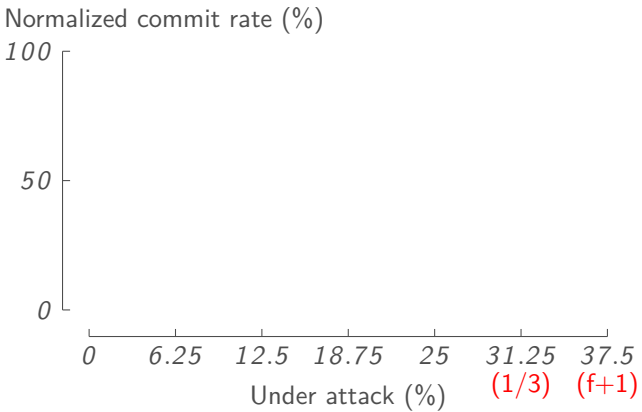
Rate limiting blocks Byzantine validators



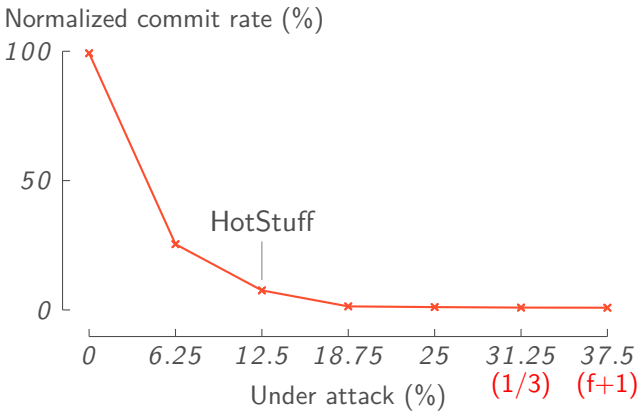
Evaluation: LF protection for consensus protocols



Evaluation: LF protection for consensus protocols



Evaluation: LF protection for consensus protocols



Evaluation: LF protection for consensus protocols