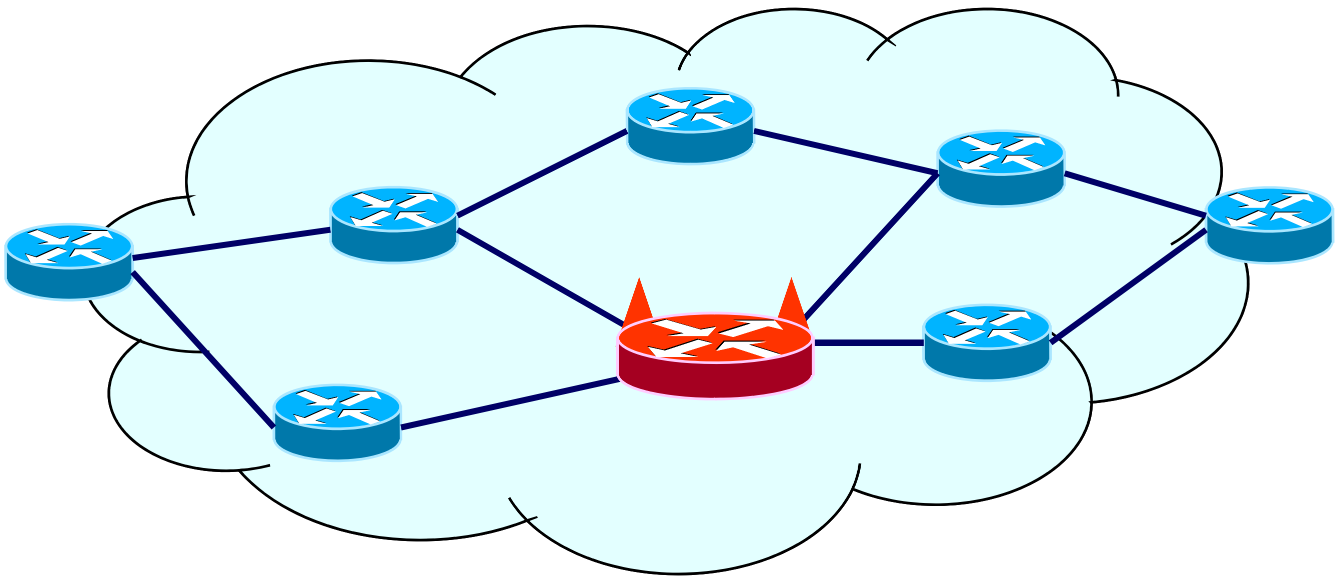
Internet Path-Quality Monitoring in the Presence of Adversaries



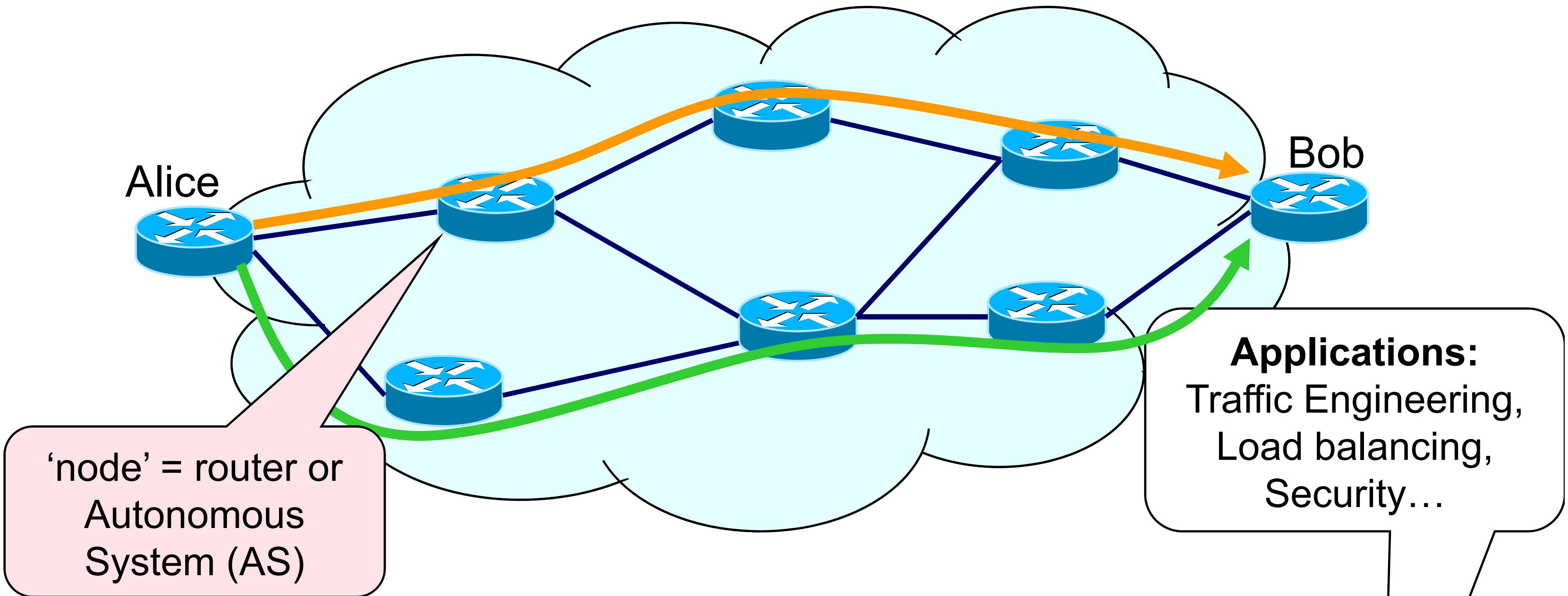
Sharon Goldberg

David Xiao, Eran Tromer, Boaz Barak, Jennifer Rexford

Princeton University

Penn State University CS Seminar
November 29, 2007

Applications of path-quality monitoring



Routers need tools to detect unacceptably high packet loss rates for…

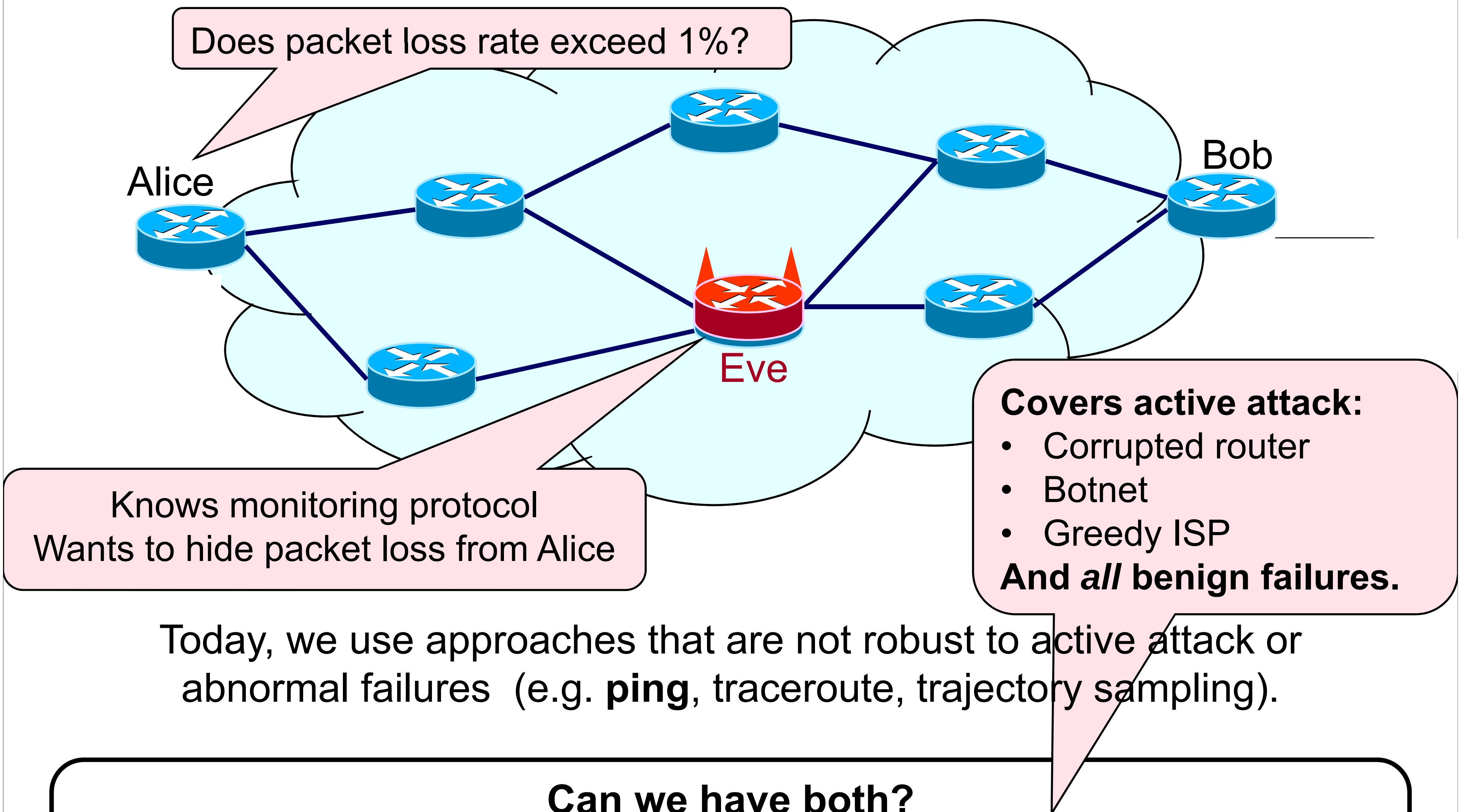
<u>Flexible Routing</u>

- Source routing: (Alice chooses nodes on path to Bob)
- Intelligent route control: (Switch paths based on performance)
- Overlay routing

<u>SLA compliance monitoring</u>

- Necessary to drive innovation! [LC06]

The presence of adversaries



Today, we use approaches that are not robust to active attack or abnormal failures (e.g. **ping**, traceroute, trajectory sampling).

Can we have both?

Strong threat model --- Eve can add/drop/delay/modify packets

Efficient protocols for high-speed routers

Design Goals

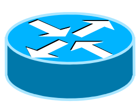
Secure path quality monitoring (PQM)

- Alice **alarms** if end-end packet loss rate exceeds **β**, *regardless of Eve's behavior*
- Alice **will not alarm** if packet loss rate is less than **α**

Only <u>detect</u> loss, not prevent loss**!**

Strong threat model

- Eve occupies node(s) on the path
- Knows the measurement protocol
- Can add, drop, delay, modify packets
- Can treat measurement packets preferentially
- Can collude with other nodes



Efficient protocols for high-speed routers

- Limited storage, computation, communication overhead
- Avoid marking or encrypting regular traffic

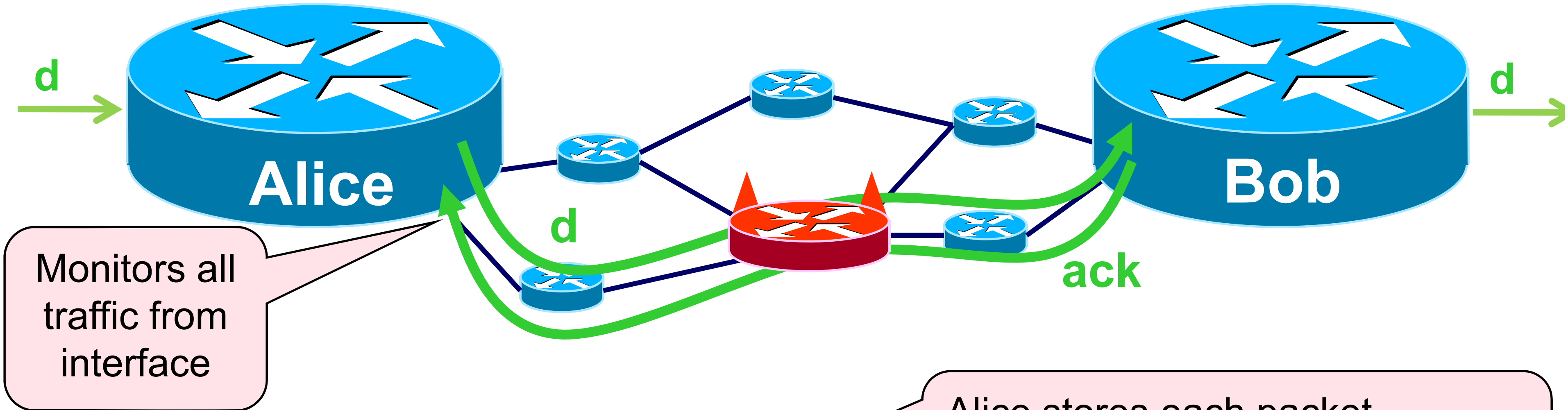
Can we have both?

(Yes)

This talk

1. Overview √
2. Secure Sketch PQM
3. Composing PQM to localize faulty links
4. Conclusion

Background: Secure Path Quality Monitoring (PQM)



Trivial PQM:

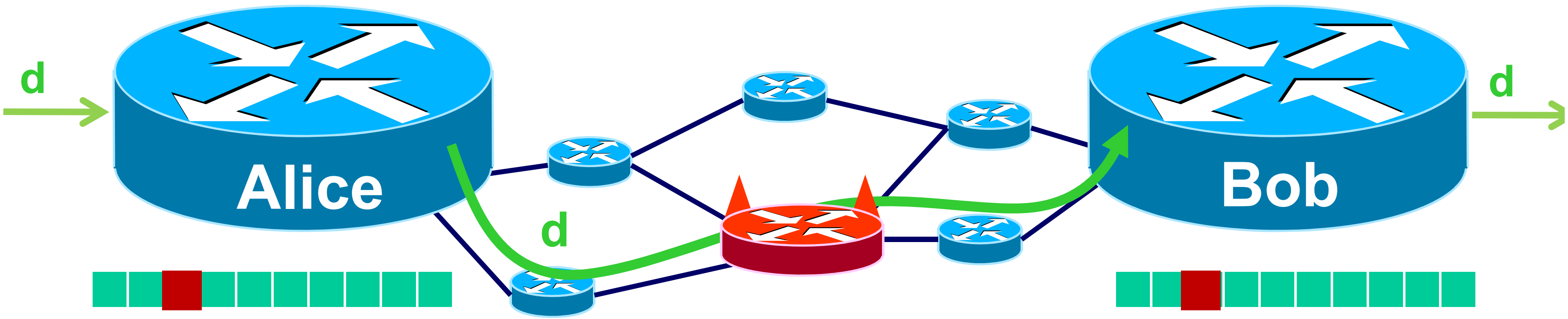
Bob acks *each* packet.

Alice detects loss if a packet is not ack'd

Alice stores each packet.
100% communication overhead.
Not practical for network layer!

Unforgability: Eve can't forge ack to a dropped packet

Secure Sketch PQM: Overview



Applies techniques from L2-norm estimation: [AMS96] [Ach01] **[CCF2004]** [TZ2004]

Sketch PQM:

Alice and Bob keep short sketch

Each maps info for **T** data packets to sketch

Bob sends Alice his sketch in a ‘report’

Alice compares sketches, decides if loss rate **> β**

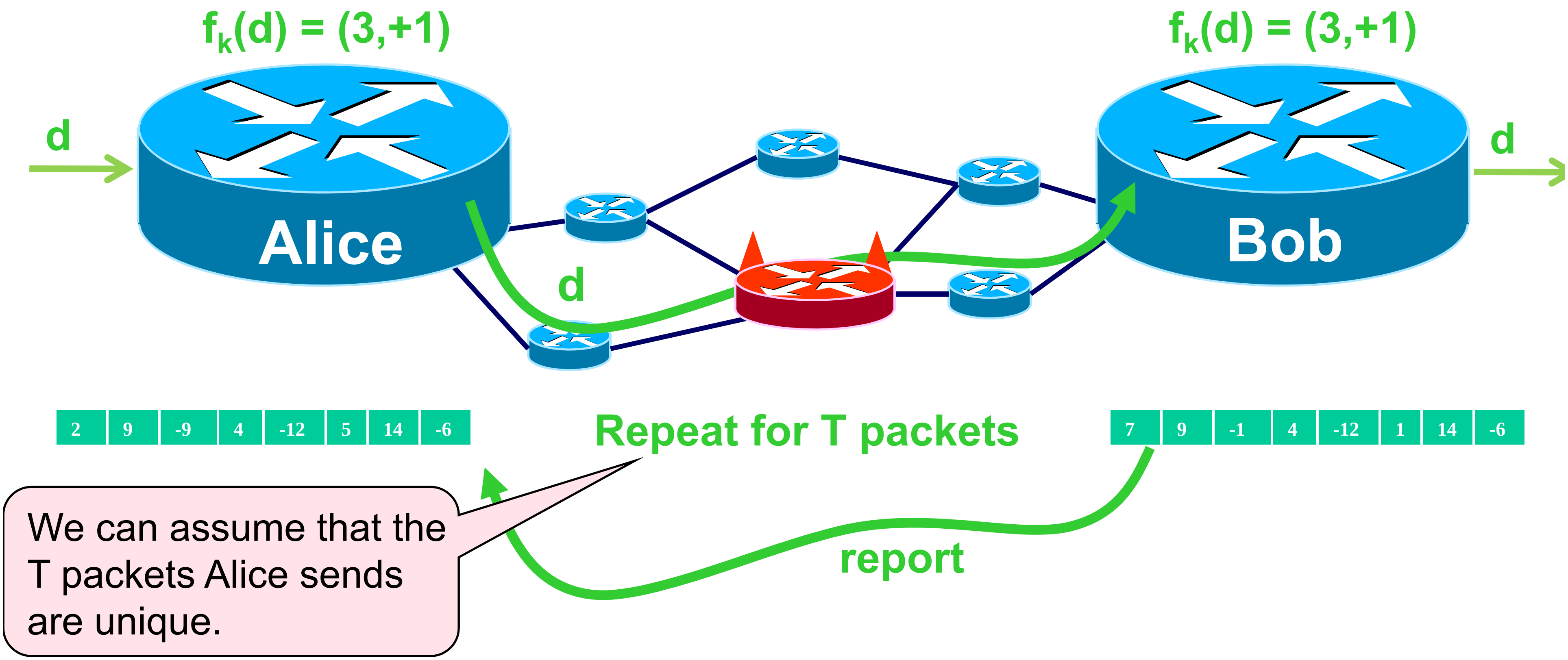
Unforgability: Eve can’t forge report

Unpredictable Mapping: Eve can’t mask packet drops with packet adds

Coordinated Mapping: Alice and Bob identically map packets to sketch

Secure Sketch PQM: Security (1)

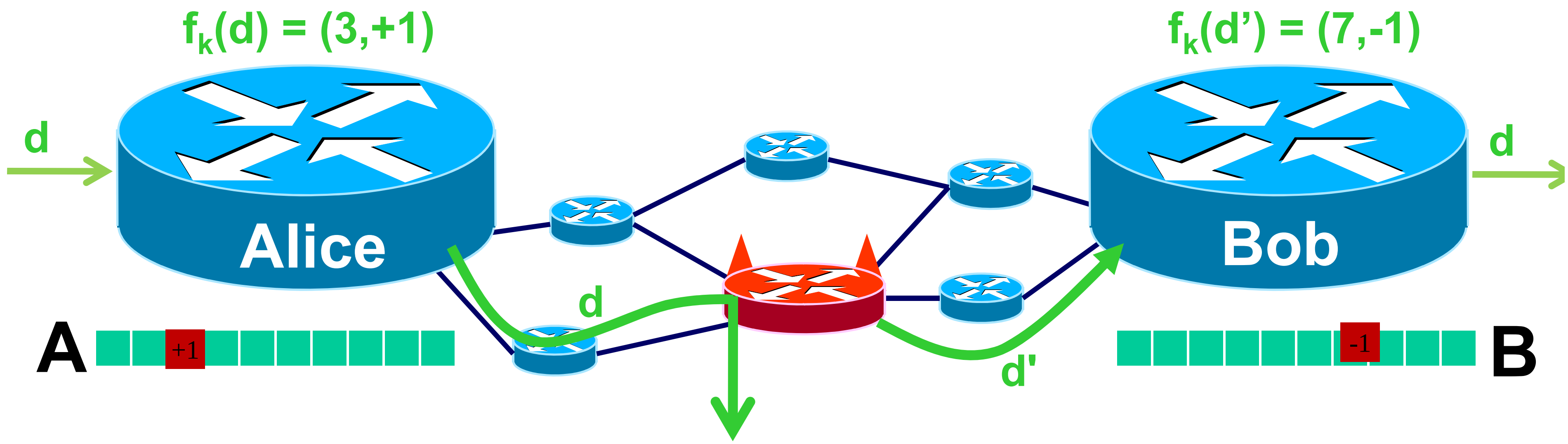
Use keyed cryptographic hash function f_k : packet → [N] x {+1,-1}



Unforgability:	Reports are cryptographically authenticated
Unpredictable Mapping:	Eve cannot predict output of hash without the key
Coordinated Mapping:	Alice and Bob have the same hash function + key

Secure Sketch PQM: Security (2)

What happens if a packet is dropped and replaced with a new packet?



Cryptographic hash ensures that w.h.p **d'** maps to new position in sketch, *regardless of what Eve does*

➔ Use the difference sketch **X = A-B** to detect packet drops + packet adds

Unforgability: Reports are cryptographically authenticated

Unpredictable Mapping: Eve cannot predict output of hash without the key

Coordinated Mapping: Alice and Bob have the same hash function + key

Secure Sketch PQM: Decision Rule

To decide between packet loss rate $< \alpha$ and $> \beta$:

- Take the difference sketch $X = A - B$
- Compute the estimator ΣX_i^2
- Raise an alarm iff $\Sigma X_i^2 > 2\alpha\beta / (\alpha + \beta)$

Equality when adds and drops are unique pkts

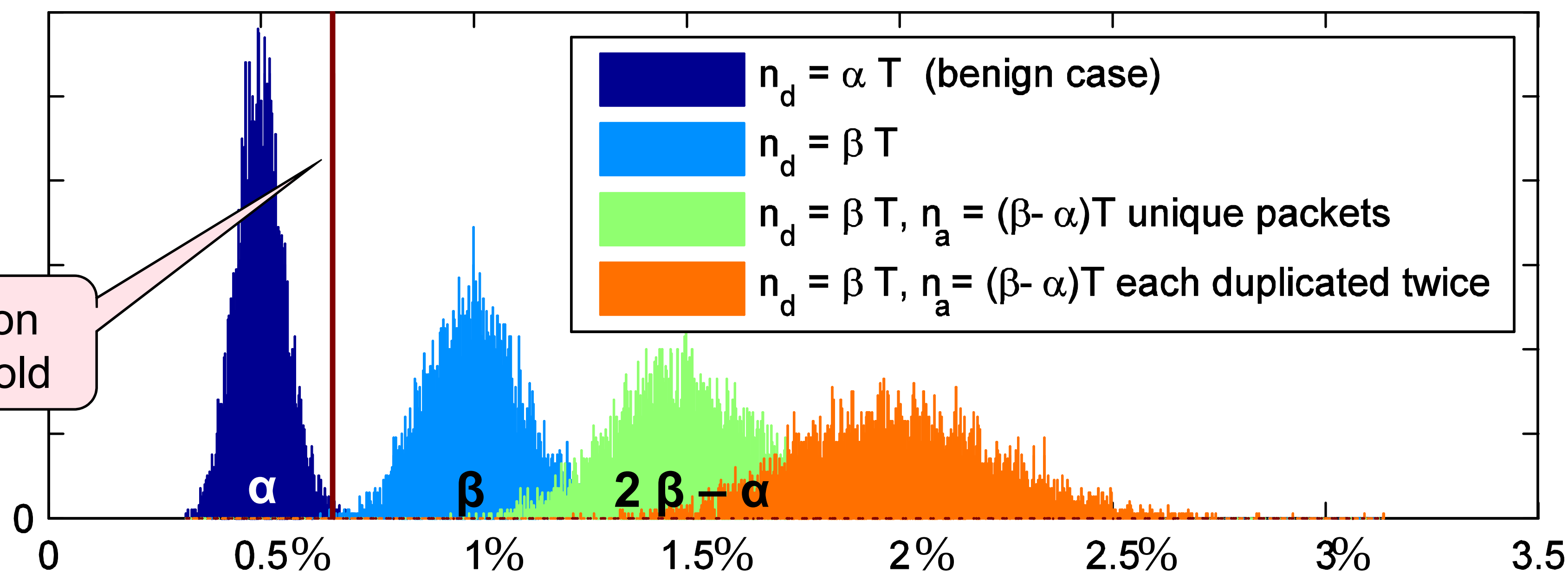
We can show that $E[\Sigma X_i^2] \geq n_d + n_a$

dropped pkts

added pkts

Histogram of Estimator. β=1%, α=0.5%, $T=10^6$ packets, N=300 bins in sketch .

Decision threshold



Normalized Estimator $\Sigma X_i^2/T$

Secure Sketch PQM: Sample Parameters (1)

Our result uses the facts that
1. Alice sends unique packets during interval.
2. The mapping function uses hash that is indistinguishable from a random function.

Thm (Simplified): Alice can use a secure sketch PQM protocol to decide between cases where packet loss rate is **< α** and **> β**, with **99%** success probability if the sketch has

$$N > 25.5\,(1/25 - (\beta-\alpha)/(\beta+\alpha))^{-2} \quad \text{bins}$$

and

$$T > 867\,N\,(\ln N + 9.21)/\alpha \quad \text{packets}$$

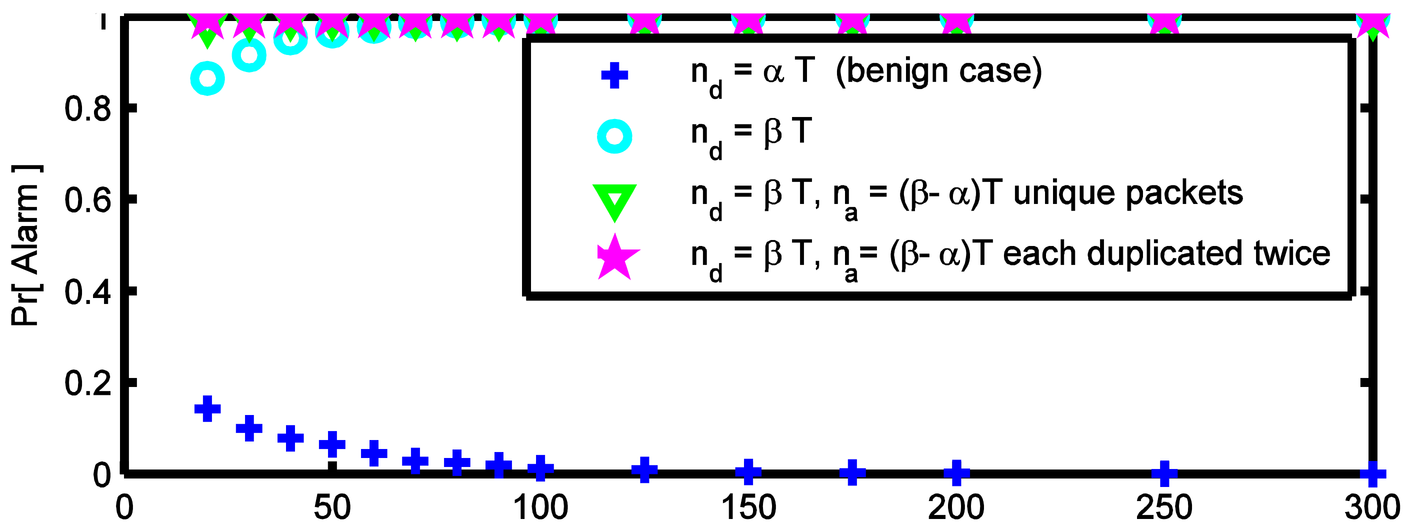
are monitored per interval.

If $\alpha = 0.5\%$, $\beta = 1\%$ then for $T = 10^9$ we need $N = 300$ bins.

Secure Sketch PQM: Sample Parameters (2)

From the Thm, if **α = 0.5%**, **β=1%** then for **$T=10^9$** we need **N=300** bins.

Numerical experiments suggest that for **$T=10^6$** or less, **N=150** bins is enough.



N, number of bins. $T=10^6$, β =1%, α=0.5%

If **N=150** ➡

T	Sketch Size
10^6	170 bytes
10^7	200 bytes
10^8	235 bytes
10^9	270 bytes

Secure Sketch PQM Summary

- Low storage overhead
- Low communication overhead
  - **1** report packet / **T** regular packets
  - Report contains sketch and authenticator

T	Sketch Size
10^6	170 bytes
10^7	200 bytes
10^8	235 bytes
10^9	270 bytes

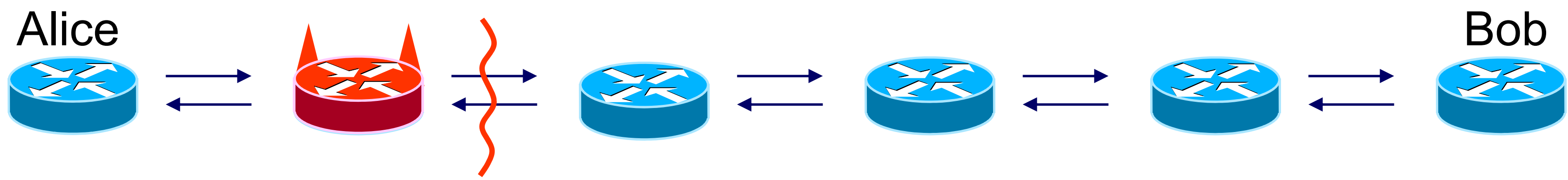
- No packet marking
  - Protocol is backward compatible.
  - Can be implemented in a monitor off the router's critical path
- One cryptographic hash computation per packet
  - *Online setting* means we can use fast hash functions
  - High-throughput
  - Latency doesn't matter, Parallelizable
- Shared keys at Alice and Bob

Thm [GXTBR08]: **Any secure PQM protocol** robust to adversarial nodes on the path that can **add/drop** packets, needs a key infrastructure and crypto.

This talk

1. Overview √
2. Secure Sketch PQM √
3. $\boxed{\text{Composing PQM to localize faulty links}}$
4. Conclusion

Fault Localization (FL)



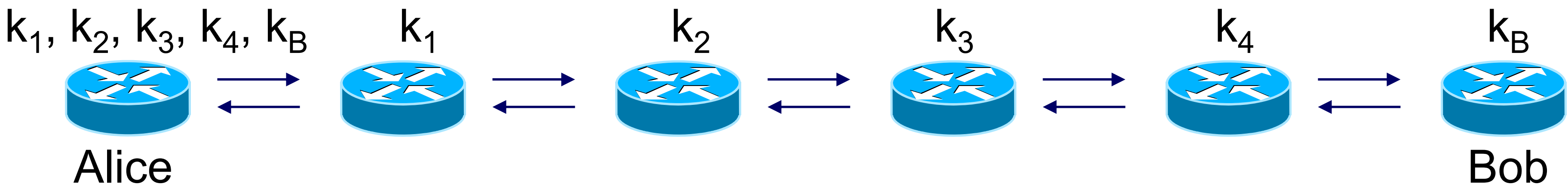
We assume:

1. Alice knows identity of nodes on path.
2. Paths are symmetric.
3. Eve occupies node(s) on the path, and can add, drop, modify packets.
4. Alice doesn't know where Eve is.

Secure fault localization (FL):
If the packet loss rate on a link exceeds **β**, Alice **outputs that link** (or a link adjacent to Eve) *regardless of Eve's behavior*
Alice **will not alarm** if packet loss rate on the path is less than **α**

Thm [BGX08]: **Any secure FL protocol** robust to adversarial nodes on the path that can **add** and **drop** packets, requires keys and crypto **at each node**.

Composition of Sketch PQM to FL (1)

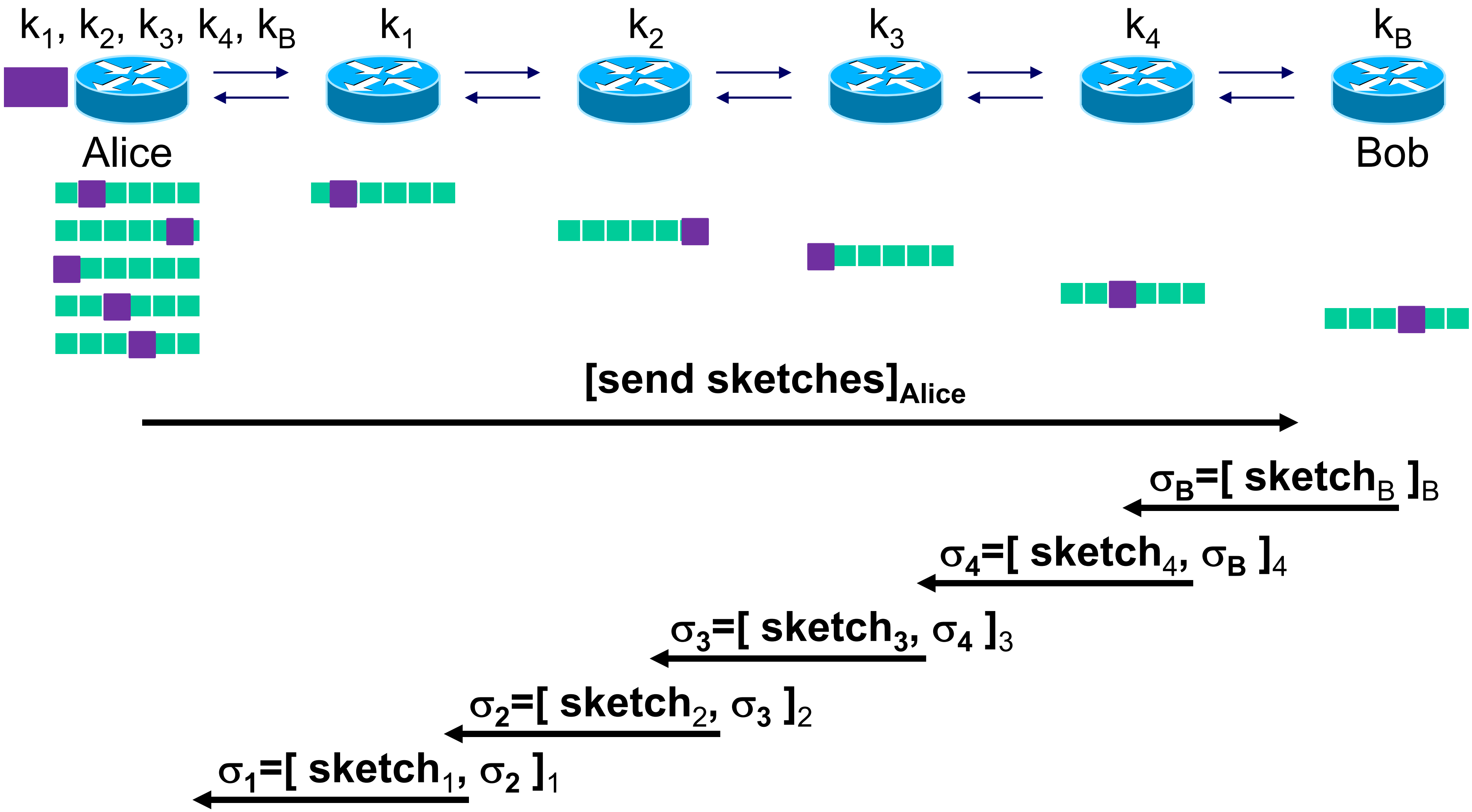


Secure fault localization (FL):
If the packet loss rate on a link exceeds **β**, Alice **outputs that link** (or a link adjacent to Eve) *regardless of Eve's behavior*
Alice **will not alarm** if packet loss rate on the path is less than **α**

Composition Overview

1. Alice shares a key with each node on the path.
2. Alice runs secure sketch PQM with each node on the path.
3. After **T** packets have been sent, Alice requests a report.
4. Each node responds with a report containing its sketch, and authenticated with an 'onion' message authentication code.

Composition of Sketch PQM to FL (2)



‘Onionizing’ the reports prevents Eve selectively dropping reports for an innocent node.

Summary of Contributions

[G., Xiao., Tromer, Barak, Rexford, "Path-Quality Monitoring in the Presence of Adversaries", SIGMETRICS 2008.]
[Barak, G., Xiao., "Protocols and Lower Bounds for Fault Localization in the Internet", EUROCRYPT 2008.]

1) "Positive" security definitions for PQM and FL, not attack taxonomies
2) Proof that Secure PQM needs keys and crypto
3) Efficient PQM is possible for very strong threat model
   a) Secure sketch protocol
      - New bound on parameters. Uses uniqueness of traffic.
   b) Secure sampling protocol for symmetric + client-server settings
      - Measure delay as well as loss
4) Proof that secure FL requires keys and crypto *at each node*
   - Non-trivial proof using black-box separations and learning theory
   - Secure FL requires participation from all nodes on path
5) Per-packet FL protocol
6) Composition of PQM protocols to statistical FL protocols.

Conclusions

What security primitives can we have in future networks ?

What is the right balance between **strength of threat model** and **efficiency of scheme** ?

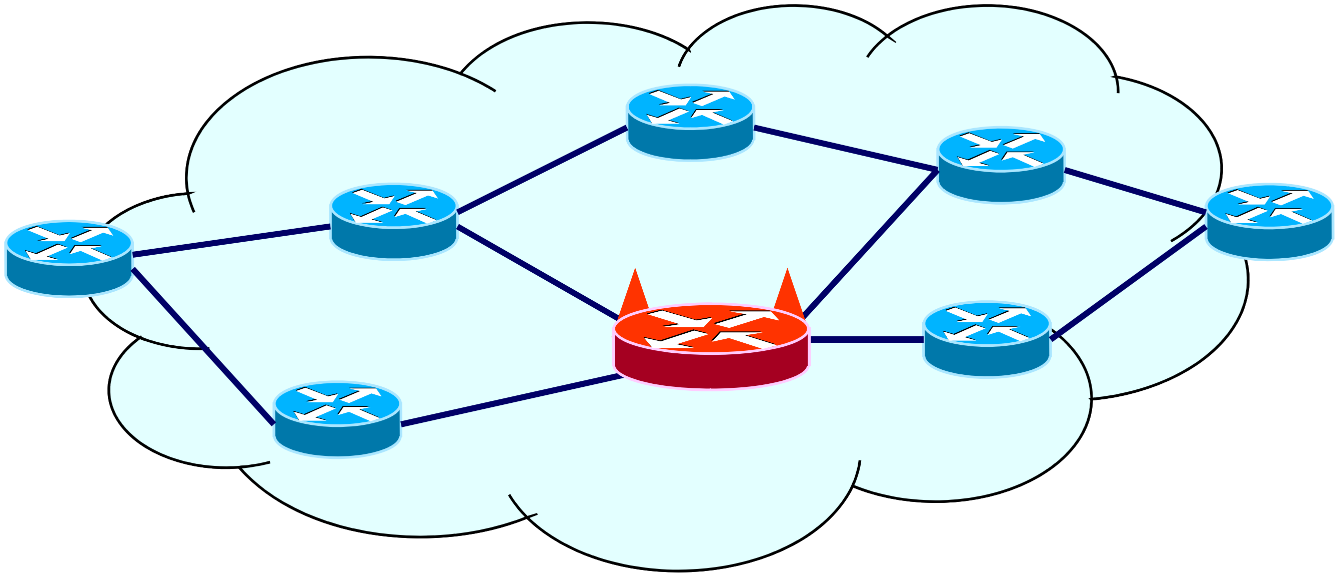
We showed here that:

1. Efficient PQM is possible for very strong threat model
2. Secure FL requires keys and crypto at *each node*,
   - ❑ Hard to get full participation in Internet
   - ❑ May be better for highly-secure networks / special traffic
3. What about other areas? BGP security, secure availability, … …

And that…

- ❑ We can use theoretical cryptography to inform what we do in practice!

Thanks!



[G., Xiao., Tromer, Barak, Rexford, "Path-Quality Monitoring in the Presence of Adversaries", in submission.]

[Barak, G., Xiao., "Protocols and Lower Bounds for Fault Localization in the Internet", in submission.]

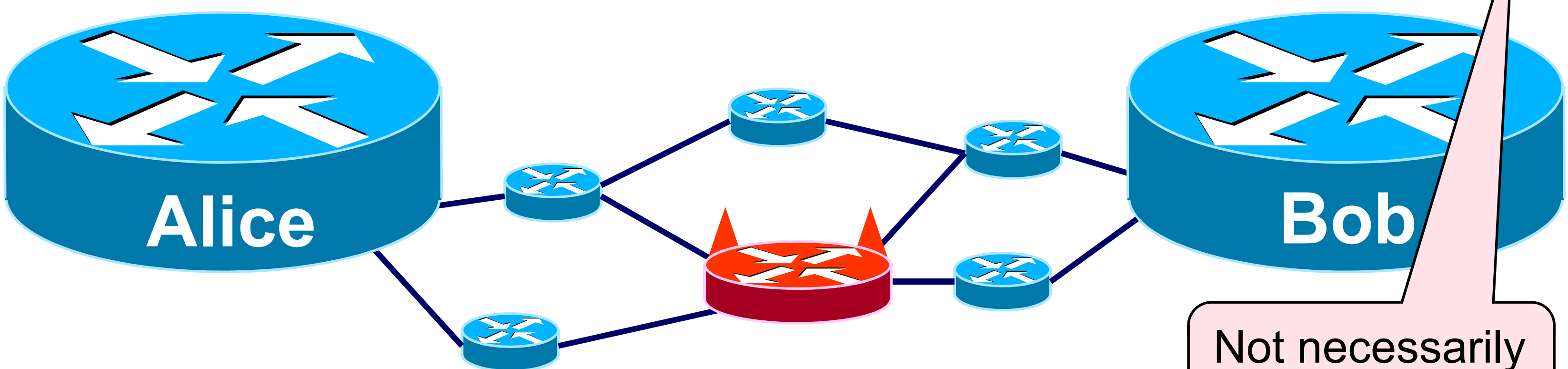
www.princeton.edu/~goldbe

Princeton **University**

Secure PQM needs keys

Our protocol requires a key infrastructure between Alice and Bob.

Thm: **Any secure PQM protocol** that is robust adversaries on the path that can **add** and **drop** packets requires a key infrastructure.



Not necessarily pairwise keys!

<u>Proof:</u> (In the contrapositive)

Assume Alice and Bob **do not** have a shared key

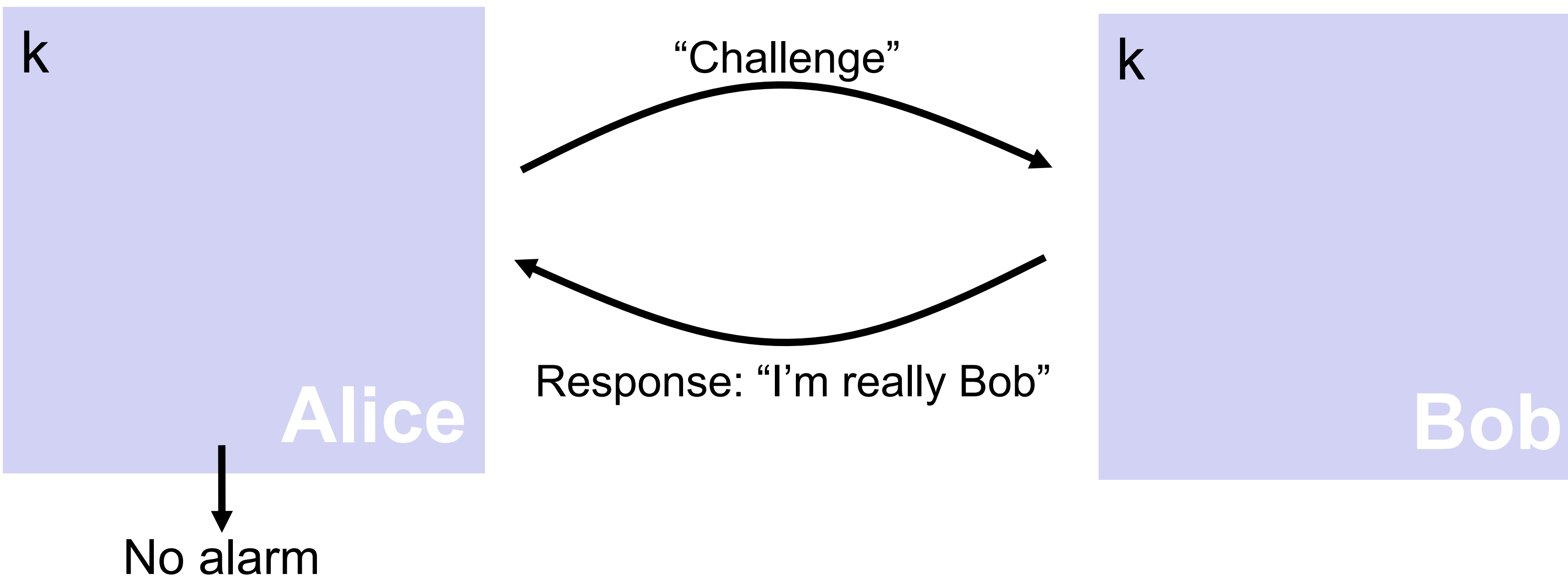
- All the packets that Alice sends to Bob pass thru Eve
- Then Eve knows everything Bob knows
- Eve drops all packets
- Eve impersonates Bob's reverse path messages (e.g. report)
- Alice won't detect packet loss, so Eve breaks security.

Secure PQM needs crypto (1)

Our protocol requires a key infrastructure between Alice and Bob.

Thm: **Any secure PQM protocol** that is robust adversaries on the path that can **add/drop** packets must invoke cryptographic operations.

Proof: (By **reduction** to keyed identification schemes (KIS))

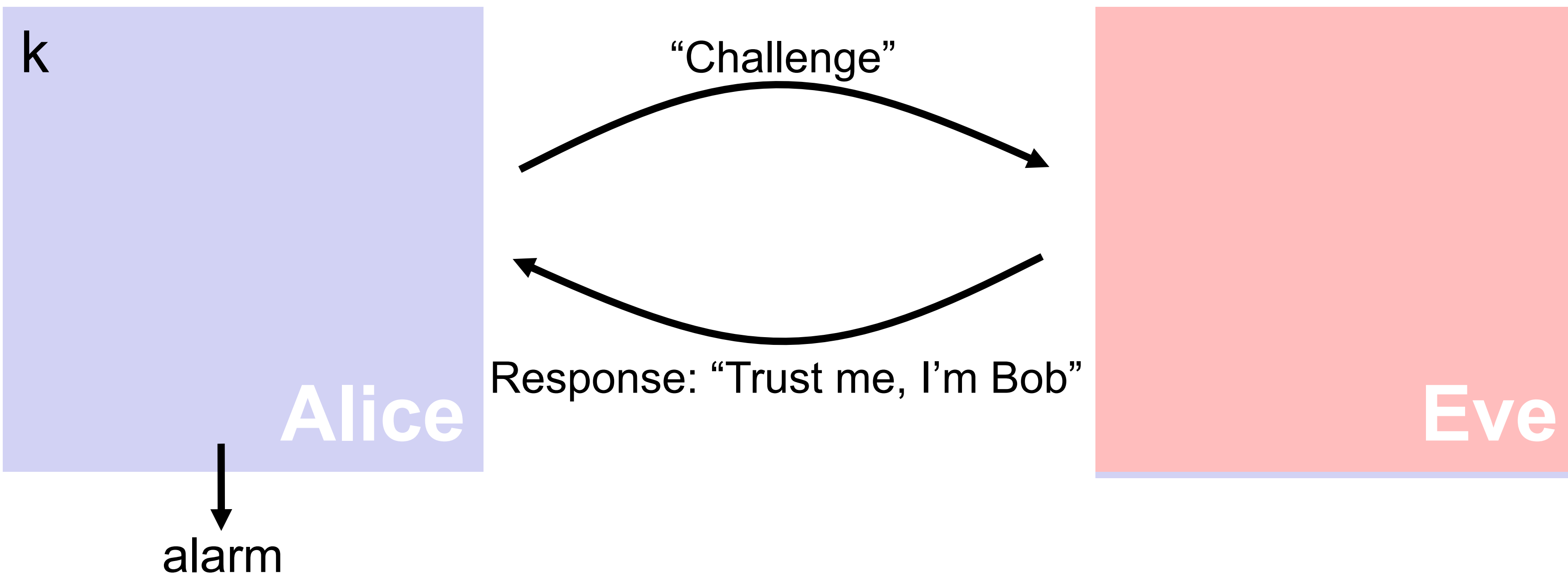


Secure PQM needs crypto (2)

Our protocol requires a key infrastructure between Alice and Bob.

Thm: **Any secure PQM protocol** that is robust adversaries on the path that can **add/drop** packets must invoke cryptographic operations.

Proof: (By **reduction** to keyed identification schemes (KIS))

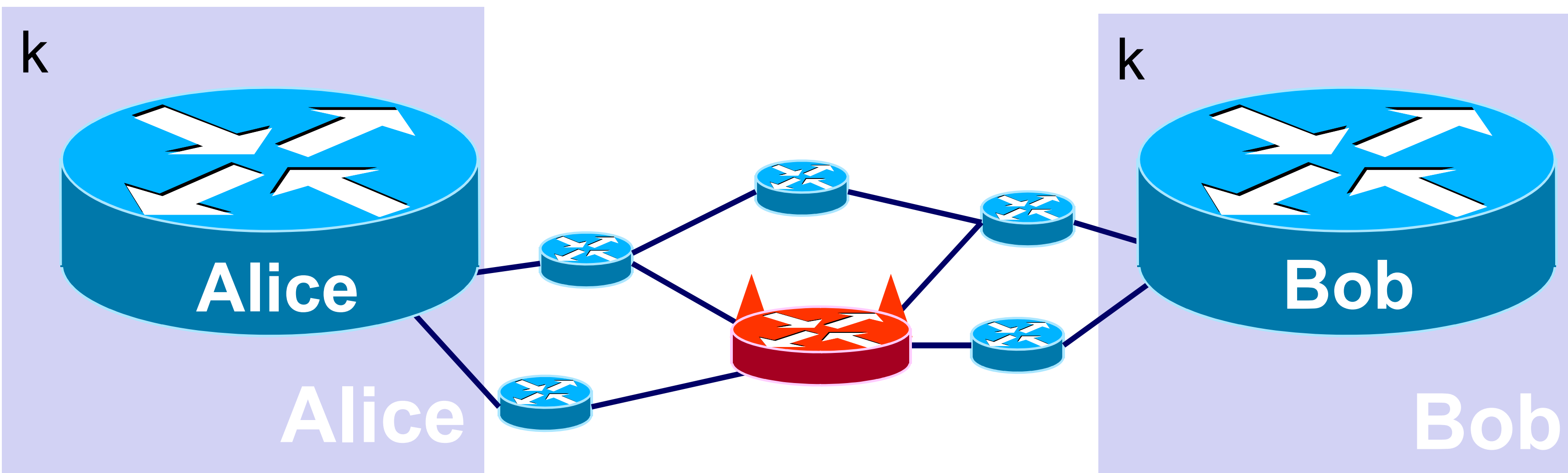


Secure PQM needs crypto (3)

Our protocol requires a key infrastructure between Alice and Bob.

Thm: **Any secure PQM protocol** that is robust adversaries on the path that can **add/drop** packets must invoke cryptographic operations.

Proof: (By **reduction** to keyed identification schemes (KIS))



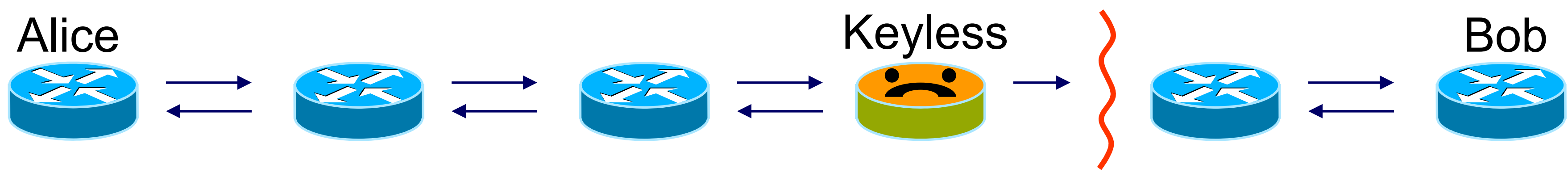
KIS are at least as computationally complex as symmetric cryptographic primitives (e.g. encryption, MAC)

➔ Secure PQM needs crypto

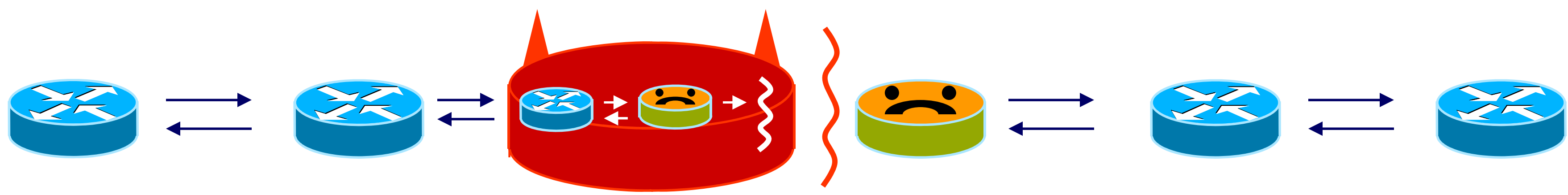
Fault Localization needs keys at each node

Theorem: Each node needs a shared secret with Alice

Proof: Suppose node *i* does not a share secret with any upstream node:



Case (a): Node *i+1* is unreachable



Case (b): Eve drops packets and impersonates keyless node *i*

Case (a) and case (b) are indistinguishable to Alice

⇒ In case (b) Eve drops packets while making innocent link (i, i+1) look guilty.

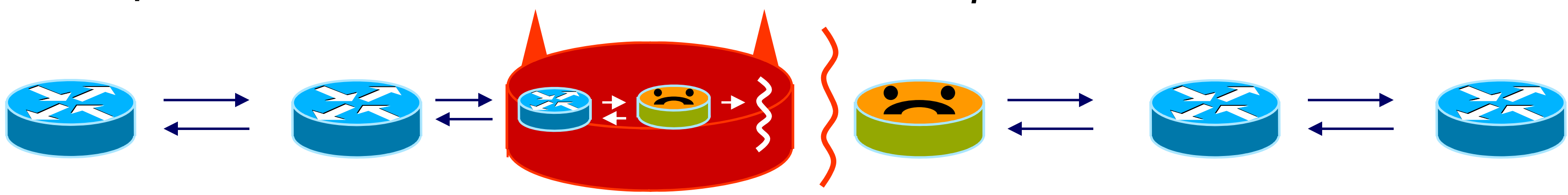
⇒ The FL protocol cannot be secure.

Fault localization needs crypto at each node

Theorem: Each node i needs to perform cryptographic operations

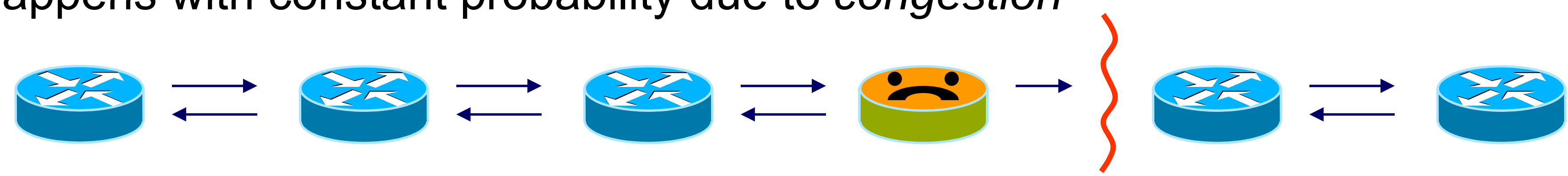
Proof: Suppose node *i* has a shares key k_i with Alice but does not do crypto.

Eve impersonates node *i* . he needs to learn k_i !



Case (b): Eve drops packets and impersonates crypto-less node *i*

Since *i* doesn't do crypto Eve can learn k_i by observing case (a), which happens with constant probability due to *congestion*



Case (a): Node *i+1* is unreachable due to congestion

⇒ Then Eve can impersonate node *i* in case (b) and FL protocol is not secure!