Karlsruhe Institute of Technology

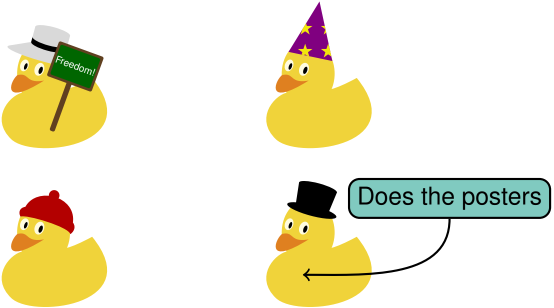
KASTEL

NEC

Panini – Anonymous Anycast and an Instantiation

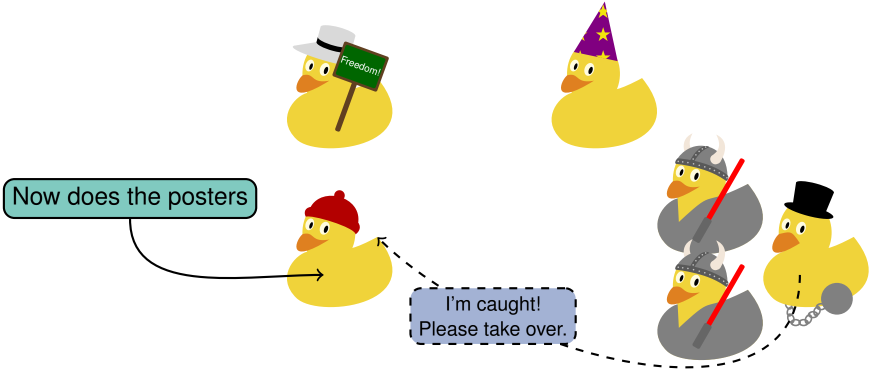
Christoph Coijanovic, Christiane Weis, and Thorsten Strufe | 09/25/2023

Motivation



Political activists

Motivation



Requirements

Hidden Receiver

even from sender

Caught activist cannot disclose their successor if they do not know them.

Requirements

Hidden Receiver

even from sender

Caught activist cannot disclose their successor if they do not know them.

Limited Receivers

chosen by sender

The successor has to be one of the activists.

Requirements

Hidden Receiver

even from sender

Caught activist cannot disclose their successor if they do not know them.

Limited Receivers

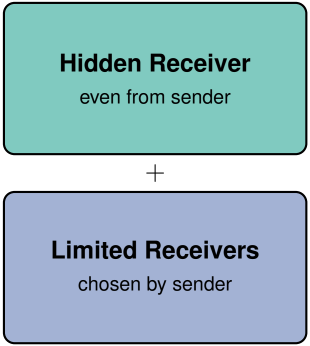
chosen by sender

The successor has to be one of the activists.

Easy Setup

for sender & receivers

Expensive setup limits adoption.



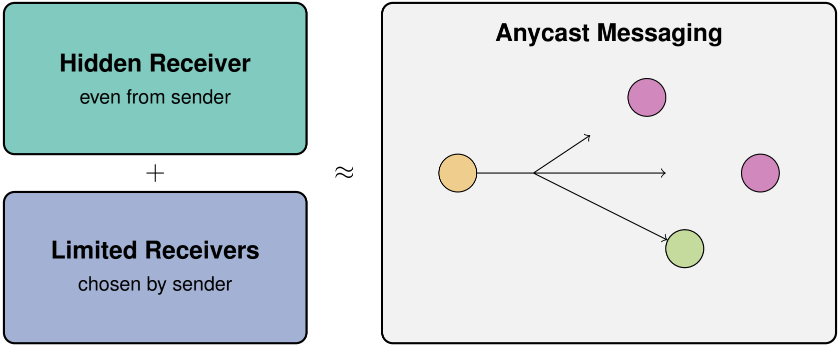
Hidden Receiver

even from sender

+

Limited Receivers

chosen by sender



Hidden Receiver

even from sender

$+$

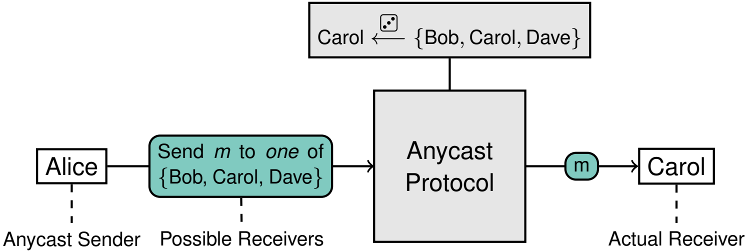
Limited Receivers

chosen by sender

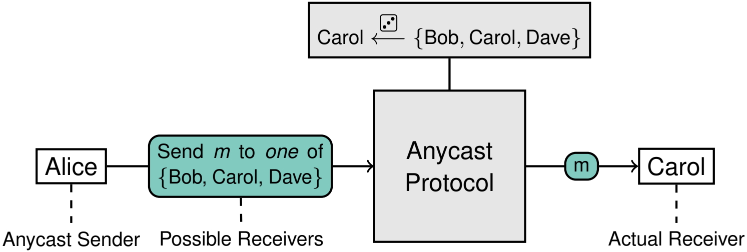
$\approx$

Anycast Messaging

Functionality & Adversary


Carol $\xleftarrow{⚂}$ {Bob, Carol, Dave}
Anycast Protocol
Alice
Send *m* to *one* of {Bob, Carol, Dave}
m
Carol
Anycast Sender
Possible Receivers
Actual Receiver

Functionality & Adversary


Carol $\xleftarrow{⚂}$ {Bob, Carol, Dave}
Alice
Send *m* to *one* of {Bob, Carol, Dave}
Anycast Protocol
m
Carol
Anycast Sender
Possible Receivers
Actual Receiver

Adversary

- Global passive observation
- Active interference on all network links
- honest-but-curious sender
- honest-but-curious fraction of possible receivers

Goals Overview

Message Confidentiality

Outside of sender and receiver, nobody shall learn information about the message.

Goals Overview

Message Confidentiality

Outside of sender and receiver, nobody shall learn information about the message.

Receiver Anonymity

Any adversary shall only learn trivial information about actual receivers.

Goals Overview

Message Confidentiality

Outside of sender and receiver, nobody shall learn information about the message.

Receiver Anonymity

Any adversary shall only learn trivial information about actual receivers.

Fairness

Any possible receivers shall be equally likely to be chosen as actual receiver.

Goals Overview

Message Confidentiality

Outside of sender and receiver, nobody shall learn information about the message.

Receiver Anonymity

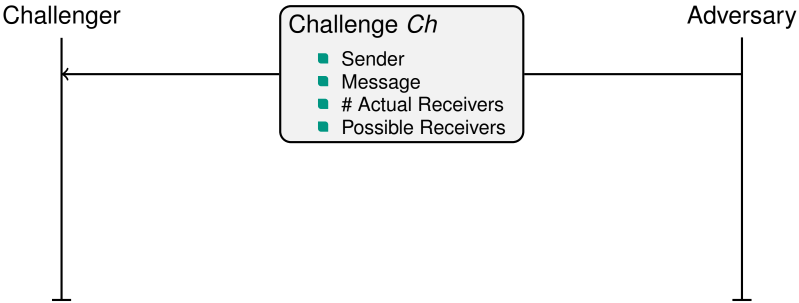
Any adversary shall only learn trivial information about actual receivers.

Fairness

Any possible receivers shall be equally likely to be chosen as actual receiver.

Receiver Anonymity

Goal: Guess an actual receiver.

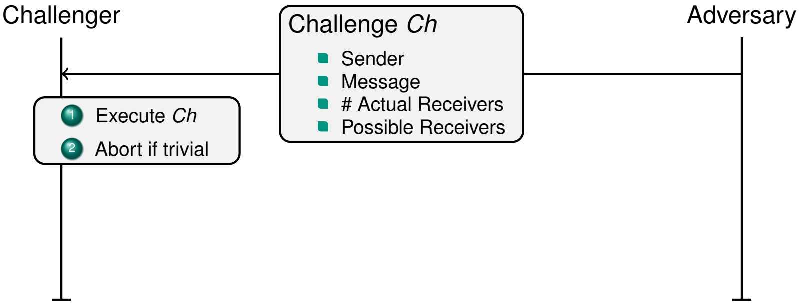


Challenge *Ch*

- Sender
- Message
- # Actual Receivers
- Possible Receivers

Receiver Anonymity

Goal: Guess an actual receiver.



Challenger

Adversary

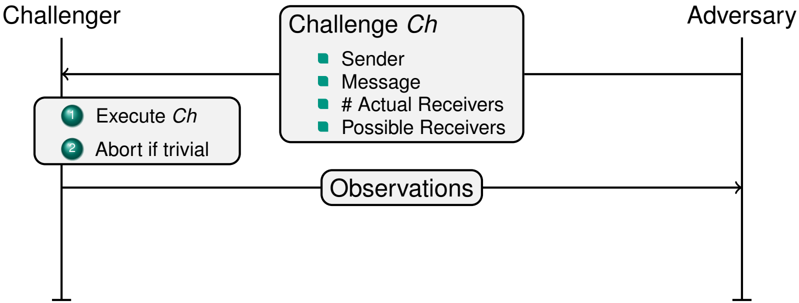
Challenge *Ch*

- Sender
- Message
- # Actual Receivers
- Possible Receivers

1. Execute *Ch*
2. Abort if trivial

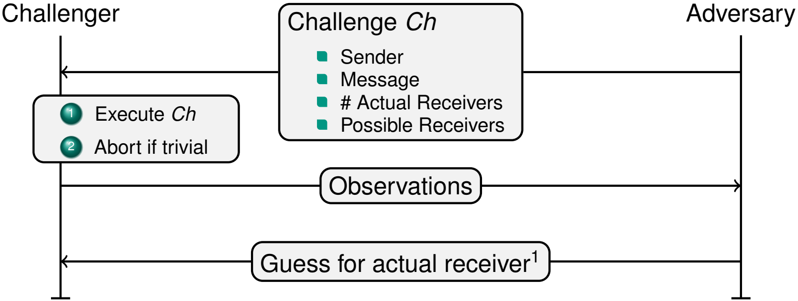
Receiver Anonymity

Goal: Guess an actual receiver.



Receiver Anonymity

Goal: Guess an actual receiver.



[1] **Or** unveil and try again.

Panini – Prerequisites

Secure Channel

- Anycast sender ↔ all possible receivers
- Confidential and authenticated
- Example: Anything with encryption and signatures, e.g., Email + S/MIME

Panini – Prerequisites

Secure Channel

- Anycast sender ↔ all possible receivers
- Confidential and authenticated
- Example: Anything with encryption and signatures, e.g., Email + S/MIME

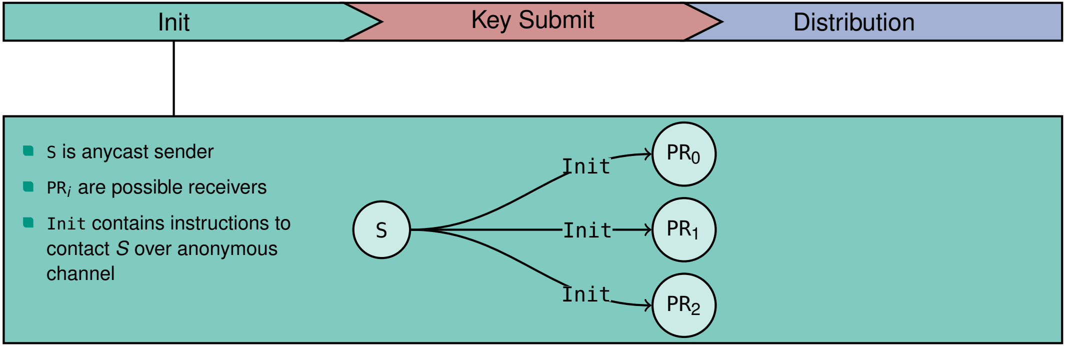
Anonymous Channel

- Every possible receiver → anycast sender
- Unlinks senders from their messages
- Example: Tor[a] against non-global adversaries or Nym[b]

[a] torproject.org

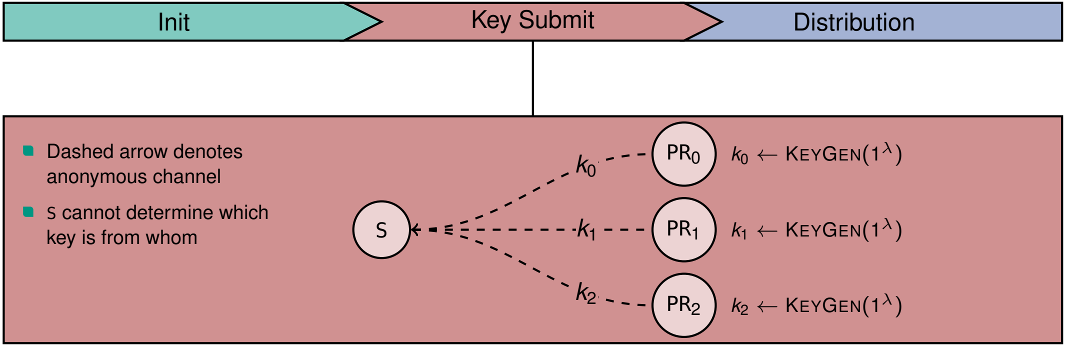
[b] nymtech.net

Panini – Base Protocol



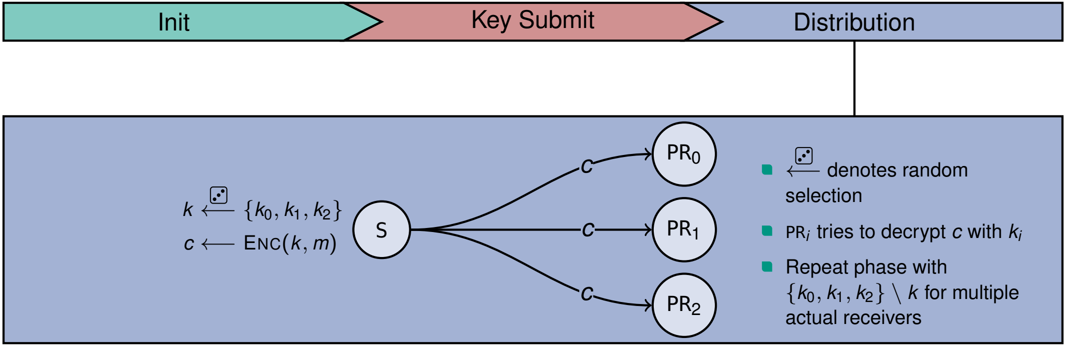
- S is anycast sender
- PR_i are possible receivers
- Init contains instructions to contact S over anonymous channel

Panini – Base Protocol


Init
Key Submit
Distribution
S
k_0
k_1
k_2
PR_0
PR_1
PR_2
$k_0 \leftarrow \textsc{KeyGen}(1^\lambda)$
$k_1 \leftarrow \textsc{KeyGen}(1^\lambda)$
$k_2 \leftarrow \textsc{KeyGen}(1^\lambda)$

- Dashed arrow denotes anonymous channel
- S cannot determine which key is from whom

Panini – Base Protocol



$k \xleftarrow{⚂} \{k_0, k_1, k_2\}$

$c \longleftarrow \textsc{Enc}(k, m)$

- $\xleftarrow{⚂}$ denotes random selection
- PR_i tries to decrypt c with k_i
- Repeat phase with $\{k_0, k_1, k_2\} \setminus k$ for multiple actual receivers

Defending against Active Adversaries

Active Key Replacement Attack

- Goal: Gain knowledge of anycast message

Defending against Active Adversaries

Active Key Replacement Attack

- Goal: Gain knowledge of anycast message
- Approach:

Defending against Active Adversaries

Active Key Replacement Attack

- Goal: Gain knowledge of anycast message
- Approach:
  - Key Submit Replace possible receiver's keys with own keys

Defending against Active Adversaries

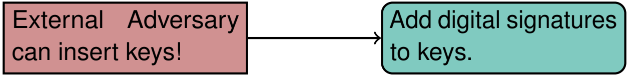
Active Key Replacement Attack

- Goal: Gain knowledge of anycast message
- Approach:
  - Key Submit Replace possible receiver's keys with own keys
  - Distribution Intercept and decrypt ciphertext

Defending against Active Adversaries

External Adversary can insert keys!

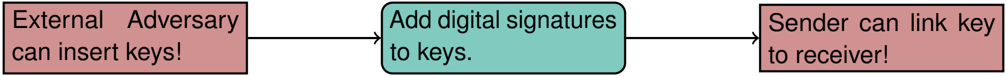
Defending against Active Adversaries



External Adversary can insert keys!

Add digital signatures to keys.

Defending against Active Adversaries

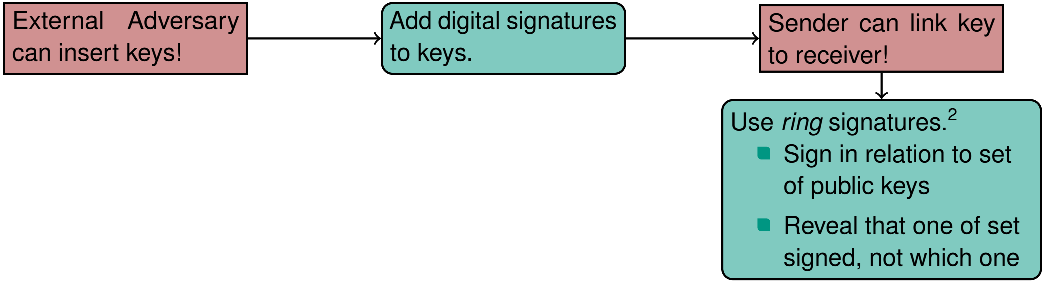


External Adversary can insert keys!

Add digital signatures to keys.

Sender can link key to receiver!

Defending against Active Adversaries



External Adversary can insert keys!

Add digital signatures to keys.

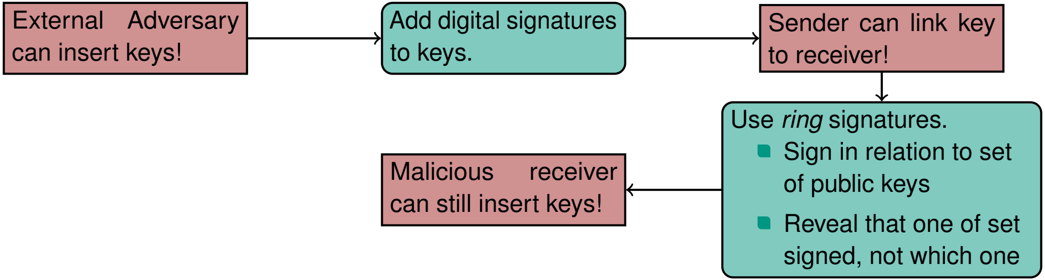
Sender can link key to receiver!

Use *ring* signatures.[2]

- Sign in relation to set of public keys
- Reveal that one of set signed, not which one

[2] Rivest, Ronald L. et al. “How to Leak a Secret.” ASIACRYPT 2001.

Defending against Active Adversaries



External Adversary can insert keys!

Add digital signatures to keys.

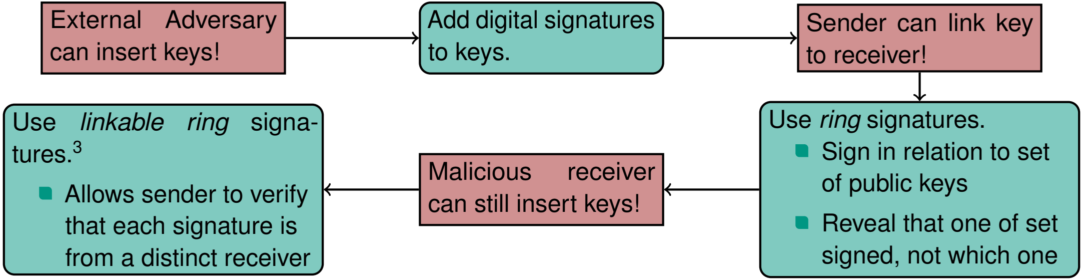
Sender can link key to receiver!

Use *ring* signatures.

- Sign in relation to set of public keys
- Reveal that one of set signed, not which one

Malicious receiver can still insert keys!

Defending against Active Adversaries



External Adversary can insert keys!

Add digital signatures to keys.

Sender can link key to receiver!

Use *ring* signatures.

- Sign in relation to set of public keys
- Reveal that one of set signed, not which one

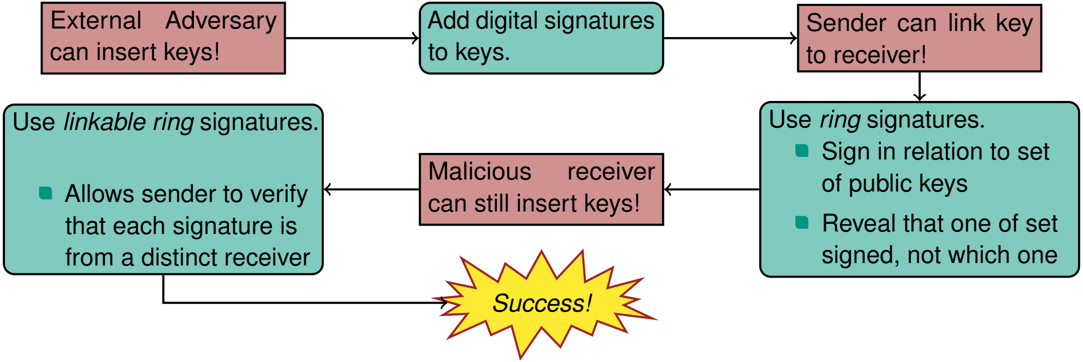
Malicious receiver can still insert keys!

Use *linkable ring* signatures.[3]

- Allows sender to verify that each signature is from a distinct receiver

[3] Liu, Joseph K. et al. “Linkable Spontaneous Anonymous Group Signature for Ad Hoc Groups (Extended Abstract).” IACR Cryptol. ePrint Arch. 2004

Defending against Active Adversaries



External Adversary can insert keys!

Add digital signatures to keys.

Sender can link key to receiver!

Use *ring* signatures.

- Sign in relation to set of public keys
- Reveal that one of set signed, not which one

Malicious receiver can still insert keys!

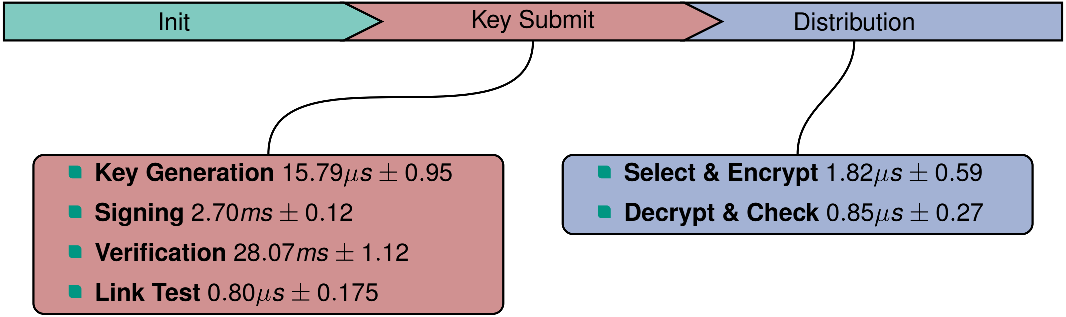
Use *linkable ring* signatures.

- Allows sender to verify that each signature is from a distinct receiver

Success!

Evaluation

Computational Overhead (2 CPU cores // 10 possible receivers // 1 KB message)

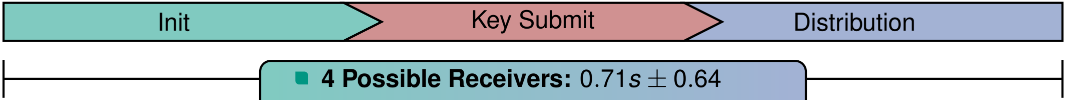


- **Key Generation** $15.79\mu s \pm 0.95$
- **Signing** $2.70ms \pm 0.12$
- **Verification** $28.07ms \pm 1.12$
- **Link Test** $0.80\mu s \pm 0.175$

- **Select & Encrypt** $1.82\mu s \pm 0.59$
- **Decrypt & Check** $0.85\mu s \pm 0.27$

Evaluation

End-to-end Latency (Secure Channel: AES+ECDSA // Anon. Channel: Nym // 512 B Message)



Init

Key Submit

Distribution

- **4 Possible Receivers:** $0.71s \pm 0.64$
- **8 Possible Receivers:** $0.76s \pm 0.34$
- **16 Possible Receivers:** $0.82s \pm 2.13$

Summary

We introduced Panini, an *anonymous anycast* protocol.

Panini is secure

First protocol to hide to hide the receiver from all entities including the sender.

Panini is efficient

<1s end-to-end latency and <30ms computation for sender.

Summary

We introduced Panini, an *anonymous anycast* protocol.

Panini is secure

First protocol to hide to hide the receiver from all entities including the sender.

Panini is efficient

<1s end-to-end latency and <30ms computation for sender.