Random-Walk Key Relaying
for Decentralized QKD Networks

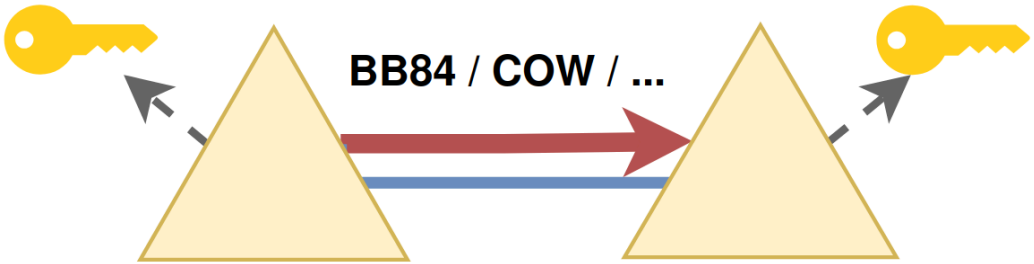
Krišjānis Petručeņa, Sergejs Kozlovičs

Joint Estonian-Latvian Theory Days, 2026-04-23

1. QKD Networks

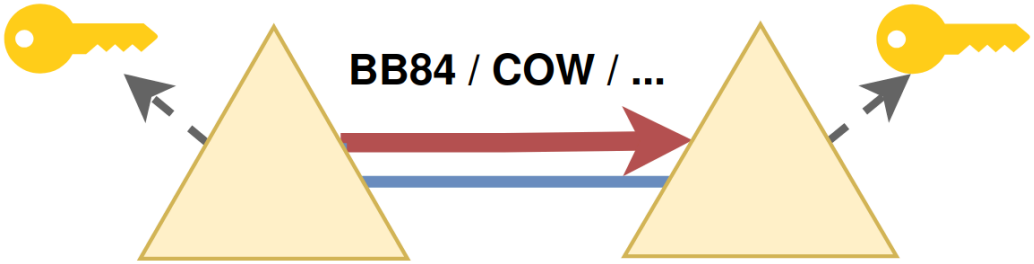
1.1 Quantum Key Distribution

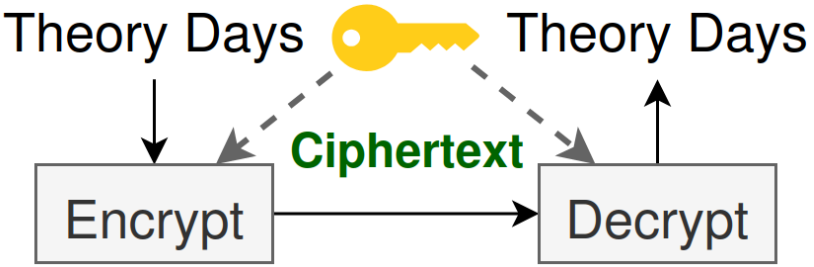
Quantum key distribution (QKD) is a complementary approach to PQC.
It is a provable means of **detecting eavesdropping**.



1.1 Quantum Key Distribution

Quantum key distribution (QKD) is a complementary approach to PQC.
It is a provable means of **detecting eavesdropping**.

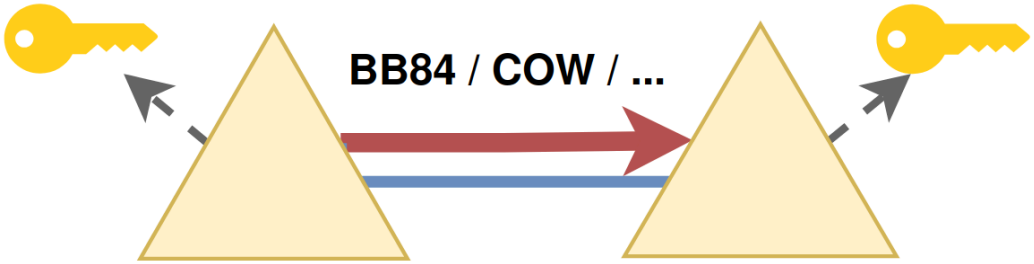


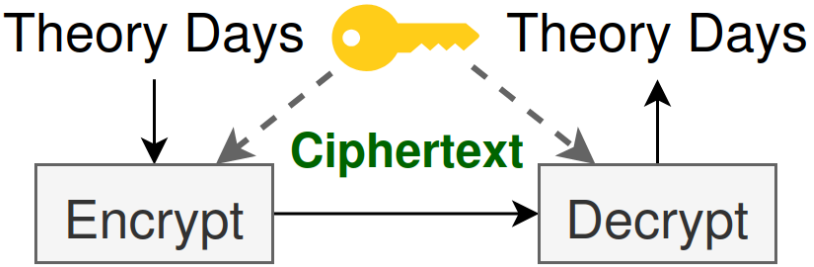


- **Information Theoretic Security** over asymmetric cryptography

1.1 Quantum Key Distribution

Quantum key distribution (QKD) is a complementary approach to PQC.
It is a provable means of **detecting eavesdropping**.



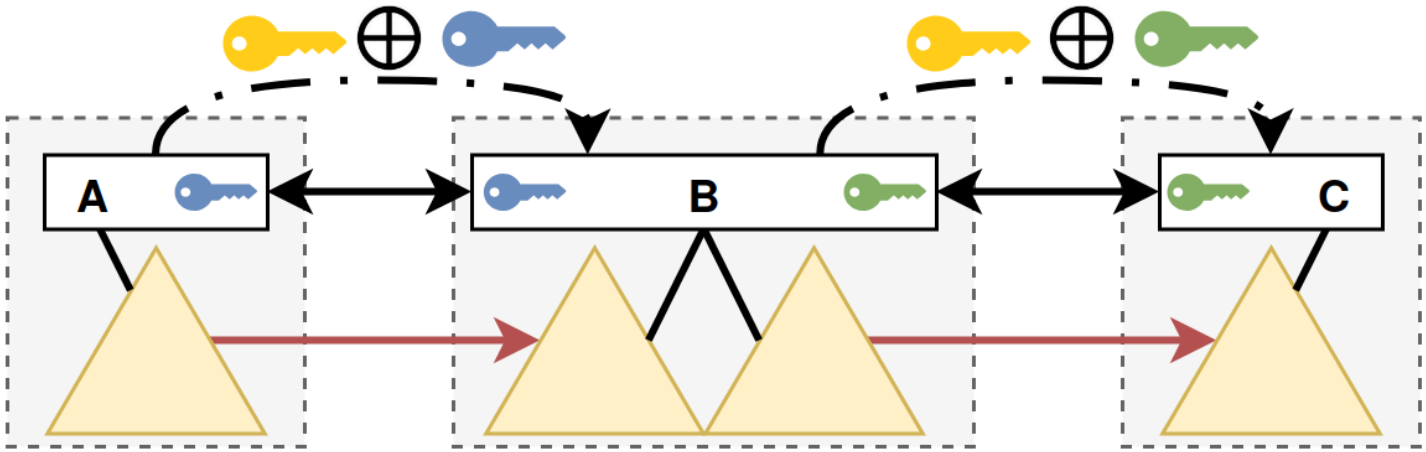


- **Information Theoretic Security** over asymmetric cryptography
- *Expensive* hardware requirements.
- Practical limitations: ***distance*** ($\approx$ 150 km) and key rate ($\approx$ 1 kbit/s).

1.2 Hop-by-hop Key Relaying

Relay hop-by-hop using **OTP** encryption to maintain **ITS**.

A generates a new key and *kindly* asks **B** to discreetly forward it to **C**.



Blue and green keys are *link* keys. Yellow key is the *end-to-end* key.

1.3 Real World Deployments

Among others, the Polish PIONIER-Q network of the EuroQCI project.

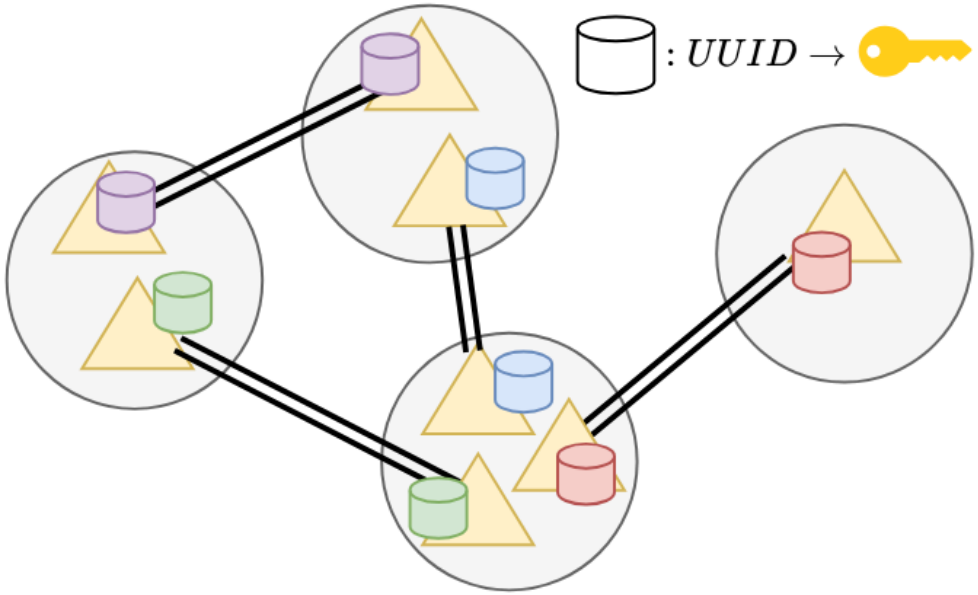
1.3 Real World Deployments

Among others, the Polish PIONIER-Q network of the EuroQCI project.

Largest one is in China and has 145 nodes but it has a star-like topology.

1.4 QKD Network Model

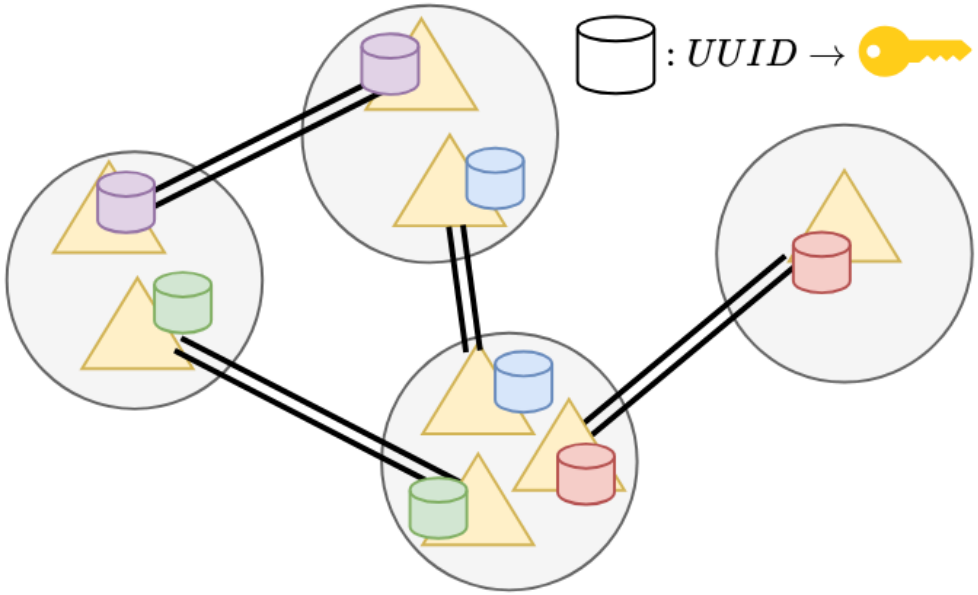
$G = (V, E)$. V (vertices) - trusted nodes, E (edges) - QKD links.

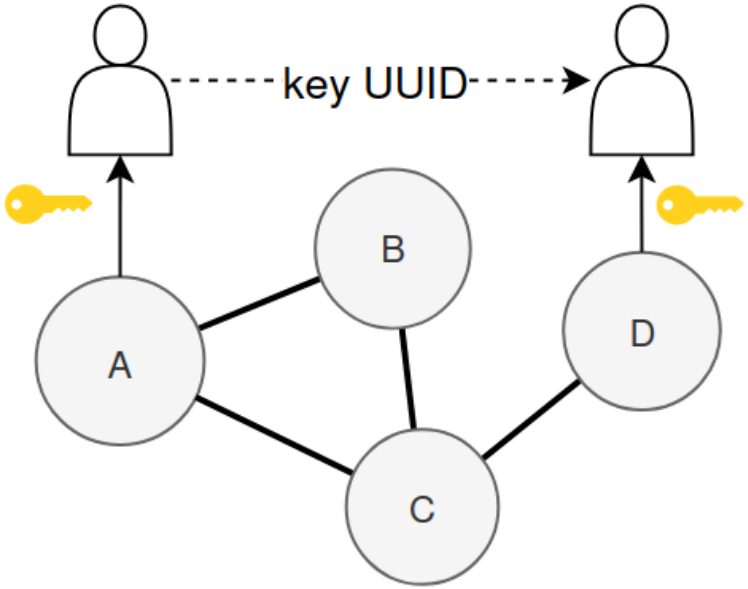


Link keys are continously buffered.

1.4 QKD Network Model

$G = (V, E)$. V (vertices) - trusted nodes, E (edges) - QKD links.

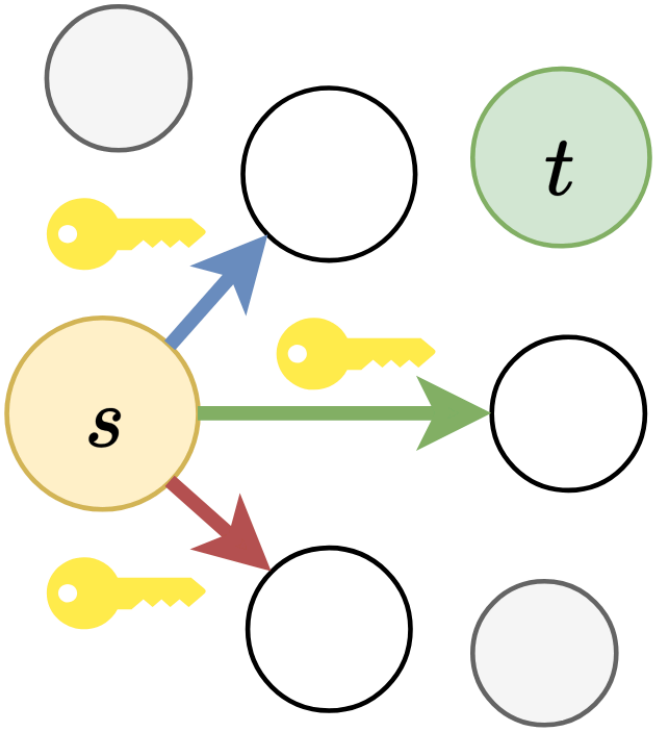




Link keys are continously buffered.

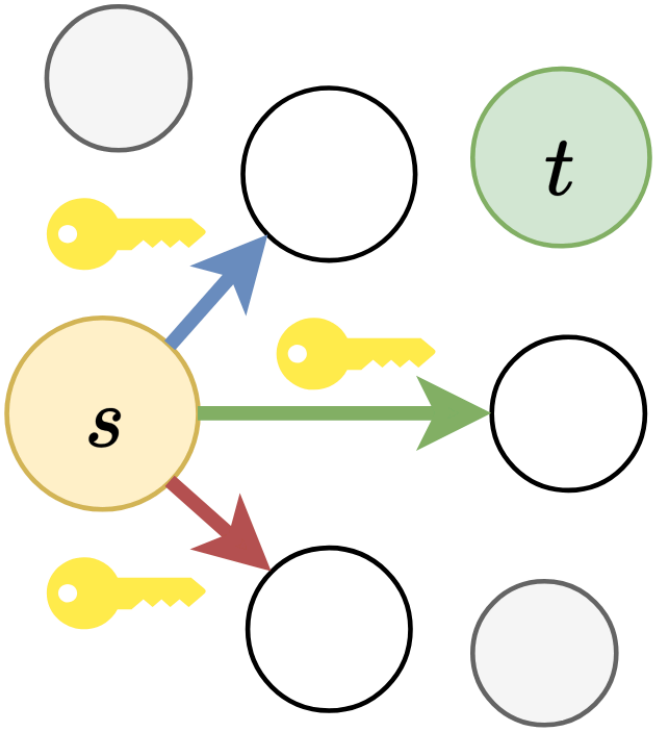
Goal: Supply end-to-end keys.

1.5 The Routing Question



Routing problem solutions

1.5 The Routing Question

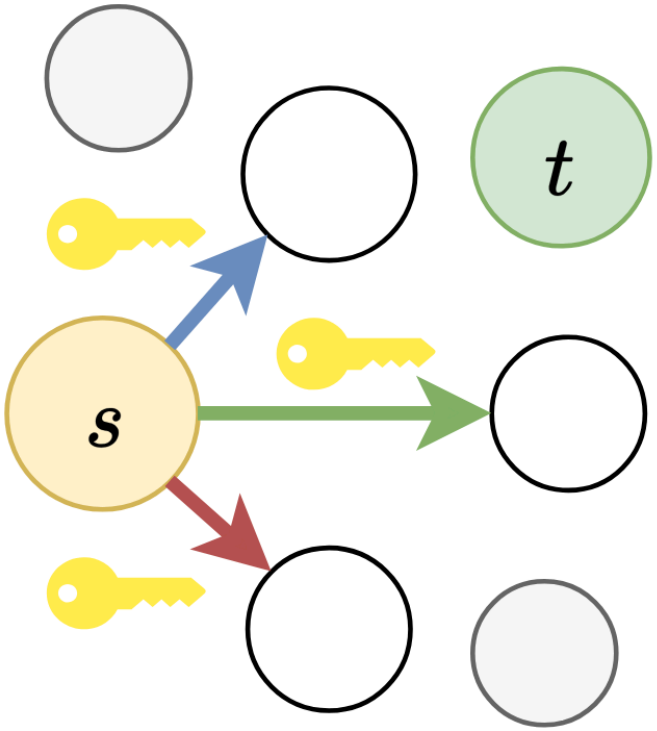


Routing problem solutions

1. Statically configured routes / topology

Rollout friction

1.5 The Routing Question



Routing problem solutions

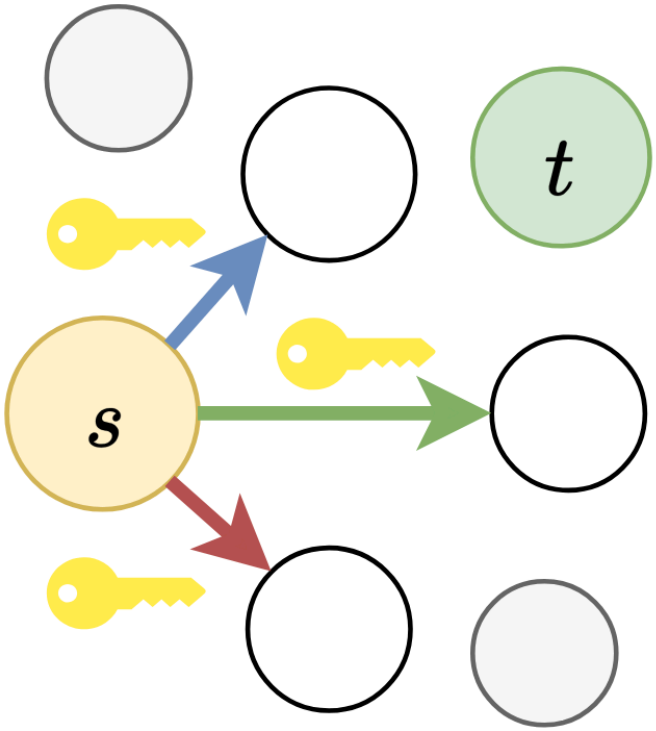
1. Statically configured routes / topology

Rollout friction

2. Gossip-like discovery (e.g. OSPF LSA)

Complexity, attack surface

1.5 The Routing Question



Routing problem solutions

1. Statically configured routes / topology

Rollout friction

2. Gossip-like discovery (e.g. OSPF LSA)

Complexity, attack surface

3. Centralized SDN controller

Single point of failure

1.6 Partial-trust in Networks

Can we trust the “trusted” nodes?

- Network administrators
- Geopolitical adversaries
- QKD hardware vendors
- Underlying OS, software

We *should* alleviate compromise.

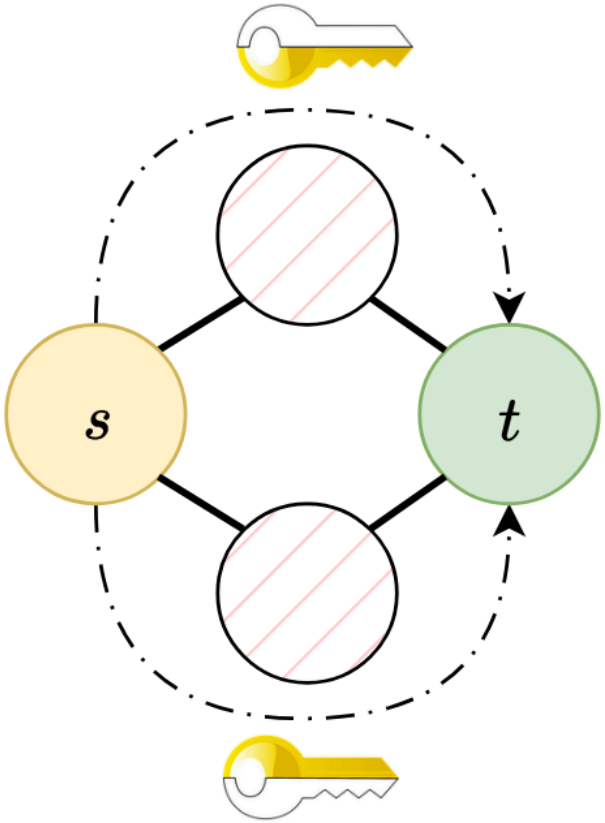
1.6 Partial-trust in Networks

Can we trust the “trusted” nodes?

- Network administrators
- Geopolitical adversaries
- QKD hardware vendors
- Underlying OS, software

We *should* alleviate compromise.

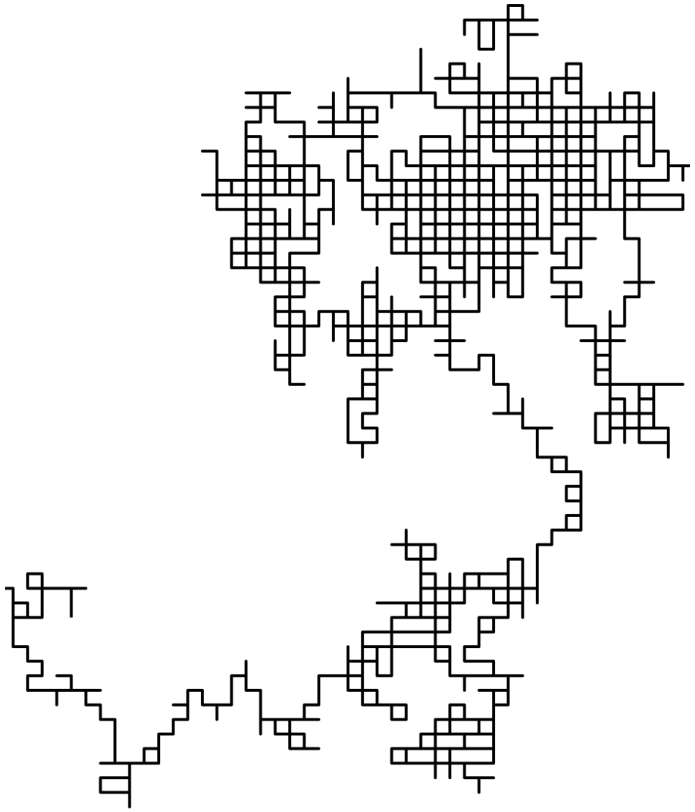
Implement **multi-path routing**!



2. Random Flow Proposal

2.1 Random Walk Relaying

Competitive throughput and security can be achieved in a **topology-oblivious setting** with random-walk forwarding!



2.2 Random Walk Variants

Goal: fewer revisits.

1. **R** (simple RW): uniform neighbor.
2. **NB** (non-backtracking): remember predecessor, don't backtrack.
3. **LRV** (least-recently-visited): pick the neighbor visited *least* recently. Prefer unvisited; otherwise prefer "older".

2.2 Random Walk Variants

Goal: fewer revisits.

1. **R** (simple RW): uniform neighbor.
2. **NB** (non-backtracking): remember predecessor, don't backtrack.
3. **LRV** (least-recently-visited): pick the neighbor visited *least* recently. Prefer unvisited; otherwise prefer "older".

Mean hops on G'EANT (no loop-erasure):

Variant	Mean hops
R	64
NB	32
LRV	18

LRV: $\approx 3.6 \times$ fewer hops

2.3 Addressing Malicious Nodes

Threat model: exactly 1 node is compromised.

Random-walk + multi-path:

1. Split key material into M 256-bit fragments
2. Route them *randomly* from S to T
3. Compress received M fragments into θ keys

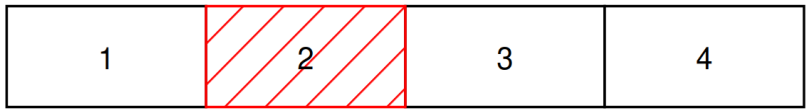
Sort of looks like a *random flow*.

What's a secure θ if T receives M chunks?

2.4 Estimating Secure Entropy

Let X be the worst-case probability of a chunk being eavesdropped.

$$\mathrm{X} = \max_{v \in V \setminus \{s,t\}} P(v), \;\; P(v) = \Pr[v \text{ is visited}]$$



$$P(g \geq G) = \sum_{i=G}^{M} \binom{M}{i} (1 - \mathrm{X})^i \mathrm{X}^{M-i}$$

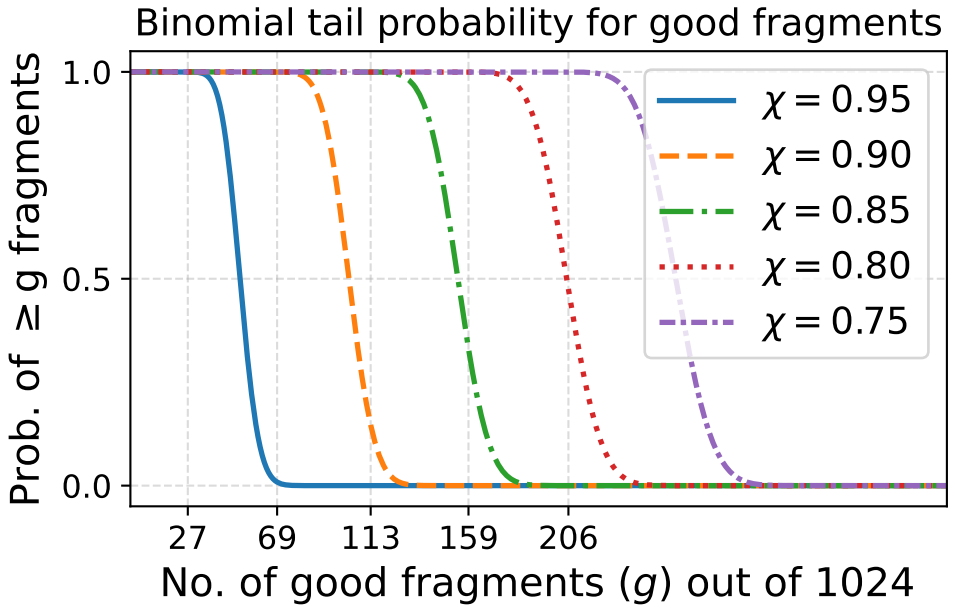
g - observed good fragment count, G - guaranteed lower bound

2.5 Estimating Secure Entropy

Given X, what is the θ no of keys we can extract from M fragments?

If X $= 95\%$, with 99.99% confidence, we can extract $\geq$ 27 keys out of $M = 1024$.

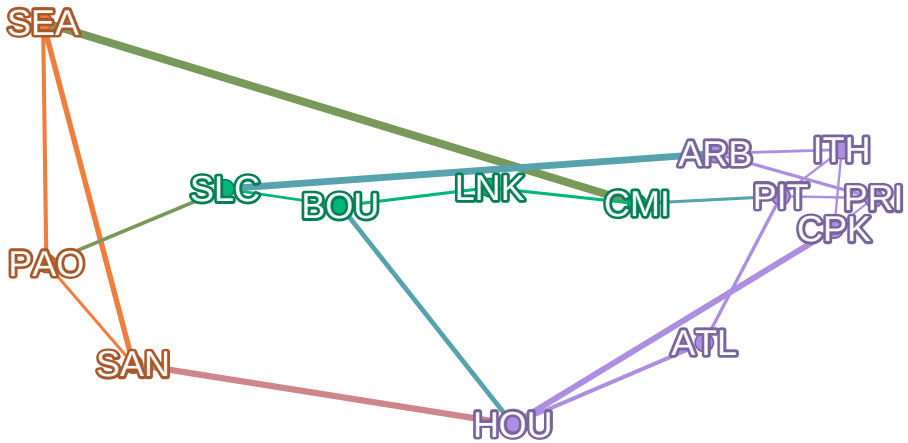
If X $= 75\%$, ..., we can extract 206 keys out of 1024.

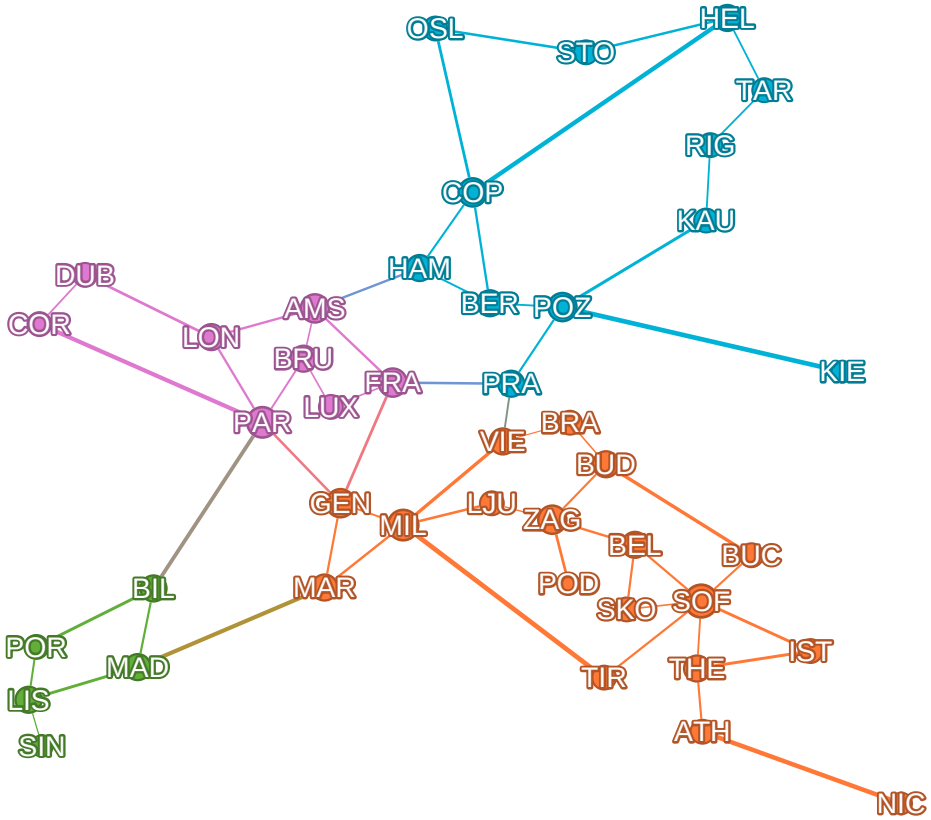


2.6 Empirical Evaluation

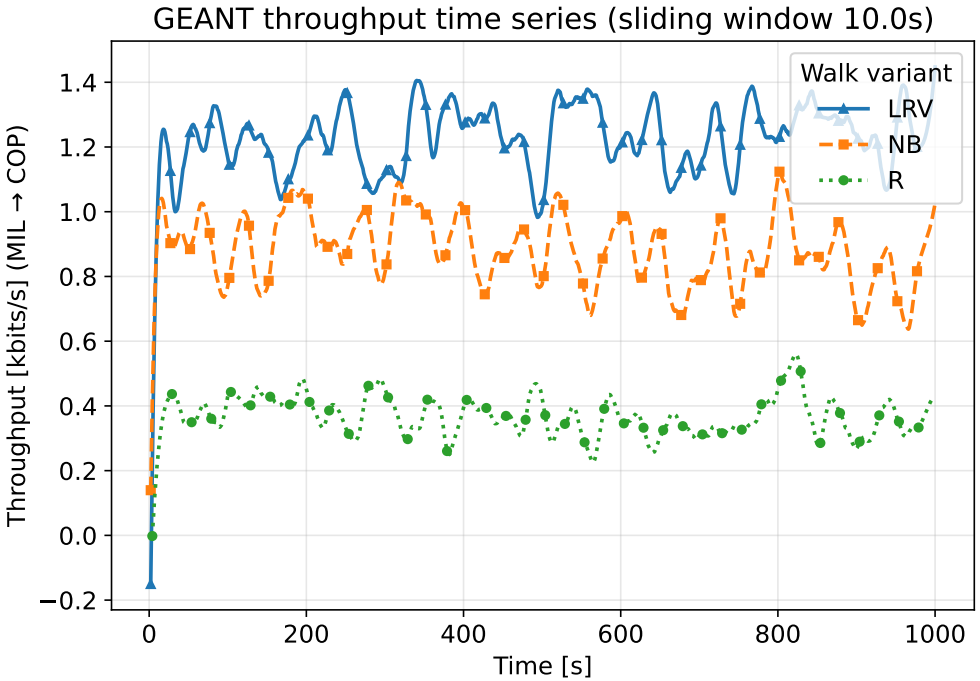
Two realistic future topologies:

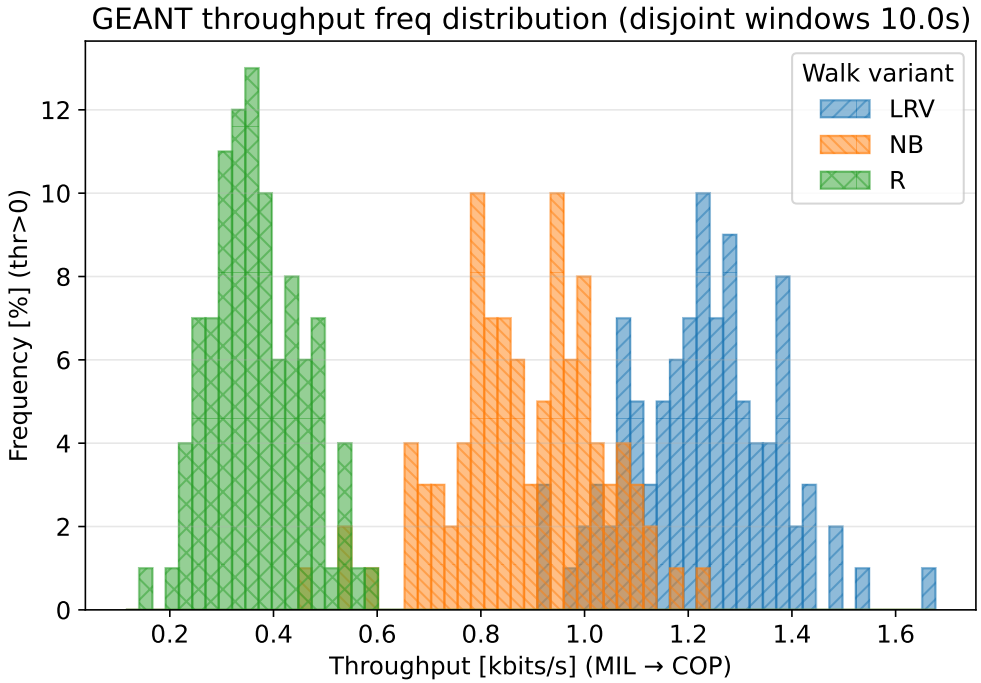
- NSFnet (14 nodes, 21 edges)
- GÉANT (43 nodes, 59 edges)





2.7 Throughput Results





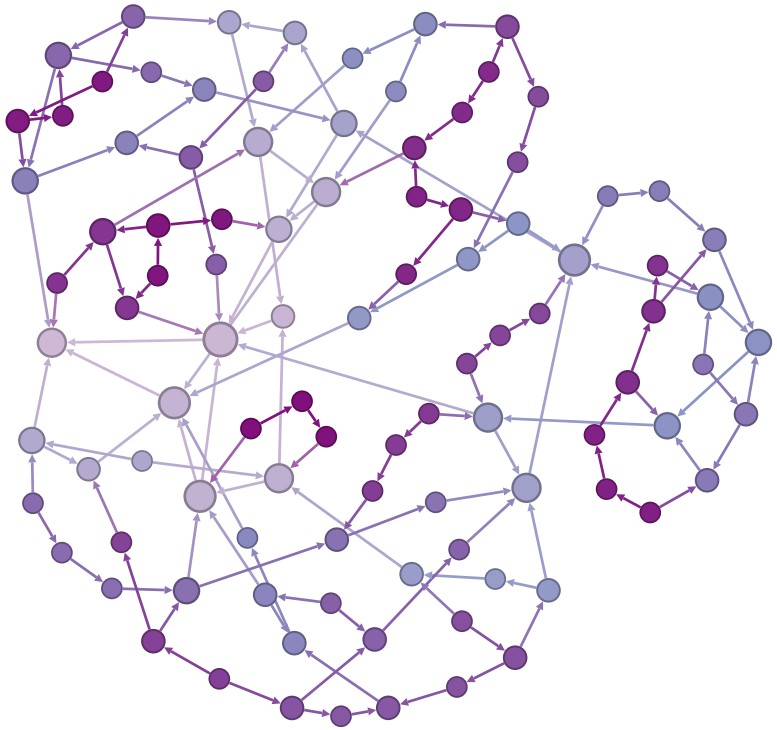
RW exceeds the throughput of single shortest-path routing!

2.8 Exposure Results

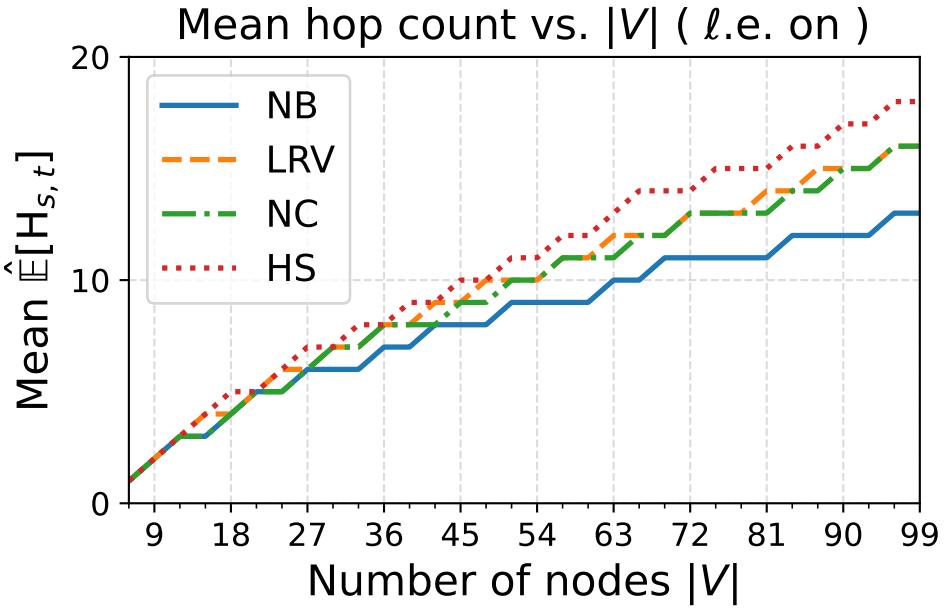
Average exposure for LRV is 72.5% vs 79.4% for R.

Variant	Max $\hat{\chi}$	s	t	v	Avg $\hat{\chi}$	Median $\hat{\chi}$
R	99.2	TIR	LIS	MIL	79.4	89.7
NB	96.4	TIR	LIS	MIL	74.1	83.5
LRV	96.1	TIR	LIS	MIL	72.5	81.1
NC	93.7	POR	COR	PAR	71.8	80.4
HS	92.6	MAD	COR	PAR	69.4	77.2

2.9 Does it Scale?

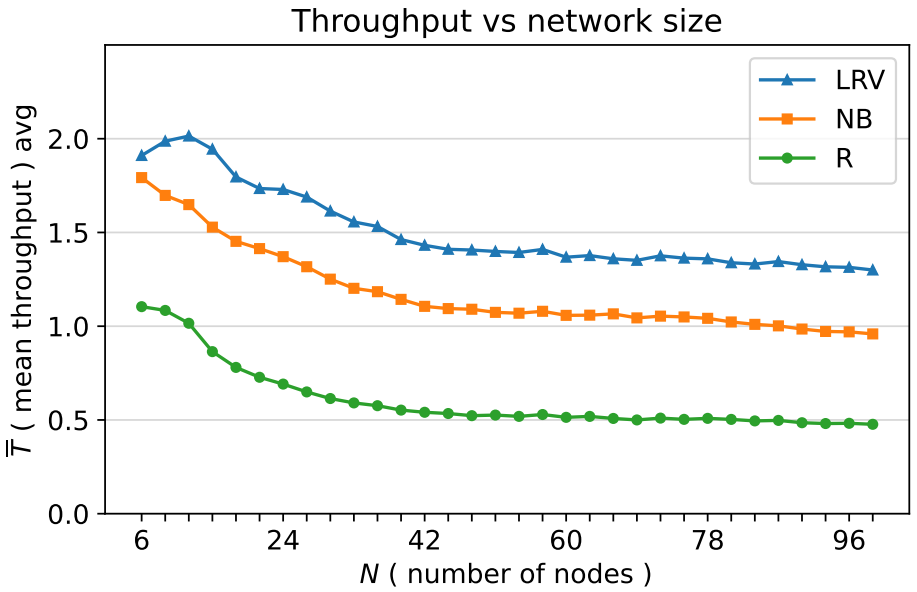


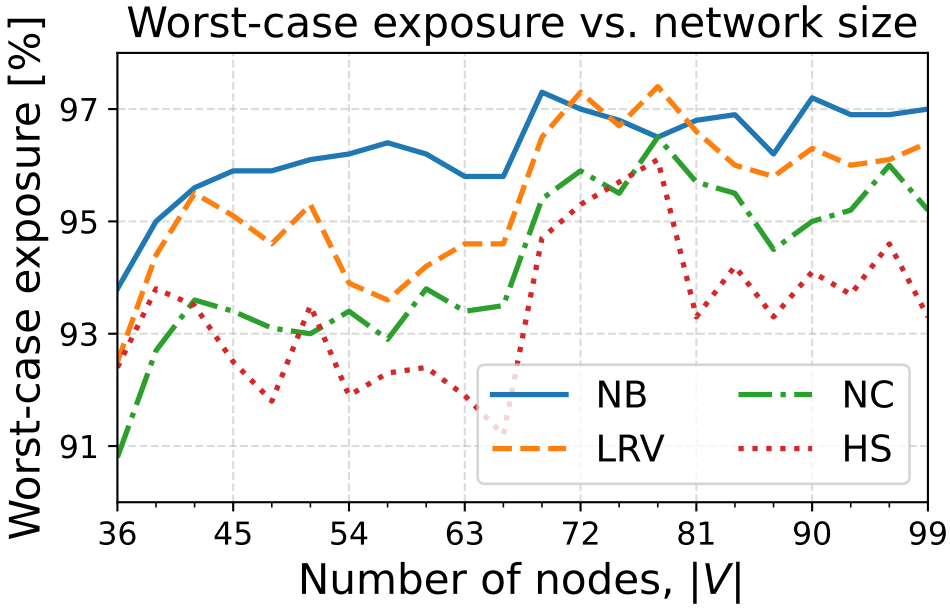
Expected hop count for a random pair of nodes scales linearly!



2.10 Does it Scale?

Throughput scales sublinearly! Worst-case X and $|V|$ is unrelated.

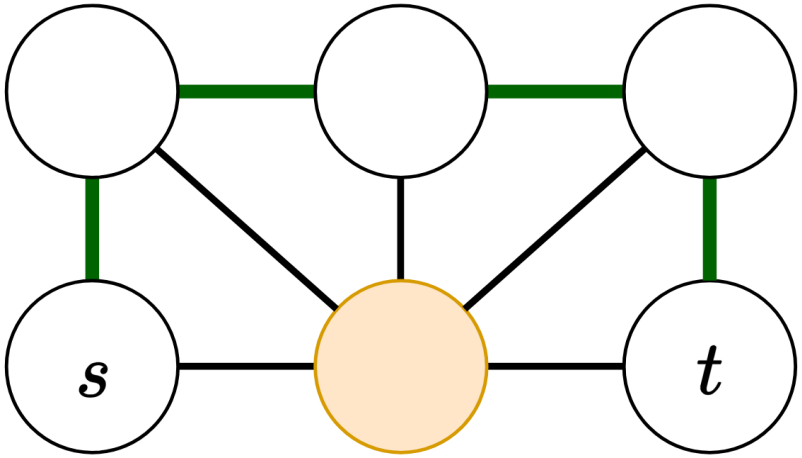




3. Further Optimization

3.1 Reducing Exposure

R, NB, LRV counterexample. Deviation to central node is likely.

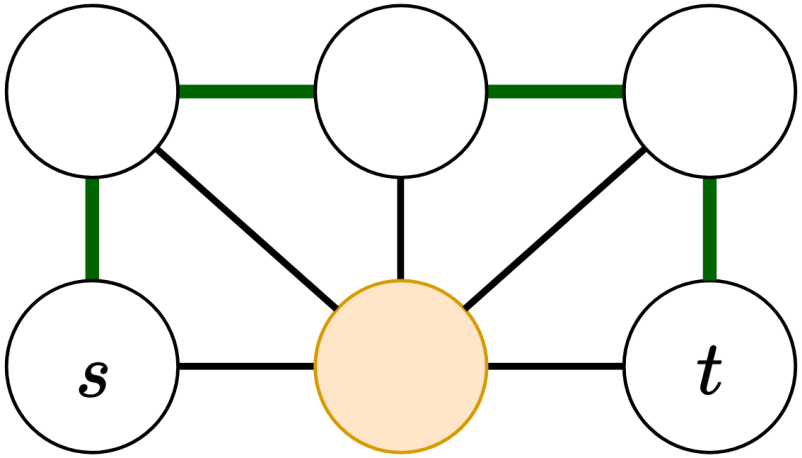


$$X = 1 - \left(\frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2}\right)$$

$$X \approx 93.75\%$$

3.1 Reducing Exposure

R, NB, LRV counterexample. Deviation to central node is likely.



$$X = 1 - \left(\frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2}\right)$$

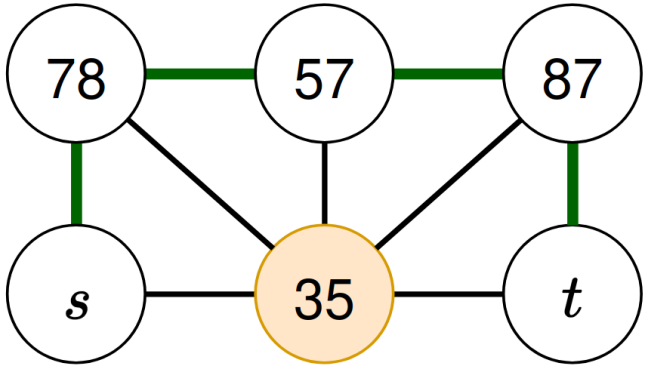
$$X \approx 93.75\%$$

Node coloring: we *disable* a node one by one for each fragment.

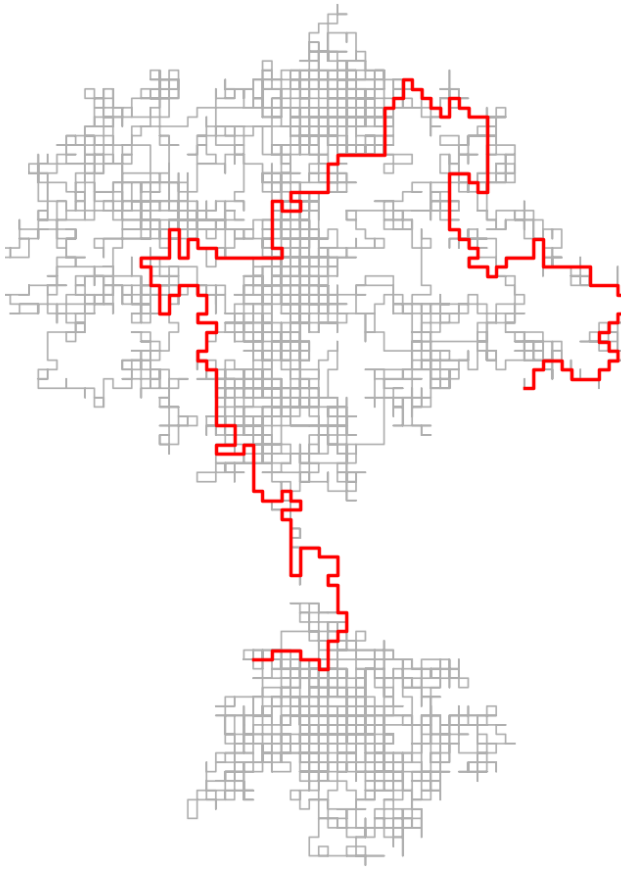
3.2 Highest Score Neighbor

HS (highest-score neighbor) is a seed-based diversification heuristic.

- Each fragment token i samples a fresh seed ζ_i.
- The seed defines a score for each node: $\text{score}_{i(u)} = h(u, \zeta_i)$
- We choose the neighbor with the highest score



3.3 Reducing QKD Key Waste



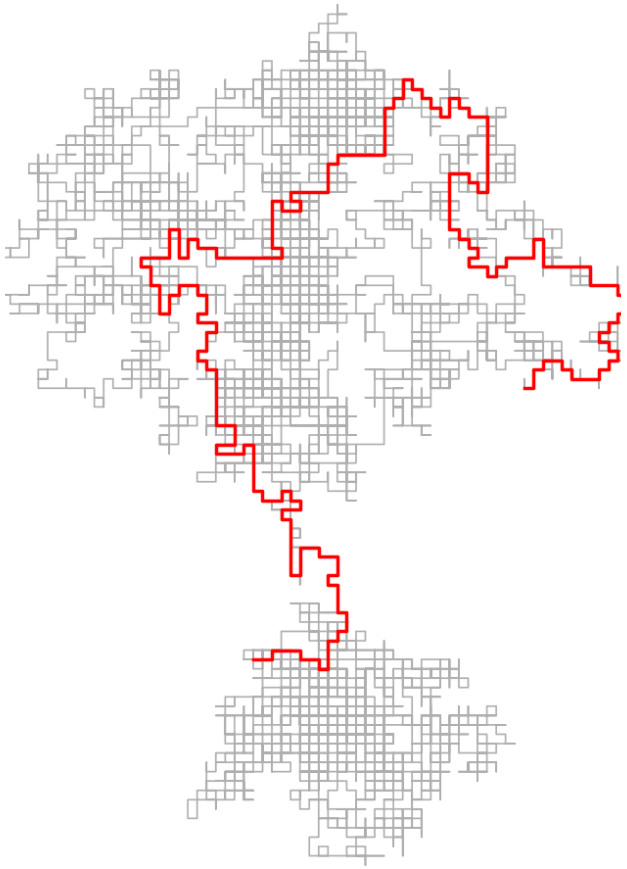
Scout with a random w., then **erase loops**!

Paper result: on **GÉANT** $\approx 80\%$ fewer hops:

- $33 \rightarrow 7$ for non-backtracking RW

Directly translates into much higher throughput! Competitive with shortest-path.

3.3 Reducing QKD Key Waste



Scout with a random w., then **erase loops**!

Paper result: on **GÉANT** $\approx 80\%$ fewer hops:

- $33 \rightarrow 7$ for non-backtracking RW

Directly translates into much higher throughput! Competitive with shortest-path.

Another important bonus:

Congestion control - better error handling on overloaded network. `503` status code.

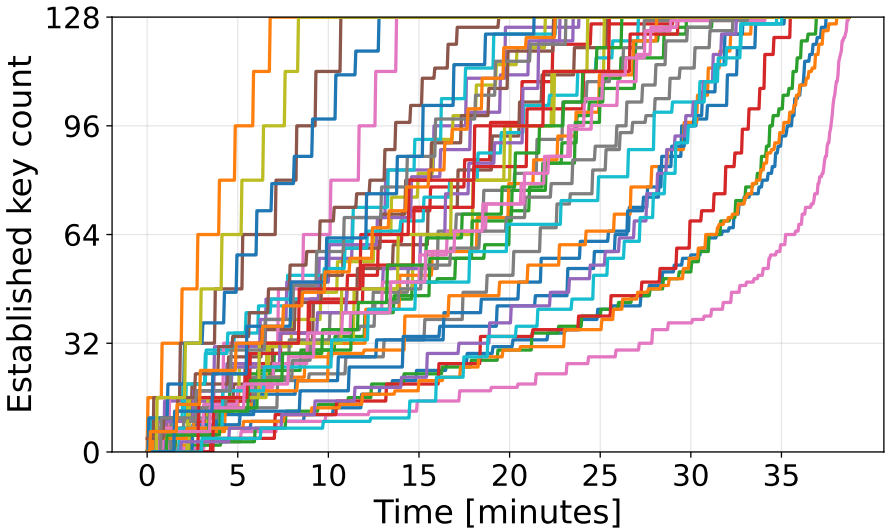
3.4 Signing History

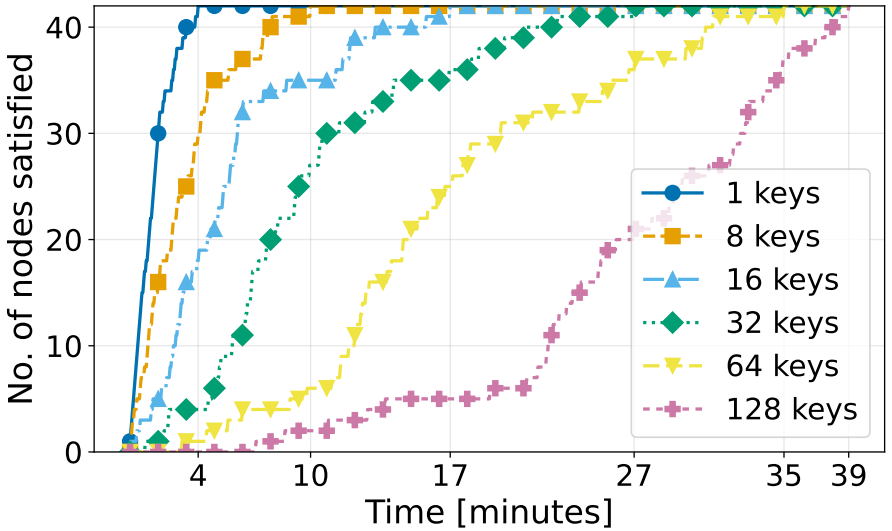
Maintain history. Digitally sign each next hop.

3.5 Proactive Relaying

Don’t set the target. Probabilistic consumption

$$p = 1 - \max(\text{buffer}, \text{ttl remaining}).$$





3.6 Thank you! Questions?

Would you tell me, please, which way I ought to go from here?”

“That depends a good deal on where you want to get to,” said the Cat.

“I don’t much care where—” said Alice.

“Then it doesn’t matter which way you go,” said the Cat.

“—so long as I get somewhere,” Alice added as an explanation.

“Oh, you’re sure to do that,” said the Cat, “if you only walk long enough.”