Internet Path-Quality Monitoring in the Presence of Adversaries

Sharon Goldberg
David Xiao, Eran Tromer, Boaz Barak, Jennifer Rexford

Excerpts of talks that have been presented in seminars at Penn State University, IBM Research, Cisco, Ben Gurion University, and the Weizmann Institute.
Packets routed from Alice to Bob via a path of intermediate routers

Routing protocols used to set up paths between routers

Today’s focus

Packets forwarded along these paths with best-effort delivery

- No guarantees on packet arrival or integrity
- Congestion (random packet dropping) and reordering
Applications of path-quality monitoring

‘node’ = router or ISP

Alice
Company Site A

Bob
Company Site B

Routers need tools to detect unacceptably high packet loss rates.

Performance Routing
- Balancing loads between multiple paths (e.g. multihomed company sites)
- Quick response for avoiding blackholed routes and brownouts
- Avoiding “suspicious” paths (e.g. that drop Skype pkts, or corrupt traffic)

SLA compliance monitoring
- e.g. Cisco IP SLA’s – detects end-to-end performance degradation
The presence of adversaries

Does packet loss rate exceed 1%?

Covers active attack:
- Corrupted router
- Botnet
- Greedy ISP
And all benign failures.

Knows monitoring protocol
Wants to hide packet loss from Alice

Can we have both?

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

Efficient protocols for high-speed routers
- Extremely limited storage, communication, computation
- No marking or encryption of existing traffic
This talk

1. Overview
2. Defining secure PQM
3. Secure Sketch PQM
4. PQM and the adversarial sketch model
5. Public-Key / Client-Server PQM protocols
6. Conclusion
Formal Definition of PQM (1)

Malicious case

Secure path quality monitoring (PQM)
With probability $1 - \delta = 99\%$,  
- Alice **alarms** if packet loss rate exceeds $\beta$ **regardless of Eve's actions**
Formal Definition of PQM (2)

Benign case

Secure path quality monitoring (PQM)
With probability $1 - \delta = 99\%$,
- Alice **alarms** if packet loss rate exceeds $\beta$ *regardless of Eve’s actions*
- Alice **will not alarm** if packet loss rate is less than $\alpha$ *in benign case*

Main result: For every $\alpha < \beta < 1$ and security parameter $k$ there exists a PQM protocol with $O(k + \log(T))$ communication and storage, one hash computation / packet and no packet marking.
Overview of (some of) our results

Secure path quality monitoring (PQM)
With probability $1 - \delta = 99\%$,
• Alice **alarms** if packet loss rate exceeds $\beta$ **regardless of Eve’s actions**
• Alice **will not alarm** if packet loss rate is less than $\alpha$ **in benign case**

Main result: For every $\alpha < \beta < 1$ and security parameter $k$, there exists a PQM protocol with $O(k+\log(T))$ communication and storage, one hash computation / packet and no packet marking.

Analysis
\[
\begin{align*}
\alpha &= 0.5\% \\
\beta &= 1\%
\end{align*}
\]
\[\downarrow\]
\[
\text{storage} = 540 \text{ bytes} \\
T = 10^9 \text{ packets}
\]

Simulations
\[
\begin{align*}
\alpha &= 0.5\% \\
\beta &= 1\%
\end{align*}
\]
\[\downarrow\]
\[
\text{storage} = 170 \text{ bytes} \\
T = 10^6 \text{ packets}
\]
This talk

1. Overview √
2. Defining secure PQM √
3. Secure Sketch PQM
4. PQM and the adversarial sketch model
5. Public-Key / Client-Server PQM protocols
6. Conclusion
Background: Secure 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!

Monitors all traffic from interface

We want to avoid encrypting all traffic.

Other related work:
[IPsec] No acks. High overhead.
[AR06] Mark and monitor only a fraction of traffic. Encrypt to hide mark.
[MSMC05] Fatih and [SRSSK04] Listen, both insecure in our model.
Secure Sketch PQM: The Protocol

Uses $\ell_2$-norm estimation sketches: [AMS96] [Ach01] [CCF2004] [TZ2004]

- Alice
  - Key $k$
  - $d_1, \ldots, d_T$
- Bob
  - Key $k$
  - $d'_1, \ldots, d'_T$

Hash each packet $f_k(d) = \text{index, bit}$
Update sketch $A[\text{index}] += \text{bit}$

Hash each packet $f_k(d) = \text{index, bit}$
Update sketch $B[\text{index}] += \text{bit}$
Secure Sketch PQM: The Protocol

Uses $\ell_2$-norm estimation sketches: [AMS96] [Ach01] [CCF2004] [TZ2004]

Key $k$

Hash each packet $f_k(d) = \text{index, bit}$

Update sketch $A[\text{index}] += \text{bit}$

[report: $B$]$_{\text{Bob}}$
Secure Sketch PQM: The Protocol

**Uses \( l_2 \)-norm estimation sketches:** [AMS96] [Ach01] [CCF2004] [TZ2004]

Bob

Alice

d\(_1\), ..., d\(_T\)

d\(_1\), ..., d\(_T\)

Hash each packet \( f_k(d) = \text{index}, \text{bit} \)

Update sketch \( A[\text{index}] += \text{bit} \)

Hash each packet \( f_k(d) = \text{index}, \text{bit} \)

Update sketch \( B[\text{index}] += \text{bit} \)

Send authenticated (MAC’d) sketch

To decide between loss rate < \( \alpha \) and > \( \beta \):

- Take difference sketch \( X = A - B \)
- Compute its \( l_2 \)-norm \( \Sigma X_i^2 \)
- Raise an alarm iff \( \Sigma X_i^2 / T > (\alpha + \beta) / 2 \)

Refresh hash key & Repeat

Refresh hash key & Repeat

Actually, we used a different threshold to optimize constants
Secure Sketch PQM: Analysis

Thm (Simplified): Alice can use (CCF-based) secure sketch PQM protocol to decide between cases where packet loss rate is $< \alpha$ and $\beta = 2\alpha$, with $1-\delta$ success probability if the sketch has

$$N > 65 \ln (100 / 99\delta)$$

bins

and

$$T > 867 N (\ln (100 N / \delta)) / \alpha$$

packets monitored per interval.

Analysis

$\alpha = 0.5\% \quad \delta = 1\%

\Downarrow

N = 300 \quad T = 10^9$

Simulations

$\alpha = 0.5\% \quad \delta = 1\%

\Downarrow

N = 150 \quad T \geq 10^6$

I’ll show the proof for PQM using classic $\ell_2$-norm sketches [AMS96] [Ach01]
Secure Sketch PQM: The “Classic” Version

Uses $\ell_2$-norm estimation sketches: [AMS96] [Ach01] [CCF2004] [TZ2004]

**Alice**

- Key $k$
- Hash each packet $f_k(d) = [b_1 \ldots b_N]$
- Update sketch $B += [b_1 \ldots b_N]$

**Bob**

- Key $k$
- PRF: $f_k()$: packets $\rightarrow \{1,-1\}^N$
- Hash each packet $f_k(d) = [b_1 \ldots b_N]$
- Update sketch $B += [b_1 \ldots b_N]$

Send authenticated (MAC’d) sketch

To decide between loss rate $< \alpha$ and $> \beta$:
- Take difference sketch $X = A - B$
- Compute its $\ell_2$-norm $\Sigma X_i^2 / N$
- Raise an alarm iff $\Sigma X_i^2 / NT > 2 \alpha \beta / (\alpha + \beta)$
Analysis with “classic” sketching (1)

PRF → random function
Packet Stream → long vector
Hashing packet → multiplication by random $R$ in $\{-1, 1\}^{Nx2}$ packet size

1 if packet sent, 0 otherwise

What Alice sends
length = $2^{\text{packet size}}$

What Bob receives
length = $2^{\text{packet size}}$

$A = Rv_A$

$B = Rv_B$

$X = Rv_A - Rv_B$
Analysis with “classic” sketching (2)

PRF → random function
Packet Stream → long vector
Hashing packet → multiplication by random $R$ in $\{-1,1\}^{N \times 2}$ packet size

What Alice sends
length = $2^{\text{packet size}}$

What Bob receives
length = $2^{\text{packet size}}$

$v_A - v_B$

$\ell_1$ norm = #drops + #adds

$X = R(v_A - v_B)$

If we used a good $\ell_1$ norm estimation sketch, we’d be done. But we use (more efficient) $\ell_2$ norm estimation.
Analysis with “classic” sketching (3)

- **PRF** → random function
- **Packet Stream** → long vector
- **Hashing packet** → multiplication by random $R$ in $\{-1,1\}^{Nx2}$ packet size

What Alice sends
- length = $2^{\text{packet size}}$
- $v_A$ = 0101100101...1

What Bob receives
- length = $2^{\text{packet size}}$
- $v_B$ = 0001100000...1

$p$ = 1 if packet sent, 0 otherwise

$\mathbf{v}_A - \mathbf{v}_B$ = 0100000101...0

difference
- $\ell_1$ norm = #drops + #adds

Benign case:
- no adds $\Rightarrow v_B$ subset $v_A$
- $v_A - v_B$ is $\{0, 1\}$ vector
- $\ell_1 = \ell_2^2$ for $\{0, 1\}$ vectors
- $\ell_2^2$ sketch estimates #drops

Malicious case:
- can have duplicate adds
- $v_A - v_B$ is $\{0, 1, -1, -2, ...\}$ vector
- $\ell_1 \leq \ell_2^2$ ( = unique adds)
- $\ell_2^2$ sketch (overestimates) #drops
- duplicates increase $\text{Pr}[\text{alarm}]$
Analysis with “classic” sketching (4)

$\mathbf{v}_A - \mathbf{v}_B$

Multiply by random $R$ in $\{-1,1\}^{N\times 2}$ packet size

$X = R(\mathbf{v}_A - \mathbf{v}_B)$

“JL-Theorem” [Ach01]: For any (long) vector $\mathbf{v}$ and random $\{-1,1\}$-matrix mapping $\mathbf{v}$ to $N$ dimensions, then w.p. $\exp(-O(N\varepsilon^2))$

$$(1-\varepsilon) \|\mathbf{v}\|^2 < \|R\mathbf{v}\|^2 / N < (1+\varepsilon) \|\mathbf{v}\|^2$$

Corollary: For error $\delta$ take a sketch of size $N = O(\log(1/\delta) 1/\varepsilon^2)$

PQM Decision Rule: To decide between drop rate $< \alpha$ and $> \beta = 2\alpha$ with confidence $1 - \delta$ alarm iff

$$\|R(\mathbf{v}_A - \mathbf{v}_B)\|^2 / N > 2\alpha\beta / (\alpha + \beta) T$$

and use sketch length $N = O(\log(1/\delta) (\beta + \alpha)^2 / (\beta - \alpha)^2)$
Secure Sketch PQM: Analysis with CCF

**Thm (Simplified):** Alice can use (CCF-based) secure sketch PQM protocol to decide between cases where packet loss rate is $< \alpha$ and $> \beta = 2\alpha$, with $1-\delta$ success probability if the sketch has

$$N > 65 \ln \left(\frac{100}{99\delta}\right)$$

bins and

$$T > 867 N \left(\ln \left(\frac{100 N}{\delta}\right)\right) / \alpha$$

packets monitored per interval.

- **Analysis**
  - $\alpha = 0.5\%$  $\delta = 1\%$
  - $N = 300$  $T = 10^9$

- **Simulations**
  - $\alpha = 0.5\%$  $\delta = 1\%$
  - $N = 150$  $T \geq 10^6$
Simulations with CCF (1)

Analysis

\[ \alpha = 0.5\% \quad \delta = 1\% \]

\[ N = 300 \quad T = 10^9 \]

Simulations

\[ \alpha = 0.5\% \quad \delta = 1\% \]

\[ N = 150 \quad T \geq 10^6 \]

Recall that \( \| R(v_A - v_B) \|_2 = \| A - B \|_2 \geq \) drops + adds

Histogram of Estimator \( \| A - B \|_2 \)

\( \alpha = 0.5\% \), \( \beta = 2 \alpha \), \( N = 300 \) bins in sketch, \( T = 10^6 \) packets,

- 0.5% drops (benign case)
- 1% drops
- 1% drops, 0.5% unique adds
- 1% drops; 0.5% duplicate adds

Normalized Estimator \( \Sigma X_i^2 / T \)
Simulations with CCF (2)

Analysis

\[ \alpha = 0.5\% \quad \delta = 1\% \]
\[ \downarrow \]
\[ N = 300 \quad T = 10^9 \]

Simulations

\[ \alpha = 0.5\% \quad \delta = 1\% \]
\[ \downarrow \]
\[ N = 150 \quad T \geq 10^6 \]

Sketch Size

<table>
<thead>
<tr>
<th>( T )</th>
<th>Sketch Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 10^6 )</td>
<td>170 bytes</td>
</tr>
<tr>
<td>( 10^7 )</td>
<td>200 bytes</td>
</tr>
<tr>
<td>( 10^8 )</td>
<td>235 bytes</td>
</tr>
<tr>
<td>( 10^9 )</td>
<td>270 bytes</td>
</tr>
</tbody>
</table>

If \( N = 150 \)
## Secure Sketch PQM Summary

- **Low storage overhead**
- **Low communication overhead**
  - 1 report packet / $T$ regular packets
  - Report contains sketch and authenticator
- **No packet marking**
  - Protocol is backward compatible.
  - Can be implemented off the fast path of the router
- **One cryptographic hash computation per packet**
  - *Online setting* so we can use fast hash functions
    - Even universal hash functions work!
  - High-throughput
  - Do not modify packets, so can compute hash after packet sent
- **Shared keys at Alice and Bob**
  - Can be derived from public key infrastructure via key exchange

<table>
<thead>
<tr>
<th>$T$</th>
<th>Sketch Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^6$</td>
<td>170 bytes</td>
</tr>
<tr>
<td>$10^7$</td>
<td>200 bytes</td>
</tr>
<tr>
<td>$10^8$</td>
<td>235 bytes</td>
</tr>
<tr>
<td>$10^9$</td>
<td>270 bytes</td>
</tr>
</tbody>
</table>
**Secure Sketch PQM Summary**

- **Low storage overhead**
  - Sketch Size
  - \(10^6\) 170 bytes
  - \(10^7\) 200 bytes
  - \(10^8\) 235 bytes
  - \(10^9\) 270 bytes

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

- **No packet marking**
  - Protocol is backward compatible.
  - Can be implemented off the fast path of the router.

- **One cryptographic hash computation per packet**
  - **Online setting** so we can use fast hash functions.
    - Even universal hash functions work!
  - High-throughput
  - Do not modify packets, so can compute hash after packet sent

- **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. Public-Key / Client-Server PQM protocol
4. PQM and the adversarial sketch model
5. Conclusion
A Public-Key / Client-Server PQM Protocol (1)

**Want:**
- Server uses the same key for each client

**Challenge:**
- Client can be malicious
- Public key operations are expensive

Can’t share same symmetric key with many senders

**Solution:**
Bob uses a temporary key (salt) that is revealed after use
Run a secure sampling protocol using the salt
Receiver can respond to many senders with same salt and PK

In each interval, choose salt at random

Alice 1

Alice 2

Bob

Alice 3
A Public-Key / Client-Server PQM Protocol (2b)

Receiver can respond to many senders with same salt and PK

Bob

Alice 1

[Bob, salt] \_SK(Bob)

Alice 2

[Bob, salt] \_SK(Bob)

Bob

[Bob, salt] \_SK(Bob)

Alice 3

At the end of the interval, release the salt.

Similar to TESLA multi-cast signatures [PCST]
Client-Server Secure Sampling

Choose salt at random

If $f_{\text{salt}}(d) < p$
Send $[d]_{\text{salt}}$
Else send nothing

With probability $q$
Store $d$ ($d$ is a probe)

Use $PK(Bob)$ to verify salt

Use salt to check which probes needed and received valid ack

Alice stores: $O(T)$ Communication: $O(T)$
Can we get $O(\log T)$ with sketching?

No!
This is the Adversarial Sketch Model
This talk

1. Overview √
2. Secure Sketch PQM √
3. Public-Key / Client-Server PQM protocol √
4. PQM and the adversarial sketch model
5. Conclusion
The Adversarial Sketch Model [MNS08]

Alice’s Set

adversary chooses sets

Bob’s Set

sketch without shared randomness

exchange sketches via secure channel

Lower bound for norm of symmetric difference

| Alice Sketch | x | Bob Sketch | = O(|size of sets|)

Via reduction to equality testing in simultaneous communication model [BK97]
Symmetric-key PQM in Adversarial Sketch Model

Alice's adversary chooses sets of packets

Received Packets

with private sketch without shared randomness

Bob's adversary chooses

Sent packets

exchange sketches via secure channel

Use symmetric keys

Lower bound for norm of symmetric difference:

\[ \| \text{Alice Sketch} \| _x \| \text{Bob Sketch} \| = O(\text{size of sets}) \]

Via reduction to equality testing in simultaneous communication model [BK97]
Public-key PQM in Adversarial Sketch Model

adversary chooses
Received Packets

Sent packets

sketch without shared randomness

exchange sketches via secure channel

Use public keys

Lower bound for norm of symmetric difference

| Alice Sketch | x | Bob Sketch | = O(|size of sets|) = O(T)

Via reduction to equality testing in simultaneous communication model [BK97]
Conclusions

Sometimes we don’t have to give up security for the sake of efficiency

1. **Efficient** and **secure** path-quality monitoring is possible
   - Combining cryptography and sketching
   - Can monitor billions of packets using ~200 bytes of storage
   - No packet marking
   - Can use faster (and weaker) hash functions

2. PQM can be seen as an application of adversarial sketch model
   - And, sadly, sometimes subject to same lower bounds

www.princeton.edu/~goldbe
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.

**Proof:** (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))

![Diagram](Diagram.png)

No alarm

Response: “I’m really Bob”
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))

\[ k \]

Alice

“Challenge”

Response: “Trust me, I’m Bob”

Eve

alarm
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) )

**Challenge:** Traffic that Alice sends on the forward path

**Response:** Reverse path messages, i.e. report.

Alarm if report is invalid.
Secure PQM needs crypto (4)

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