Path-Quality Monitoring in the Presence of Adversaries: The Secure Sketch Protocols

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

ABSTRACT

Edge networks connected to the Internet need effective monitoring techniques to inform routing decisions and detect violations of Service Level Agreements (SLAs). However, existing measurement tools, like ping, traceroute, and trajectory sampling, are vulnerable to attacks that can make a path look better than it really is. Here we design and analyze a lightweight *path-quality monitoring protocol* that reliably raises an alarm when the packet-loss rate exceed a threshold, even when an adversary tries to bias monitoring results by selectively delaying, dropping, modifying, injecting, or preferentially treating packets. Our protocol is based on sublinear algorithms for sketching the second moment of stream of items, and can monitor billions of packets using only 250–600 bytes of storage and the periodic transmission of a comparably sized IP packet. We also show how this protocol can be used to construct a more sophisticated protocol that allows the sender to localize the link responsible for the dropped packets. We prove that our protocols satisfy a precise definition of security, analyze their performance using numerical experiments, and derive analytic expressions for the trade-off between statistical accuracy and system overhead. This paper contains a deeper treatment of results from earlier conference papers [11], [24], and several new results.

I. INTRODUCTION

Path-quality monitoring is a crucial component of flexible routing techniques (*e.g.,* intelligent route control, source routing, and overlay routing) that give edge networks greater control over path selection. Monitoring is also necessary to verify that service providers deliver the performance specified in Service-Level Agreements (SLAs). In both applications, edge networks need to determine when path quality degrades beyond some threshold, in order to switch from one path to another or report an SLA violation. The problem is complicated by the presence of nodes along the path who try to interfere with the measurement process, out of greed, malice, or just misconfiguration. In this paper, we design and analyze light-weight *path-quality monitoring* (PQM) protocols that detect when packet loss exceeds a threshold, even when adversaries try to bias monitoring results. We also study *failure localization* techniques that allow a sender to localize the specific links along a path where packets were dropped or modified; while protocols for this task are available today (*e.g.,* traceroute), they can easily be gamed in the adversarial setting we consider here. Our PQM protocol require surprisingly little storage and communication, and are intended to run at line rate at the network layer on high-speed routers at the edge and core of the Internet.

S. Goldberg is with the Department of Computer Science at Boston University, D. Xiao is with CNRS, LIAFA and Université Paris Diderot - Paris 7, E. Tromer is with the Computer Science Department at Tel Aviv University, B. Barak is with Microsoft Research, New England, and J. Rexford is with the Computer Science Department at Princeton University.

Manuscript submitted November 25, 2013. Revised June XXX, 2014.

A. The presence of adversaries

Today, path-quality monitoring relies on active measurement techniques, like ping and traceroute, that inject special probe packets into the network. In addition to imparting extra load on the network, active measurements are vulnerable to adversaries that bias the results by correctly transmitting and responding to probe packets while dropping and damaging normal network traffic. Here we design protocols that provide accurate information even when intermediate nodes *adversarially delay, drop, modify, inject or preferentially treat* packets in order to confound measurement. Our motivations for studying this adversarial threat model are threefold:

1. It covers active attacks. Our strong threat model covers a broad class of malicious behavior. Malicious adversaries can attack routing in order to draw packets to (or through) a node of their choosing [17], or compromise a routers along an existing path through the Internet [22], [27]. Biasing path-quality measurements allows the adversaries to evade detection, while continuing to degrade performance or impersonate the legitimate destination at will. In addition, ISPs have both the economic incentive and the technical means to preferentially handle probe packets, to hide discrimination against unwanted traffic like Skype [39] or BitTorrent [48], and evade detection of SLA violations. (In fact, commercial monitoring services, like Keynote, claim to employ "anti-gaming" techniques to prevent providers from biasing measurement results [30].) Finally, adversaries controlling arbitrary end hosts can add spoofed packets to the stream of traffic from one edge network to another, to confound simplistic measurement techniques (*e.g.,* maintaining a counter of received packets).

2. It covers all possible benign failures. In the adversarial setting, we avoid making *ad hoc* assumptions about the nature of failures caused by normal congestion, malfunction or misconfiguration. Even benign modification of packets may take place in a seemingly adversarial manner. For example, an MTU mismatch may cause a router to drop large packets while continuing to forward the small probe packets sent by ping or traceroute [34]. As another example, link-level CRC checks are surprisingly ineffective at detecting the kinds of errors that corrupt IP packets [45]. Since the adversarial model is the strongest possible model, any protocol that is robust in this setting is automatically robust to all other kind of failures.

3. It is challenging to satisfy in high-speed routers. We choose to work in a difficult space, where we assume the strongest possible adversarial model, and yet design solutions for high-speed routers on multi-Gbit/sec links, where computation and storage resources are extremely limited. We view it as an important research goal to understand what can and cannot be done in this setting, to inform practical decisions about what level of threats future networks should be designed to withstand. Furthermore, designing protocols for this adversarial setting is not simply a matter of adding standard cryptographic tools to existing non-adversarial measurement protocols. Indeed, naive ways of combining such protocols with cryptographic tools may be either insecure or less efficient (*e.g.,* encrypting and authenticating all traffic).

B. Our results

Secure sketch PQM. Despite this strong threat model, we present a secure PQM protocol that is competitive, in terms of storage and communication, with solutions designed for significantly weaker threat models [16], [21]. Our secure sketch PQM protocol, presented in Section III-B, allows a pair of cooperating sender and receiver routers to *detect* when the fraction of *packet delivery failures* on a path between them rises above a certain rate β (say $\beta = 0.01$). Our protocol achieves its low overhead by combining sublinear algorithms for sketching the second moment of a stream of items [1], [3], [15], [47] with simple cryptographic primitives like message authentication codes and pseudorandom functions.

The protocol, which is run by a sender and receiver sharing a symmetric secret key, only requires the transmission of two small control messages for every T data packets sent, while its storage overhead is limited to a single sketch (*i.e.,* an array of counters) of size $O(\log T)$ bits. If $T = 10^7$ packets are sent during an 100ms interval, our protocol requires between 250–600 bytes of storage at the source and destination, and control messages that can easily fit into a single IP packet. Moreover, our protocol does not mark, modify or authenticate data packets in any way, and so it may be implemented off the critical packet-processing path in the router. Not marking packets also makes our protocol backwards compatible with IP, minimizes latency at the router, allows the parties to turn on/off PQM protocols without the need to coordinate with each other, and avoids problems with increasing packet size and possibly exceeding the MTU. As such, we believe that this protocol is strong candidate for deployment in future networks, even in networks where our strong security guarantees are not be essential.

The performance and cost of any particular implementation of protocol would depend on memory speed and the particular choice of cryptographic primitives. As such, we analytically bound the different resources consumed —computation, storage and communication— (Section III-D), and also show somewhat better bounds through numerical experiments (Section III-E). In the course of this analysis we also present a new analysis of [15]'s sketching scheme that may be of independent interest (Theorem III.2). Moreover, the security of our protocol is based on the computation of cryptographic hash over the contents of every packet send during an interval. Fortunately, our protocols use cryptographic hash functions in an *online* setting, where an adversary has very limited time to break the security before the hash parameters are refreshed; in Section III-F we discuss exploiting this for fast packet hashing.

Composing sketches for secure FL. Next, we use the secure sketching protocol to construct a secure fault localization (FL) protocol (Section V-A). Our FL protocol is designed for a trusted sender and receiver that send traffic over a symmetric path of K nodes, where $\sqrt{K}$ nodes may be adversarial; if the rate of packet-delivery failures exceeds β, the sender can localize the responsible link. (Depending on the application, nodes could represent routers (and thus $K < 20$) or Autonomous Systems (and thus $K \approx 4$ on average).) The protocol requires each node on the path to share a symmetric key with the sender, and requires each node to store an $O(K^2 \log T)$-sized sketch. The communication overhead of the protocol is still only two control messages for every T packets sent, and no modifications to data packets are required.

Precise definitions of security. Evaluating the security of a protocol is often challenging. In many problem domains, *e.g.,* intrusion detection, the only viable approach is to enumerate a set of possible attacks, and then show how the protocol defends against these specific attacks. Fortunately, in our problem domain, a more comprehensive security evaluation is possible; we can give a precise definition of the functionality we require from the protocol, and then guarantee that the protocol can carry out these functions *even in the face of all possible attacks by an adversary* with a specific set of powers. Thus, we precisely define the powers that we give to the adversary and our requirements for secure PQM and FL protocols (Sections II-A, IV-A). To evaluate security, we prove that our protocols achieve their required functionalities, *no matter what* the adversary does, short of breaking the security of the basic cryptographic primitives (*e.g.,* message authentication codes and hash functions) from which the protocol is constructed (Section III-C, V-B). Due to space limitations, we defer some proofs to a technical report [25].

C. Bibliographic note and omitted results

This paper is a deeper treatment of a subset of results that appeared in earlier conference papers [11], [24]. Specifically, the security definitions we present here are the same as those used in these earlier papers. The secure sketch PQM protocol we discuss here was mentioned in [24], but the earlier paper concentrated on showing generic reduction from PQM to second moment estimation. Here we deepen and simplify the presentation by focusing on an instantiation of the secure sketch protocol using [15]'s sketching scheme, and present new results on fast packet hashing (Section III-F) and stronger versions of theorems from [24] (Theorem III.2 and Theorem E.1 in our technical report [25]). We also present completely new results showing the secure sketch PQM can be composed to create an FL protocol (Section V).

In focusing on the construction of efficient protocols, we have chosen to omit a number of negative results that we developed in [11], [24]. These negative results indicate that *any* secure PQM or FL protocol would need to employ the same basic security machinery (secret keys and cryptographic operations) used by our protocols. Specifically:

1. In [24] we showed that any PQM protocol satisfying Definition II.1 requires (1) Alice and Bob to have some form of shared secret (*e.g.,* shared symmetric secret keys), and (2) the shared secret must be used in a "cryptographically-strong" manner. To prove the latter, we used a reduction that shows that any secure PQM protocol is at least as complex as a secure keyed identification scheme (KIS). Because KIS are equivalent to cryptographic tasks like encryption and message authentication [28], this proves that cryptographic computations are necessary for the security of *any* PQM protocol.

2. In [11] we show that any FL protocol satisfying Definition IV-A requires a key infrastructure, or more precisely, that intermediate nodes and Alice and Bob must all share some secret information. We also used block-box separation techniques from cryptography [29] to give evidence that a secure FL protocol must use these keys in a cryptographically-strong manner at *every* node on the path.

In addition to these lower bounds, we also omitted a number of protocols from in [11], [24], that are less efficient in terms of storage and communication that the ones presented here, including (1) secure PQM and FL protocols that use PRF-based packet sampling, (2) PQM protocols that use public-key cryptography and timed information release to remove the need for symmetric shared secrets between Alice and Bob, and (3) FL protocols that are designed to detect and localize failures on a per-packet basis.

Online Appendices. Due to space limitations, we have had to defer the appendices of this paper to our technical report [25].

II. Threat Model for Path Quality Monitoring

We first define security for path quality monitoring.

A. Security definition for path-quality monitoring (PQM)

In our model, a source *Alice* sends packets to a trusted destination *Bob* over a path through the Internet. Fix a set of T consecutive packets sent by Alice, which we call an *interval*, we define a *packet delivery failure* to be any instance where a packet that was sent by Alice during the interval fails to arrive unmodified at Bob (before the last packet in the interval arrives at Bob). An adversary *Eve* can sit anywhere on the path between Alice and Bob, and we empower Eve to drop, modify, or delay every packet, or to add her own packets. A *path quality monitoring (PQM) protocol* is a protocol that Alice and Bob run to detect whether the number of failures during the interval exceeds a certain fraction of total packets transmitted.

Definition II.1. Given parameters $0 < \alpha < \beta < 1$ and $0 < \delta < 1$, we say a protocol is a *(α, β, δ) secure PQM protocol* if, letting T be the number of packets sent during the interval:

1) (Few false negatives.) In the *malicious case*, where an adversary Eve can drop, modify, delay, or add packets, the protocol must raise an alarm with probability at least $1 - \delta$ if more than βT packet-delivery failures occur.

2) (Few false positives.) In the *benign case*, where no intermediate node is adversarial (*i.e.,* no packets are added or modified on the path, but packets may be reordered / dropped due to congestion), the protocol must alarm with probability at most δ if at most αT packet-delivery failures occur.

We assume that the T packets sent during an interval are distinct, because of natural variation in packet contents, and the fact that even successive packets sent by the same host have different ID fields in the IP header [21] (note that even retransmissions of the same TCP segment correspond to distinct IP packets, because of the IP ID field).

B. Properties of our security definition

Our definition is motivated by our intended application of enabling routing decisions or detecting SLA violations. It's most important security guarantee is that, regardless of Eve's actions, Eve cannot prevent Alice from raising an alarm when the failure rate for packets that Alice sent to Bob exceeds β. As such, our definition encompasses attacks by nodes on the data path that include (but of course are not limited to): colluding nodes that work together to hide packet loss, an adversarial node that intelligently injects packets based on timing observations or deep packet inspection, a node that preferentially treats packets that it knows are part of the PQM protocol, and a node that masks packet loss by injecting an equal number of nonsense packets onto the data path. We emphasize that we never make any assumptions on the distribution of packet loss on the path; our model allows for any possible failure model, including one where, say, packet loss is correlated across different packets.

On the other hand, we only require PQM protocols to detect failures (so that SLA violations can be confirmed, or packets can be rerouted) but not to *prevent* them. Moreover, PQM protocols must only detect if the number of failures exceeds a certain threshold, rather than determining exactly how many failures occurred. (While solutions that exactly count failures certainly exist, they typically require cryptographically authenticating and/or encrypting all traffic, which we wish to avoid here.) Third, we do not require our protocols to distinguish between packet failures occurring due to adversarial tampering or due to benign congestion or malfunction.

Next, while our security definition requires that our protocols do not raise a (false) alarm when the one-way failure rate is less than α for the *benign setting*, we do allow for the possibility of raising an alarm due to adversarial tampering even when fewer than an α-fraction of failures occur. This is because Eve can always make a path look worse by selectively dropping all PQM protocol messages (*e.g.,* acknowledgments, report messages) that Bob sends to Alice, even if all the original packets that Alice sent to Bob were actually delivered.[1]

[1]We will assume that any acknowledgment or report messages that Bob sends to Alice are sent repeatedly to ensure that, with high probability, they are not dropped due to normal congestion.

In such cases, the PQM protocol will raise an alarm, and the router should look for a different path. We also do not model denial of service attacks, where an adversary exhausts capacity by flooding the path with packets; these attacks can be addressed using standard techniques, *e.g.,* rate limiting.

Choosing α, β. In principle $\alpha, \beta > 0$ can be chosen arbitrarily. However, several practical issues constrain the choice of these parameters. In Section III-D we find that the communication and storage overhead of our protocols is related to the ratio $\frac{\alpha}{\beta}$; a smaller ratio leads to less overhead. The value of α is also constrained by issues related to interval synchronization; we discuss this in Appendix A.

III. Secure sketch PQM

In our secure sketch PQM protocol, Alice and Bob aggregate traffic Alice sends to Bob into a short data structure called a *sketch*. At the end of the interval, Bob sends his sketch to Alice and she compares the sketches to decide whether the failure rate exceeded β. We first discuss notation and the cryptographic building blocks used in our protocol (Section III-A), and then describe the protocol (Section III-B). Next, we discuss its security (Section III-C) and derive analytic bounds that explain the relationship between the protocol's security its storage and communication overhead (Section III-D). We further analyze this relationship using numerical experiments (Section III-E) and conclude by discussing techniques for fast packet hashing (Section III-F).

A. Preliminaries

Notation. We use the notation $[\kappa]$ to represent the set of integers $\{1, ...\kappa\}$. $\mathbf{v}$ is a vector, and its i^{th} entry is v_i. The second moment of a vector $\mathbf{v}$ is $\|\mathbf{v}\|_2^2 = \sum_i (v_i)^2$, and the first moment of a vector is $|\mathbf{v}|_1 = \sum_i |v_i|$. We say that a quantity w (ε, δ)*-estimates* a quantity v if

$$\Pr[(1-\varepsilon)v \leq w \leq (1+\varepsilon)v] \geq 1-\delta$$

We also use the following cryptographic building blocks:

Pseudorandom Function (PRF). A PRF is a keyed function $\mathrm{PRF}_k(\cdot)$ that maps an arbitrary length string to an n-bit string using a key k [26]. If the key k is chosen uniformly at random, then to an adversary with no knowledge of k, the output of the function $\mathrm{PRF}_k(\cdot)$ looks totally unpredictable and cannot be distinguished (except with an insignificant probability) from a truly random function, where each input is mapped to a independent uniformly-random output. Hence, in our analysis we may treat $\mathrm{PRF}_k(\cdot)$ as if it is truly random. PRFs are typically realized via a full-fledged cryptographic hash functions such as SHA-512 in HMAC mode [31], or with a block cipher like AES in a MAC mode of operation.

Keyed packet-hashing function. All our protocols require a hash computation on the invariant contents of every sent packet;[2] and all subsequent processing of the packet relies only on this hash value. Our packet-hashing function will always be keyed with an ephemeral interval key k_u, which is used only for the duration of single interval consisting of T packets (typically $T = 10^7$ and an interval lasts for about 100ms). Once the interval ends, k_u no longer needs to be kept secret (because Bob has already received the packets sent during the interval). In Sections III-D, III-F we consider instantiating the keyed-packet hashing function model with both a PRF and a 4-wise independent hash function [14]. In either case, our keyed packet-hashing function should be (a) fast enough to keep up with multi-Gbit/sec packet streams, (b) remain secure for the duration of an interval, *i.e.,* after about $T = 10^7$ applications and/or for about 100ms.

Message Authentication Code (MAC) is a basic cryptographic primitive that can be realized using a PRF: using a shared key k, for a message m, one party will send $(m, \mathrm{PRF}_k(m))$ and the other party can verify that a pair (m, t) satisfies $t = \mathrm{PRF}_k(m)$. The value t cannot be feasibly forged by an adversary that does not know k. We use the notation $[m]_k$ to denote $(m, \mathrm{PRF}_k(m))$, a message m MAC'd with key k.

B. Description of the secure sketch protocol

Our protocol works in intervals. We assume Alice and Bob share a secret master keys (k_1, k_2), and derive an ephemeral interval key k_u for each interval u. Pairwise master keys are be derived via *e.g.,* authenticated Diffie-Hellman key exchange (as used in TLS/SSL [19]) or some other out-of-band secure channel. Interval keys are computed by using a pseudorandom function PRF keyed with master key k_2 (*i.e.,* let $k_u = \mathrm{PRF}_{k_2}(u)$), and Appendix Adetails how intervals can synchronized. Alice and Bob also store a sketch, or an array of N counters, each of which can count from $[-\kappa, +\kappa]$. Within interval u, our secure sketch protocol proceeds in four phases:

(Sketch.) Alice runs a sketching algorithm, using a keyed packet-hashing function $h_{k_u}(\cdot)$ keyed with secret interval key k_u to incrementally compute a sketch $\mathbf{w}_\mathrm{A}$ of the set of packet she sends during the interval. $h_{k_u}(\cdot)$ may be a 4-wise independent hash function, or a PRF (see Section III-D). For each packet d that Alice sends, she (a) computes $h_{k_u}(d)$ to obtain a (pseudorandom or 4-wise independent) pair of numbers (i, b) where $i \in [N]$ and $b \in \{-1, +1\}$, and (b) adds b to the i^{th} counter in the sketch $\mathbf{w}_\mathrm{A}$. Bob similarly uses $h_{k_u}(\cdot)$ to compute a sketch $\mathbf{w}_\mathrm{B}$ of the set of packets he receives.

(Interval End.) After sending the T^{th} packet in the interval, Alice sends an 'Interval End' message to Bob containing her sketch $\mathbf{w}_\mathrm{A}$ and the next interval number $u+1$, which she signs with a message authentication code (MAC) keyed with k_1. She then refreshes her sketch (*i.e.,* sets $\mathbf{w}_\mathrm{B} = 0$) and computes the next interval key using a pseudorandom function PRF keyed with master key k_2 (*i.e.,* lets $k_{u+1} = \mathrm{PRF}_{k_2}(u+1)$).

(Report.) Upon receiving the 'Interval End' message and verifying the correctness of its MAC, Bob computes the *difference sketch* $\mathbf{w}_\mathrm{A} - \mathbf{w}_\mathrm{B}$. Bob then sends a 'Report' message to Alice, containing the difference sketch and the current

[2]Whenever the packet-hashing function is applied to a packet, the non-invariant fields of the packet header are discarded from the input. In the case of IPv4, this means excluding the ToS, TTL and IP checksum (see [21, SectionII.A]).

interval number u, which he signs with a MAC keyed with k_2. Bob then refreshes his sketch and computes the interval key for the next interval $u+1$.

(Security Check.) Upon verifying the MAC on the 'Report' message, Alice uses the difference sketch $\mathbf{w}_\mathrm{A} - \mathbf{w}_\mathrm{B}$ to raise an alarm if

$$\|\mathbf{w}_\mathrm{A} - \mathbf{w}_\mathrm{B}\|_2^2 > \Gamma = 2\frac{\alpha\beta T}{\beta+\alpha}$$

or if the 'Report' message is missing or has an invalid MAC. We call $\|\mathbf{w}_\mathrm{A} - \mathbf{w}_\mathrm{B}\|_2^2$ the *estimator* and Γ the *threshold*.

Our protocol has a few attractive properties. First, we need only transmit two control messages ('Interval End', and 'Report') per interval; we require no other packet modifications. Second, the 'Security Check' phase can be computed offline. Thirdly, Alice and Bob need only store single sketch at any given time; at the end of each interval, Alice and Bob immediately transmit their sketches as control messages, refresh their sketches, and begin monitoring a new interval. Finally, our protocol has low storage and communication requirement; in Section III-D we show that the sketch is not much larger than an ordinary counter, having size $O(\log T)$ where T is the number of packets sent during the interval. In fact, the number of counters N in the sketch depends only on security parameters α, β, δ but not on T, while the size of each counter $\log\kappa$ is $O(\log T)$.

A performance optimization. To reduce resource consumption, a router can 'turn off' the secure sketching protocol during certain intervals. However, to prevent an adversary from exploiting this to bias monitoring results (*e.g.,* by selectively dropping packets when the protocol is 'off', and behaving itself while the protocol is 'on'), intervals when the protocol is 'off' must be indistinguishable from intervals when the protocol is 'on'. Fortunately, the only indication that the protocol is 'on' are the two control messages ('Interval End' and 'Report'). Thus, while Alice and Bob need not compute hashes over packet contents or to maintain sketches in an 'off' interval, we still require (a) Alice to count the number of packets she sends to Bob and send a dummy 'Interval End' message each time the counter reaches T, and (b) Bob to respond with a dummy 'Report' packet. To make the dummy control messages indistinguishable from real control messages, we will also require (c) that *all* information fields in the control messages sent by the protocol are encrypted and padded to a fixed length (and subsequently authenticated). Selection of 'on' intervals should also be random, to prevent an adversary from selectively attacking the 'off' intervals by using side-channel information (*e.g.,* observing if the sender switches to a new path) to distinguish between 'on' or 'off' intervals.

C. Security analysis

We now prove the security of our protocol by explaining the connection between PQM, and sketches that can be used to estimate the second moment of a set of items. To do this, we explain why the process of sketching can be viewed as a linear transformation that preserves second moment, and then show why this suffices to satisfy our security definition.

Packet streams as vectors. We can think of the stream of packets sent by Alice during a given interval as vector. Let U be the "universe" of all possible packets sent by Alice (*e.g.,* if packets are 1500 bytes long then $|U| \approx 2^{1500\cdot 8}$). Let the characteristic vector of a stream of packets be a $|U|$-dimensional vector that has integer c in the position corresponding to packet x if packet x appears in the stream c times (*e.g.,* for the stream 1,2,4,2,2 of packets drawn from universe $U = [4]$, the characteristic vector is [1 3 0 1].) Naturally, a characteristic vector is too long (*e.g.,* $2^{1500\cdot 8}$) to be represented explicitly; we will use it only as a tool for analyzing the security of our protocol. Let $\mathbf{v}_\mathrm{A}$ be the characteristic vector for the stream of packets sent by Alice during a given interval, and analogously $\mathbf{v}_\mathrm{B}$ for the stream of packets received by Bob.

Sketching as matrix multiplication. The sketch we use in our protocol was first proposed by [15]. While Section III-B presented sketching as an incremental streaming process applied individually to each packet, we now view sketching as a single linear transformation applied to a characteristic vector. Specifically, for a sketch $\mathbf{w}$ computed from the packet stream represented by characteristic vector $\mathbf{v}$ per the description in Section III-B, we can write

$$\mathbf{w} = R\mathbf{v} \tag{1}$$

where R is a $N \times |U|$ matrix that is completely determined by the keyed packet-hashing function $h_{k_u}(\cdot)$; specifically, for every packet $d \in U$, the d^{th} column of matrix R has its i^{th} row equal to b, where $(i,b) = h_{k_u}(d)$ for $i \in [N]$ and $b \in \{-1,+1\}$, and all its other rows are zero.

A sufficiently large sketch $\mathbf{w}$ can (ε,δ)-estimate the second moment of $\mathbf{v}$. That is, for a sketch with N counters that counts from $[-\kappa,\kappa]$, where the packet hashing function is either a PRF or a 4-wise independent hash function that is chosen independently of the characteristic vector $\mathbf{v}$, then

$$(1-\varepsilon)\|\mathbf{v}\|_2^2 < \|\mathbf{w}\|_2^2 = \|R\mathbf{v}\|_2^2 < (1+\varepsilon)\|\mathbf{v}\|_2^2 \tag{2}$$

with probability $1-\delta$ as long as N and κ are sufficiently large [15], [47]. In Section III-D, we show how to size N and κ so that (2) holds.

PQM as second moment estimation. We are now ready to see how PQM can be derived from second moment estimation. Observe that characteristic vector $\mathbf{v}_\mathrm{A} - \mathbf{v}_\mathrm{B}$ can be decomposed into two vectors $\mathbf{d} - \mathbf{a}$. Vector $\mathbf{d}$ is the characteristic vector of packets *dropped* on the path from Alice to Bob, and contains the non-negative components of $\mathbf{v}_B - \mathbf{v}_A$. Vector $\mathbf{a}$ is the characteristic vector of packets *added* on the path from Alice to Bob, and corresponds to the non-positive components of $\mathbf{v}_B - \mathbf{v}_A$. (Note that a packet modification amounts to a dropped packet *plus* an added packet.) Also notice that the non-zero coordinates of $\mathbf{d}$ and $\mathbf{a}$ are disjoint. Now let D be the number of packets dropped on the path from Alice to Bob during the interval, and let A be the number of packets added during the interval. Thus, we have the following simple and very useful identity:

$$||\mathbf{v}_\mathrm{A} - \mathbf{v}_\mathrm{B}||_p^p = ||\mathbf{d}||_p^p + ||\mathbf{a}||_p^p = D + ||\mathbf{a}||_p^p \geq D + A \tag{3}$$

The identity tells us that, for any integer $p \geq 1$, the p^{th} moment of the characteristic vector $\mathbf{v}_\mathrm{A} - \mathbf{v}_\mathrm{B}$ overestimates the number of packets that are added and dropped during a given interval. The first equality in (3) follows because the non-zero coordinates of $\mathbf{d}$ and $\mathbf{a}$ are disjoint. The second equality follows because every packet that Alice send is unique, so that that $\mathbf{d}$ is a $\{0,1\}$-vector. Finally, the last inequality follows because $\mathbf{a}$ is an integer vector, so that for any $p \geq 1$, it follows that $\|\mathbf{a}\|_2^2 \geq |\mathbf{a}|_1 = A$ with equality when $p = 1$.

We are now ready to prove security. Set $\varepsilon = \frac{\beta-\alpha}{\beta+\alpha}$.

1) Few false positives. To satisfy this condition we need to consider the *benign case* in which at most $\|\mathbf{d}\|_2^2 = D \leq \alpha T$ packets are dropped during the interval, and there is no adversary Eve on the path. Because Eve is absent, we can assume that no packets are added, so that $\|\mathbf{a}\|_2^2 = 0$. Equation 3 gives

$$\|\mathbf{v}_{\mathrm{A}} - \mathbf{v}_{\mathrm{B}}\|_2^2 = \|\mathbf{a}\|_2^2 + \|\mathbf{d}\|_2^2 \leq \alpha T \tag{4}$$

Next, we observe that in the benign case, the keyed-packet hashing function $h_{k_u}(\cdot)$ is chosen independently of $\mathbf{v}_{\mathrm{A}}$ and $\mathbf{v}_{\mathrm{B}}$. Thus, for an appropriately-sized sketch (where the number of counters N and the size of each counter κ is sufficiently large), we know that equation (2) holds with probability $1-\delta$. Thus we have:

$$\begin{aligned}
\|\mathbf{w}_{\mathrm{A}} - \mathbf{w}_{\mathrm{B}}\|_2^2 &= \|R\mathbf{v}_{\mathrm{A}} - R\mathbf{v}_{\mathrm{B}}\|_2^2 && \text{(From (1))}\\
&= \|R(\mathbf{v}_{\mathrm{A}} - \mathbf{v}_{\mathrm{B}})\|_2^2 \\
&\leq (1+\varepsilon)\|\mathbf{v}_{\mathrm{A}} - \mathbf{v}_{\mathrm{B}}\|_2^2 && \text{(From (2))}\\
&\leq (1+\varepsilon)\alpha T && \text{(From (4))}\\
&= 2\tfrac{\alpha\beta T}{\beta+\alpha}T = \Gamma && (\text{Since } \varepsilon = \tfrac{\beta-\alpha}{\beta+\alpha})
\end{aligned}$$

Thus, Alice will not raise an alarm in the benign case, with probability $1-\delta$, if N, κ are sufficiently large.

2) Few false negatives. To satisfy this condition, we need to consider the malicious case. First observe that in this case, Eve cannot forge the 'Interval End' or 'Report' control messages, since the control messages are authenticated using a secure MAC (and dropping the report will only cause Alice to raise an alarm). Thus, we can suppose that Alice correctly receives the difference sketch $\mathbf{w}_{\mathrm{A}} - \mathbf{w}_{\mathrm{B}}$ from the 'Report' message. In the malicious case, Eve drops $D \geq \beta T$ packets, and adds an arbitrary number of (potentially non-unique) packets $A \geq 0$. In this case, (3) tell us that

$$\|\mathbf{v}_{\mathrm{A}} - \mathbf{v}_{\mathrm{B}}\|_2^2 = \|\mathbf{a}\|_2^2 + \|\mathbf{d}\|_2^2 \geq \beta T \tag{5}$$

Now, observe that (b) k_u is chosen independently of the packets sent by Alice $\mathbf{v}_{\mathrm{A}}$, and (b) k_u is kept secret from Eve until packets reach Bob at the end of the interval. k_u is therefore independent or the packets received by Bob $\mathbf{v}_{\mathrm{B}}$ (some of which may have been sent by Eve). We can therefore assume equation (2) holds with probability $1-\delta$ for a sketch with an appropriately-sized N and κ. Thus

$$\begin{aligned}
\|\mathbf{w}_{\mathrm{A}} - \mathbf{w}_{\mathrm{B}}\|_2^2 &= \|R(\mathbf{v}_{\mathrm{A}} - \mathbf{v}_{\mathrm{B}})\|_2^2 && \text{(From (1))}\\
&\geq (1-\varepsilon)\|\mathbf{v}_{\mathrm{A}} - \mathbf{v}_{\mathrm{B}}\|_2^2 && \text{(From (2))}\\
&\geq (1-\varepsilon)\beta T && \text{(From (5))}\\
&= \tfrac{2\beta\alpha}{\beta+\alpha}T \;=\; \Gamma && (\text{Since } \varepsilon = \tfrac{\beta-\alpha}{\beta+\alpha})
\end{aligned}$$

with probability $1-\delta$ Alice will thus alarm in the malicious case, with probability $1-\delta$, if N, κ are sufficiently large.

This concludes our argument, since Alice can use the decision threshold Γ to decide between the benign and malicious cases with probability $1-\delta$, as long as the sketch is sized appropriately.

D. Sizing the sketch

We now move on to determining the parameters N (number of counters in the sketch) and κ (the size of each counter) when the keyed packet-hashing function $h_k(\cdot)$ is (a) 4-wise independent hash function, and (b) it is a PRF. In both cases, however, we shall show that N depends only on α, β, δ, while $\kappa = O(\log T)$.

Sizing each counter. Each counter holds integers in $[-\kappa, +\kappa]$. We therefore set

$$\log_2(2\kappa) = 1 + \tfrac{1}{2}\log_2\left(4\tfrac{T}{N}\ln(\tfrac{200N}{\delta})\right) \quad \text{bits / counter} \tag{6}$$

to ensure that the probability that each counter overflows is at most $\frac{\delta}{N}\frac{1}{100}$; it follows from the union bound that, with probability $1-\delta/100$, no counter will overflow. To see how we obtained (6), observe that if X_i is a random variable that equals 1 with probability $\frac{1}{2N}$, -1 with probability $\frac{1}{2N}$, and 0 otherwise, then the count in each bin is the random variable $X = \sum_{i=1}^{T} X_i$. Then, adapting the Chernoff bound that appears in [33], we have that

$$\Pr[\;|X| \;\geq\; \kappa\;] \;\leq\; 2\exp(-\tfrac{\kappa^2}{4T\mathrm{VAR}[X_i]}) \;\leq\; \tfrac{\delta}{100N}$$

Finally, we get (6) since $\mathrm{VAR}[X_i] = 1/N$.

Sizing the number of counters N. While N depends only on α, β, δ, its exact value depends on the instantiation of the keyed-packet hashing function. We first consider its instantiation with a 4-wise independent hash, and then with a PRF.

Packet-hashing with 4-wise independent hashes. A 4-wise independent hash is a function $h: \{0,1\}^{|k_u|} \times \{0,1\}^* \to \{0,1\}^n$ that guarantees that for any four distinct inputs x_1, x_2, x_3, x_4 and (possibly non-distinct) outputs y_1, y_2, y_3, y_4, then

$$\Pr\left[h_{k_u}(x_i) = y, \forall i = 1...4\right] = \left(\tfrac{1}{2^n}\right)^4 \tag{7}$$

where the probability is over the choice of k_u used to key h. A 4-wise independent hash can be realized using polynomials of degree 3 [14]. For example, to compute $h_{k_u}(x)$ set key $k_u = (a_0, a_1, a_2, a_3)$ and output $a_3x^3 + a_2x^2 + a_1x + a_0$; this can be done in three multiplications with Horner's rule.

Thorup and Zhang [47] showed that 4-wise independent hashing suffices to realize second moment estimation using [15]'s sketching algorithm:

Theorem III.1 (From [47])**.** *If we construct sketch $\mathbf{w}$ using two 4-wise independent packet hashing functions $h: U \to [N]$ and $b: U \to \{-1,1\}$ then*

$$\mathbb{E}[\|\mathbf{w}\|_2^2] = \|\mathbf{v}\|_2^2 \tag{8}$$

$$\mathbb{VAR}[\|\mathbf{w}\|_2^2] = \tfrac{2}{N}(\|\mathbf{v}\|_2^4 - \|\mathbf{v}\|_4^4) \tag{9}$$

And $\|\mathbf{w}\|_2^2$ (ε,δ)-estimates $\|\mathbf{v}\|_2^2$ as long as $N > \frac{2}{\varepsilon^2\delta}$.

Since $\varepsilon = \frac{\beta-\alpha}{\beta+\alpha}$, it suffices to take

$$N \geq \tfrac{2}{\varepsilon^2 0.99\delta} = \tfrac{2.02}{\delta}\tfrac{(\beta+\alpha)^2}{(\beta-\alpha)^2} \tag{10}$$

counters in our sketch. What is remarkable about this result is the fact that N, the number of counters in our sketch, is completely independent of the number of packets T sent during the interval! We shall see in Section III-E that this means that our sketches can monitor a large number of packets using a very limited amount of space.

Packet-hashing with a PRF. Since every PRF is also indistinguishable from a 4-wise independent hash function, choosing N as in (7) also suffices when the keyed packet-hashing function is instantiated with a PRF. However, we can do better when the packet hashing function is a PRF. Specifically, can take $N = O\left(\frac{1}{\varepsilon^2}\log\frac{1}{\delta}\right)$ by exploiting a few special properties of our PQM setting, including (a) the assumption that Alice sends unique packets, and (b) the fact that we only care about deciding whether $\|\mathbf{v}\|_2^2$ lies above or below a threshold, rather than getting an accurate estimate of $\|\mathbf{v}\|_2^2$.

The following theorem, which may be of independent interest, supposes that packet hashing uses two independent random functions: one to chose $i \in [N]$ and another to choose $b \in \{-1, 1\}$. We can therefore think of the sketching matrix R as chosen *uniformly at random* from set of matrices $\mathcal{S}$, where $\mathcal{S}$ is the set of $N \times |U|$ matrices where each column contains a single ± 1 entry in one row, and zeros in all other rows. We prove the following in Appendix E of [25]:

Theorem III.2. *For any vector $\mathbf{v} \in \mathbb{Z}^U$, choose the $N \times |U|$ sketching matrix R uniformly at random from $\mathcal{S}$. If $\mathbf{w} = R\mathbf{v}$, then for all $\varepsilon \in [0, 1)$ and η such that*

$$\left(\frac{1-\eta}{1+\eta}\right)^2 = \max\left(\frac{1+\frac{\varepsilon}{2}}{1+\varepsilon}\ ,\ \frac{1-\frac{3\varepsilon}{4}}{1-\frac{\varepsilon}{2}}\right) \tag{11}$$

the following two items occur with probability at least $1 - \delta$:

1) *If $\mathbf{v} \in \{-1, 0, 1\}^U$, and $\|\mathbf{v}\|_2^2 \leq q$, then $\|\mathbf{w}\|_2^2 < (1 + \varepsilon)q$.*
2) *The number of* non-zero *entries in $\mathbf{v}$ is r, then $\|\mathbf{w}\|_2^2 > (1 - \varepsilon)r$.*

as long as

$$N \geq \frac{24}{\varepsilon^2}\ln\frac{2}{\delta} \tag{12}$$

$$q, r \geq \frac{3N}{\eta^2}\ln\frac{4N}{\delta} \tag{13}$$

To apply the theorem into our setting, we plug $\varepsilon = \frac{\beta-\alpha}{\beta+\alpha}$ and δ into (12) to obtain a bound on N, the number of counters in our sketch. Then, we plug ε in (11) to determine η. Finally, we set $q = \alpha T$ and $r = \beta T$ in (13) to obtain a lower bound on T, the minimum number of packets sent in an interval. (This lower bound on T is awkward, and we believe that it is an artifact of our proof technique; indeed, numerical experiments in Section III-E suggest that this bound on T is not tight.)

We show why these parameters guarantee security. First, assume that the PRF used for packet hashing is indistinguishable from a random function. Then, using the language of Section III-C, the false positive condition is satisfied because in the benign case we have $\mathbf{v}_\mathrm{A} - \mathbf{v}_\mathrm{B} \in \{0, 1\}^U$ (since packets may only be dropped, and all packets are distinct) and $\|\mathbf{v}_\mathrm{A} - \mathbf{v}_\mathrm{B}\|_2^2 = D \leq \alpha T = q$, so with probability $1 - \delta$, the first item in Theorem III.2 gives

$$\|\mathbf{w}_\mathrm{A} - \mathbf{w}_\mathrm{B}\|_2^2 \leq (1+\varepsilon)\|\mathbf{v}_\mathrm{A} - \mathbf{v}_\mathrm{B}\|_2^2 \leq (1+\varepsilon)\alpha T = \frac{2\alpha\beta}{\alpha+\beta}T = \Gamma$$

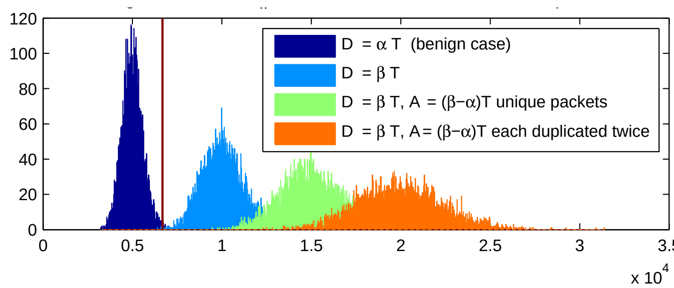

Fig. 1. Distribution of estimator $\|\mathbf{w}_\mathrm{A} - \mathbf{w}_\mathrm{B}\|_2$ using packet-hashing with a PRF and with $N = 300$, $T = 10^6$, $\beta = 2\alpha = 1\%$ and threshold $\Gamma = 6667$, computed via numerical experiments.

The false negative condition is satisfied because in the malicious case, we have $\|\mathbf{v}_B - \mathbf{v}_A\|_2^2 = \|\mathbf{d}\|_2^2 + \|\mathbf{a}\|_2^2$ where $\|\mathbf{d}\|_2^2$ is a $\{0, 1\}$-vector (since all dropped packets are distinct) that has at least $r = \beta T$ non-zero entries. So, with probability $1 - \delta$, the second item in Theorem III.2 gives

$$\|\mathbf{w}_\mathrm{A} - \mathbf{w}_\mathrm{B}\|_2^2 \geq (1-\varepsilon)\|\mathbf{v}_\mathrm{A} - \mathbf{v}_\mathrm{B}\|_2^2 \geq (1-\varepsilon)r \geq (1-\varepsilon)\beta T = \frac{2\alpha\beta}{\alpha+\beta}T \quad =$$

E. Some sample parameters and experiments

We now compute the size of our sketch, using both our analytic results and numerical experiments, for the following sample parameters: We suppose the detection threshold is $\beta = 0.01$, the false alarm threshold is $\alpha = \beta/2$ and about $T = 10^7$ packets are sent during an interval. We will require a confidence of $1 - \delta = 99\%$.

Analytic results. We plug these parameters into our analytic results. When 4-wise independent hashing is used with these parameters, equation (10) indicates that we require $N = 1800$ counters, and equation (6) requires 10 bits/counter; the total size of the sketch is therefore 2.25KB. Storage become even smaller when we use a PRF; applying the refined version of Theorem III.2 in Appendix E of our technical report [25] to obtain bounds on N, we find that we can use $N = 300$ counters if there are at least $T_{min} = 1.2 \times 10^{10}$ packets in the interval, for a total sketch size of only 375B!

Numerical experiments. We preformed a number of numerical experiments of the case where keyed-packet hashing function is instantiated with PRF. In each numerical experiment, we operate on synthetic traffic, where we model every distinct packet sent during the interval with fresh pair of uniformly-independent random numbers (i, b) where $i \in [N]$ and $b \in \{-1, 1\}$, and use these numbers to create the difference sketch $\mathbf{w}_\mathrm{A} - \mathbf{w}_\mathrm{B}$ in the natural way (incrementing the i^{th} counter by b every time that packet appeared in the stream); the final output of the numerical experiment is the estimator $\|\mathbf{w}_\mathrm{A} - \mathbf{w}_\mathrm{B}\|_2^2$. For a given stream of packets, we can repeat this process multiple times to obtain the distribution of the estimator $\|\mathbf{w}_\mathrm{A} - \mathbf{w}_\mathrm{B}\|_2^2$.

Figure 1. Figure 1 shows the resulting distribution of the $\|\mathbf{w}_\mathrm{A} - \mathbf{w}_\mathrm{B}\|_2^2$ for a number of cases. From left to right, we have: The benign case where $D = \alpha T$ (here we want the estimator to be below the threshold Γ so that Alice does not raise an alarm), and three cases where $D = \beta T$ (here we want Alice to raise an alarm): a case where Eve does not add any packets, a case where Eve adds $A = (\beta - \alpha)T$ distinct packets, and a case where Eve adds a total of $A = (\beta - \alpha)T$

β/α	N	Sketch Size (in bytes) $T = 10^4$	$T = 10^5$	$T = 10^6$	$T = 10^7$	$T = 10^8$
2	128	128B	144B	176B	208B	208B
4	64	64B	88B	88B	104B	104B
8	32	36B	48B	48B	52B	52B
16	32	36B	48B	48B	52B	52B
32	32	36B	48B	48B	52B	52B
64	32	36B	48B	48B	52B	52B

Fig. 2. Minimum number of counters N in the sketch when N is taken as a power of 2, computed via numerical experiments where keyed packet hashing is performed using a PRF. The number of bits per counter is obtained from (6). We fix $\beta = \delta = 1\%$.

packets where each packet is duplicated twice. The figure reveals that the threshold $\Gamma = 2\frac{\alpha\beta T}{\alpha+\beta}$ clearly distinguishes between cases where $D = \beta T$ and the benign cases where $D = \alpha T$. Moreover, when Eve adds packets to the link, she only increases the probability that Alice raises an alarm, as predicted by equation (3). Figure 1 also suggests that taking $N = 300$ suffices even if we have shorter interval lengths of $T = 10^6$, suggesting that the awkward bound of T in Theorem III.2 is an artifact of our proof technique.

Figure 2. We preformed more numerical experiments to determine N, the number of bins in the sketch, for a given choice of α, β, δ. Recall from Section III-C that threshold $\Gamma = 2\frac{\alpha\beta T}{\alpha+\beta}$ must be able to distinguish between $\|\mathbf{w}_\mathrm{A} - \mathbf{w}_\mathrm{B}\|_2^2$ in the benign case vs. the malicious case, with probability $1 - \delta$. We therefore need to compare the extremes of the benign and the malicious cases. In the malicious case, the expected value of $\|\mathbf{w}_\mathrm{A} - \mathbf{w}_\mathrm{B}\|_2^2$ is minimized when the number of dropped packets $D = \beta T$ and the number of added packets is $A = 0$. Meanwhile, in the benign case, the expected value of $\|\mathbf{w}_\mathrm{A} - \mathbf{w}_\mathrm{B}\|_2^2$ is maximized when $D = \alpha T$. (This follows from equations (8) and (3) and is confirmed by Figure 1.) We therefore preform numerical experiments (as in Figure 1) to obtain the distribution of $\|\mathbf{w}_\mathrm{A} - \mathbf{w}_\mathrm{B}\|_2^2$ in these two extreme cases. We do this for various values of N that are powers of 2,[3]and then find the smallest value of N for which threshold Γ is able to distinguish between the two extreme cases with probability at least $1 - \delta$.

Our results are presented in Figure 2. We see that N varies with the ratio β/α. Meanwhile, as long as there are $T > 1/\alpha$ packets/interval, the choice of T does not impact the value of N; for a given β/α ratio, the minimum choice of N as power of 2 was the same for any value of T ranging from 10^4 to 10^8, which suggests that the lower bound on T in Theorem III.2 is not tight. Moreover, Figure 2 also confirms that since total sketch size grows logarithmically with T (per (6)), the total storage required by our sketching scheme remain modest.

F. Implementation issues and fast packet hashing.

As we see it, two main barriers stand in the way of the deployment of our secure sketch PQM protocol.

The first is establishing shared (symmetric) cryptographic keys between Alice and Bob; unfortunately, this overhead is unavoidable, since our lower bound in [24] establishes that keys are *necessary* for any secure PQM scheme satisfying our definition. However, shared keys can always be established in an enterprise setting, where *e.g.,* a central office (Alice) wants to monitor its connection to a branch office or datacenter (Bob) over the public Internet. Moreover, if a public-key infrastructure is in place (either localized within the enterprise, or just using the SSL/TLS PKI) a key-exchange protocol [19] could always be used to establish the shared keys.

The second barrier is computational overhead. The most expensive part of our sketching protocol is the computation of the per-packet hash, which must be computed over the contents of the entire packet.[4] However, we now discuss several hashing techniques that can be used to speed this computation up, and estimate its computational overhead by citing recent results on the implementation of fash hash functions.

For speedy packet hashing, we suggest first hashing packets using ε_g-almost universal hash function to obtain a short n_1 bit string, and then applying a fast PRF.

ε_g-almost universal hash function is a keyed hash function $g : \{0,1\}^{|k_u|} \times \{0,1\}^* \to \{0,1\}^n$ that maps variable-length inputs to n-bit outputs, and guarantees that for any pair of distinct inputs x, x' that

$$\Pr\left[g_{k_u}(x) = g_{k_u}(x')\right] \leq \varepsilon \tag{14}$$

where the probability is over the choice of k_u keying g.

There are many practical realizations of ε_g-almost universal hash functions that are significantly faster than PRFs and 4-wise independent hashes [12]. The value of ε_g depend on the parameters of the hash; Bernstein has a nice survey of these functions and their parameters [12].

We prove that we can reduce the complexity of packet-hash computation by first hashing each packet using a fast ε_g-almost universal hash function (converting a packet of length ℓ bits to a string of length n_1 bits), and then hashing the resulting n_1-bit string using a PRF or 4-wise independent hash (to obtain the pair of values (i, b) for $i \in [N]$ and $b \in \{-1, 1\}$ used for sketching). Appendix B proves that it suffices to take:

$$\varepsilon_g < \frac{\delta}{10^3 T}\frac{\beta - \alpha}{\alpha + \beta} \tag{15}$$

We consider using GHASH [35] as our ε_g-almost 2-wise independent hash function. Suppose GHASH produces outputs of length n_1, and each packet is at most 1500B long, and block lengths are m, where m is taken as a power of two. Since $\varepsilon_g = \frac{1500\cdot 8}{m}2^{-n_1}$, for $T = 10^7$ packets/interval, and $\delta = \beta = 2\alpha = 1\%$, applying (15) we find that it suffices to choose GHASH with $n_1 = m = 64$ bits. Smaller block sizes lead to faster implementations, and this choice of m and n_1 is quite short! (For typical applications, GHASH uses block lengths of $m = 128$ bits [35].) Moreover, because n_1 is shorter than standard hashing block lengths (*i.e.,* 128, 256, or 512 bits), we only requires a single invocation of a fixed-input-length PRF (or 4-wise independent hash function) to hash the final n_1-bit GHASH output.

[3]We take N to be a power of two because this makes packet hashing more convenient. That is, if $N = 2^q$ and we use PRF that produces q pseudorandom bits, then the binary representation of these q bits is a uniformly chosen element of $[N]$; if N is not a power of 2, a more complicated mapping is required to uniformly choose an element of $[N]$, so we avoid this.

[4]Hashing just the packet header won't work, because then the protocol could not detect an adversary that tampers with the payload of the packet but keeps the header intact.

$\boxed{R_A \text{ (Alice)}} \rightleftarrows \boxed{R_1} \rightleftarrows \boxed{R_2} \rightleftarrows \ldots \rightleftarrows \boxed{R_K} \rightleftarrows \boxed{R_B \text{ (Bob)}}$

Fig. 3. A path from Alice to Bob via K intermediate nodes.

Our next task is to determine how the n_1-bit string be hashed. Is it faster to use a PRF or a 4-wise independent hash? Our answer is very counterintuitive — we find that a PRF is actually faster in practice. This is quite surprising, given that a PRF provides a strictly stronger theoretical guarantees that a 4-wise independent hash function (since every PRF is indistinguishable from a 4-wise independent hash function). To explain this, we observe that 4-wise independent hash functions are typically based on constructions with rigorous proofs of correctness (*e.g.,* polynomials of degree 3 [14]). Meanwhile, practical PRFs come with only *heuristic* guarantees (*e.g.,* GHASH-AES [35] is a PRF under the heuristic assumption that AES is a fixed-input-length PRF). Moreover, significantly more research effort has been expended on developing fast implementations of PRFs in software and hardware.

Sample implementation numbers. Since the task of evaluating the performance of a hash function on the different platforms that be used in high-speed Internet routers (software, FPGA, ASICS) is a entire research project in itself, here we only mention some performance numbers from other published work. We follow the literature [35] by assuming that AES is a fixed-input length PRF. In 2007, Krovetz and Dai found that VMAC instantiated with AES allows for hashing in software at a speeds of between 1-10 cycles per message byte [32]. In 2004, the designers of GHASH found that it could process 127 bits per clock cycle in hardware and 2666 bits per clock cycle in software, assuming 1500 byte packets and 128-bit block lengths [35]. There have been various efforts to improve the performance of GHASH. In 2014, for example, a new FGPA implementation of GHASH claims a throughput of 43.32Gbps [10]. (Note that [10] used 128-bit blocks lengths; recall that we can use even short block lengths of 64-bits.)

Before we conclude, we note that the security of our PRF need not be especially high. Our PQM adversary is presented with an online problem: Eve must break the security of the PRF only during the short time interval before the ephemeral interval key k_u is refreshed. Thus, while a full-fledged block-cipher (*e.g.,* AES) could be used to realize the PRF, performance could further be improved by the replacing AES with a weaker block cipher like DES, or by reducing the number of rounds of AES [18]; an adversary would still require enormous resources to break the PRF's security within the short time limit (≈100ms) imposed by our online setting.

IV. Threat model for Failure localization (FL)

We now move on to showing how secure sketch PQM can be composed to create a secure failure localization protocol. In this section, we present a security definition for fault localization (Section IV-A), and discuss its properties (Section IV-B). We then present our FL protocol in Section V.

A. Security definition for failure localization (FL)

In a failure localization (FL) protocol, Alice learns if the packets she sent to a trusted destination Bob arrived correctly; if they did not, Alice learns at least one link along the path where the failures occurred.

We work in a model where Alice knows the identities of all the nodes on the path to Bob, and all traffic travels on symmetric paths as in Figure 3 (*i.e.,* intermediate nodes $R_1, ..., R_K$, have bi-directional communication links with their neighbors, and messages that sender Alice sends to receiver Bob traverse the same path as the messages that Bob sends back to Alice). We let K be the number of nodes on the path between Alice and Bob. We say that messages traveling towards Alice are going *upstream*, and messages traveling towards Bob are going *downstream*. As in Section II-A, we fix a set of T consecutive packets sent by Alice to be an *interval*, and define a *failure* to be any instance where a packet that was sent by Alice during the interval fails to arrive unmodified at Bob (before the last packet of interval arrives at Bob).

We suppose that an adversary Eve can occupy any subset of nodes $R_1, ..., R_K$ on the network path between Alice and Bob; Eve will remain at those nodes for the duration of the interval. Eve can add, drop, or modify messages sent on the links adjacent to any of the nodes she controls. She can also use timing information to attack the protocol. Because packet loss occurs naturally in the network layer, even in the absence of adversarial behavior, FL protocols should also be able to deal with packet failures that do *not* results from Eve's actions. Therefore, we model congestion by supposing that each link can randomly and independently drop each packet transmitted over that link with probability $\rho > 0$.

A failure localization (FL) protocol allows Alice to detect (1) whether the number of failures during the interval exceeds a certain fraction of total packets transmitted, and (2) if it does, Alice can localize the failures to a link on the path. At the end of each interval of an FL protocol, Alice either (a) decides not to alarm and outputs $\surd$, or (b) or raises an alarms and outputs a link ℓ to which she localized the failures.

Definition IV.1 ((α, β, δ)-security for FL). *Given parameters $0 < \alpha < \beta < 1$ and $0 < \delta < 1$, an FL protocol is (α, β, δ)-secure if, letting T be the number of packets sent during the interval:*

1) (Secure localization). *Consider the* malicious case, *where the adversary Eve can drop, modify, delay, or add packets. We require that if more that βT failures occur, then Alice raises alarm for a link ℓ that is adjacent to a node occupied by Eve, or a link ℓ whose failure rate exceeds $\frac{\alpha}{K+1}$, with probability $1 - \delta$.*
2) (Few false positives). *In the* benign case, *where no intermediate node behaves adversarially (*i.e., *no packets are added or modified on the path, but packets may be reordered or dropped due to congestion) and the failure rate on each link is below the (per-link) false alarm threshold $\frac{\alpha}{K+1}$, then the probability that Alice outputs $\surd$ is at least $1 - \delta$.*

As in Section II-A, we assume that the T packets sent during an interval are distinct.

B. Properties of our FL security definition

We discuss a few properties of our security definition.

Localizing links, not nodes. It is well known (see *e.g.,* [24]) that an FL protocol can only pinpoint a *link* where a failure occurred, rather than the *node* responsible for the failure.

Benign and malicious failures. Our definition requires Alice to accurately localize failures, but these failures may be caused by Eve, or may be the result of *benign causes*, such as congestion. We do not require Alice to distinguish between benign or malicious (*i.e.,* due to Eve) failures, because Eve can always drop packets in a way that "looks like" congestion.

Modeling congestion. An important feature of our definition is that it accounts for messages that are dropped naturally for benign reasons like congestion. If we had not included this in our model, then we would end up with a model where Eve causes *every* packet delivery. This is problematic because such a definition would admit protocols that work to only localize a *single* packet failure (rather than the overall packet loss rate) since this link would necessarily be adjacent to Eve. Indeed, previous work has fallen into this trap. For example, [9] implements FL using a binary search through the path, where a single step of the binary search is initiated each time a packet is dropped; this search can be confounded by natural congestion-related packet loss that causes the binary search algorithm to search for Eve in the wrong part of the path.

Movements of the adversary. Our model does not allow Eve to move from node to node in a single interval. This assumption does not significantly limit the practicality of our protocols for a number of reasons. Firstly, when Eve models a Internet service provider (ISP) that tries to bias the results of FL protocol for business reasons, it is reasonable to assume that she may only occupy nodes owned by her ISP. Furthermore, when Eve is an external attacker or malware that compromises a router, "leaving" a router means that the legitimate owner of the router removed the attacker from the router, *e.g.,* by refreshing its keys. We model key refresh as a re-start of the security game. Furthermore, "movements" to a new router happen infrequently, since an external attacker is likely to need a different strategy each time it compromises a router owned by a different business entity.

V. From Secure Sketch PQM to FL

We show how to compose the secure sketch PQM protocol of Section III-B to obtain a efficient FL protocol with about $O(K^2 \log T + n)$ storage overhead at each node and only two control messages. Our composition of secure sketch PQM to statistical FL will have Alice run K simultaneous secure sketch PQM protocols with each of the intermediate nodes in Figure 3, and use the statistics from each protocol to infer behavior at each link. We first describe the protocol (Section V-A) and then prove its security (Section V-B).

A. Description of the secure sketch FL protocol

As usual, the protocol works in intervals. Every node R_i shares pairwise symmetric master keys k_i, k_i' with Alice. In interval u, the protocol proceeds as follows:

(Sketch.) Using k_i', each intermediate node runs a secure sketch PQM protocol with Alice, so that Alice will keep a sketch $\mathbf{w}_i^A$ for every $i \in [K]$ and every other node R_i will keep a single sketch $\mathbf{w}_i$. Sketching is accomplished as described in the (Sketch) phase of the PQM protocol of Section III-B.

(Interval End.) At the end of interval u, Alice sends a single control 'Interval End' control message formatted as an *onion report*. Using notation $[m]_k$ to denote message m authenticate by a MAC with k, the 'Interval End' message contains a series of nested MAC'd messages as follows:

$$q = [(u,1,\mathbf{w}_1^A)[(u,2,\mathbf{w}_2^A)...[(u,B,\mathbf{w}_B^A)]_{k_B}...]_{k_2}]_{k_1}$$

The 'Interval End' control message is sent upstream along the path from Alice to Bob. Upon receiving a validly-MAC'd 'Interval End' message, intermediate node R_i (a) extracts the sketch $\mathbf{w}_i^A$, (b) strips off his portion of the message $(u, i, \mathbf{w}_i^A)$ and its associated MAC, (c) passes what remains to R_{i+1}, and (d) finally initializes a local timer. R_i must drop the 'Interval End' message if the MAC is invalid.

(Report.) Bob initiates reporting, by forming an 'Onion Report' message $\theta_B = [u, B, V_B]_{k_B}$ and sending it downstream. Upon receipt of the 'Onion Report', each node R_i appends his own information as $\theta_i = [u, i, V_i, \theta_{i+1}]_{k_i}$ and passes θ_i downstream to R_{i-1} until it eventually reaches Alice. Here, V_i is node R_i's *estimator* computed as

$$V_i = \|\mathbf{w}_i^A - \mathbf{w}_i\|_2^2$$

If a node R_i's local timer expires before it receives θ_i from its upstream neighbor R_{i+1}, then R_i constructs his own Report θ_i as above, but setting $\theta_{i+1} = \perp$ to indicate his upstream neighbor failed to send his report.

(Security Check.) Letting α, β be the false alarm and detection thresholds, when Alice receives the final onion report θ_1, she computes

$$F_\ell = V_i - V_{i+1}$$

for each link $\ell = (i, i+1)$, and outputs ℓ if $\ell = (i, i+1)$ is the upstream-most link where

$$F_\ell > \tfrac{T}{K+1} \tfrac{\beta(2\alpha+\beta)}{\alpha+2\beta} = \Gamma$$

or the onion report θ_{i+1} refers to the wrong interval, is missing, or is invalidly MAC'd. If there is no such link, she outputs $\surd$.

B. Security analysis for secure sketch FL

To prove that this scheme is secure, we need to assume that interval length T is long enough, the sketches are big enough, and the congestion rate ρ is small enough. Our proof also relies on limiting the number of links occupied by Eve to $O(\sqrt{K})$.

Theorem V.1. *The composition of secure sketch PQM described above satisfies (α, β, δ)-statistical security if the congestion rate satisfies $\rho K^2 \leq \beta$, Eve occupies*

$$M \leq \sqrt{(K+1)(1 - \tfrac{\rho}{\beta}K^2)} \tag{16}$$

links, each interval contains at least $T > \frac{K+1}{\alpha}$ packets, and for each $i \in [K]$, sketches $\mathbf{w}_i, \mathbf{w}_i^A$ have

$$N_i = \tfrac{32}{\delta}\left(i(K+1)\tfrac{2\beta+\alpha}{\beta-\alpha}\right)^2 \tag{17}$$

counters, each of size $1 + \frac{1}{2}\log_2\left(4\frac{T}{N}\ln(\frac{200N}{\delta})\right)$ *bits.*

We remark that, for a given interval of length T, this protocol requires $O(K^2 \log T)$ storage overhead at Bob and each intermediate node, while the storage overhead at Alice is $O(K^3 \log T)$. The communication overhead of the protocol is two control messages of length $O(K^3 \log T)$ each for every T packets sent.

Proof. First, the probability that any efficient adversary Eve successfully forges the interval end message or onion report of an honest node (by forging the MAC) is negligible. We argue that Eve does not tamper with the control messages:

Claim V.2. *If Eve tampers with the 'Interval End' message or any θ_i in the 'Onion Report' message, then Alice will localize a node adjacent to Eve.*

Proof. Let R_E be the upstream-most node where Eve tampered with either the 'Interval End' message or the 'Onion Report'. Let R_j be the first honest node that is downstream of node R_E (we know such a node exists because Eve cannot occupy Bob's node). Since all the $R_1, ..., R_{E-1}$ behave honestly, their all reports $\theta_1, ..., \theta_{E-1}$ will be present and valid. Also, conditioned on Eve not forging R_j's MAC, θ_j will either be invalid (*e.g.,* if Eve tampered with some θ_ℓ for $\ell > j$, since θ_ℓ is nested inside θ_j) or missing (*e.g.,* if Eve dropped the 'Interval End' message). It follows that the upstream-most invalid report θ_x occurs on some link between R_{E-1} and R_j, so that Alice will output a link adjacent to Eve. □

We may now suppose that Alice receives correct reports from all honest nodes. We next present some notation.

Notation. Let D_i be a count of the number of failures that occurred on the path between Alice and R_i. Let $\mathbf{v}_A$ be the characteristic vector of the stream of packets that Alice sends and let $\mathbf{v}_i$ for $i \in [K+1]$ be the characteristic vector of the stream of data packets that R_i receives. Let $\mathbf{x}_i = \mathbf{v}_A - \mathbf{v}_i$. As in equation (3), we can decompose $\mathbf{x}_i$ into two vectors $\mathbf{x}_i = \mathbf{d}_i + \mathbf{a}_i$, where $\mathbf{d}_i$ is the characteristic vector of packets *dropped* on the path from Alice to R_i, and contains the non-negative components of $\mathbf{x}_i$. The vector $\mathbf{a}$ is the characteristic vector of packets *added* on the path from Alice to R_i, and contains the non-positive components of $\mathbf{x}_i$.

The following lemma, proved in Appendix C of [25], proves the "few false positives" and "secure localization" conditions of Definition IV.1:

Lemma V.3. *Let* $\Gamma = \frac{T}{K+1}\frac{\beta(2\alpha+\beta)}{\alpha+2\beta}$ *and* $\varepsilon_i = \frac{1}{2i}\frac{\beta-\alpha}{2\beta+\alpha}$*. For every* $i \in [K]$*, assume that* R_i *computes an estimate* V_i *that* (ε_i, δ')*-estimates* $\|\mathbf{x}_i\|_2^2$*. Suppose also that* $\|\mathbf{x}_i\|_2^2 \leq \frac{\beta i}{K+1}$*. Then with probability at least* $1 - 2\delta'$ *it follows that:*

1) *If "link* $(i, i+1)$ *is good" so that* $\|\mathbf{x}_{i+1}\|_2^2 - \|\mathbf{x}_i\|_2^2 \leq \frac{\alpha}{K+1}T$ *then* $V_{i+1} - V_i \leq \Gamma$.
2) *If "link* $(i, i+1)$ *is bad" so that* $\|\mathbf{x}_{i+1}\|_2^2 - \|\mathbf{x}_i\|_2^2 \geq \frac{\beta}{K+1}T$ *then* $V_{i+1} - V_i \geq \Gamma$.

Few false positives: To prove this, we consider an interval where all the nodes on the path behave honestly. During this interval, we know that no packets were added anywhere on the path (so that $\|\mathbf{a}_i\|_2^2 = 0$ for each $i \in [K+1]$) and less than $\frac{\alpha}{K+1}$ packets were dropped at each link. We can apply equation (3) to find that for each link $(i, i+1)$ we have

$$\|\mathbf{x}_{i+1}\|_2^2 - \|\mathbf{x}_i\|_2^2 = (D_{i+1} + 0) - (D_i + 0) \leq \frac{\alpha}{K+1} \quad (18)$$

and the telescoping nature of (18) gives us that

$$\|\mathbf{x}_i\|_2^2 = (\|\mathbf{x}_i\|_2^2 - \|\mathbf{x}_{i-1}\|_2^2) + ... + (\|\mathbf{x}_2\|_2^2 - \|\mathbf{x}_1\|_2^2) + \|\mathbf{x}_1\|_2^2 \leq \frac{\alpha i}{K+1} \quad (19)$$

We can now apply Lemma V.3 to show that, with probability at least $1 - 2\delta'$ we have that $V_{i+1} - V_i \leq \Gamma$ so that Alice will not output link $(i, i+1)$. A union bound over the $K+1$ links gives us that Alice will output $\surd$ during this interval with probability at least $1 - 2(K+1)\delta'$.

Secure localization: We now show that if Eve causes more than a β fraction of failures in the interval, then with probability at least $1 - \delta$, Alice will either catch Eve or output a link with more than $\frac{\alpha}{K+1}$ failures. Recall that Alice outputs the upstream-most link $\ell = (i, i+1)$ for which there is an "alarm", *i.e.,* where $V_{i+1} - V_i \geq \Gamma$. We need the following simple observation:

Lemma V.4. *Define event* E_i *as the event that* $\|\mathbf{x}_i\|_2^2 \leq \frac{\beta i}{K+1}$ *.For each* $i \in [K+1]$*, if Alice does not raise an alarm for any link upstream of link i, then* E_i *holds with probability* $1 - 2i\delta'$*.*

Proof. Suppose that Alice does *not* raise an alarm for all links upstream of node R_i. Lemma V.3 implies that $\|\mathbf{x}_{j+1}\|_2^2 - \|\mathbf{x}_j\|_2^2 \leq \frac{\beta}{K+1}$ with probability $1 - 2\delta'$, for each link $(j, j+1)$ where $j \in [i-1]$. The lemma follows from a union bound over these links and a telescoping sum as in (19). □

First we show that the with high probability Alice will not output an honest link. Let link $(i, i+1)$ be "honest", *i.e.,* have a fewer than $\frac{\alpha}{K+1}$ failures, and assume that Alice does not raise alarm for any links upstream of R_i. Now, Lemma V.3 shows that, conditioned on E_i, Alice will not raise an alarm for link $(i, i+1)$ with probability at least $1 - 2\delta'$. Since Alice does not alarm for any links upstream of R_i, we can apply Lemma V.4 to remove the conditioning on E_i. It follows that Alice will not output honest link $(i, i+1)$ with probability at least $1 - 2(i+1)\delta'$. Taking a union bound over all honest links gives that Alice will not alarm for any honest link with probability at least $1 - 2(K+1)^2\delta'$. Next, we need to show that Alice either will raise an alarm for a link adjacent to Eve or link with more than $\frac{\alpha}{K+1}$ failures. The proof hinges on the following technical lemma, proved in Appendix D of our technical report [25]:

Lemma V.5. *If Eve occupies* $M \leq \sqrt{(K+1)(1 - \frac{\rho}{\beta}K^2)}$ *links and causes a* β*-fraction of failures in the interval, then there must be a link* $(i, i+1)$ *that is adjacent to Eve with*

$$\|\mathbf{x}_{i+1}\|_2^2 - \|\mathbf{x}_i\|_2^2 \geq \frac{\beta}{K+1}T \quad (20)$$

Now let link $(i, i+1)$ be the upstream-most link that is adjacent to Eve such that (20) holds. (Lemma V.5 guarantees the existence of such a link.) We have two cases:

1. Suppose Alice did not raise an alarm for a link upstream of R_i. Combining Lemma V.4 and Lemma V.3 it follows that Alice will alarm for link $(i, i+1)$ adjacent to Eve with probability $1 - 2(i+1)\delta'$.

2. Suppose Alice did raise an alarm for a link upstream of R_i. It follows from Lemma V.3 that there is some link $(j, j+1)$ for $j \leq [i-1]$ where, with probability $1 - 2\delta'$,

$$\tfrac{\alpha}{K+1} \leq \|\mathbf{x}_{j+1}\|_2^2 - \|\mathbf{x}_j\|_2^2 = D_{j+1} - D_j + \|\mathbf{a}_{j+1}\|_2^2 - \|\mathbf{a}_j\|_2^2$$

where the equality comes from applying equation (3). Now if link $(j, j+1)$ is adjacent to Eve, it follows that Alice alarms for a link adjacent to Eve, and we are done. Thus, suppose that link $(j, j+1)$ is *not* adjacent to Eve. Then, it follows that no new packets could have been added to this link, and so we have that $\|\mathbf{a}_{j+1}\|_2^2 = \|\mathbf{a}_j\|_2^2$. Thus, if link $(j, j+1)$ is *not* adjacent to Eve, then Alice must have raised an alarm for a link with $D_{j+1} - D_j \geq \frac{\alpha}{K+1}$ failures, as required.

Combining these cases, we see that with probability at least $1 - 2(K+1)\delta'$, Alice will either raise an alarm for a link that is either (a) adjacent to Eve, or (b) has more than $\frac{\alpha}{K+1}$ failures, as required.

Sizing the sketches. To ensure that (α, β, δ)-statistical security holds, we take $\delta' = \delta/4(K+1)^2$. Next, recall that Lemma V.3 requires sketches that (ε_i, δ')-estimate the p^{th} moment with $\varepsilon_i = \frac{1}{2i}\frac{\beta-\alpha}{2\beta+\alpha}$. For simplicity, we now suppose that the sketches are constructed using 4-wise independent hashing functions, so we plug ε_i, δ in Theorem III.1 to find that for $i \in [K+1]$ it suffices to take sketches $\mathbf{w}_i$, $\mathbf{w}_i^A$ of with $N_i > \frac{2}{\varepsilon_i^2 \delta}$, where the number of bits per counter is as in (6). Substituting in the values for ε_i, δ' gives us (17) as required. $\square$

A note on Eve's strategy. While Theorem V.1 guarantees that our protocol accommodates all possible strategies by Eve (as long as they satisfy the conditions in the theorem), the curious reader might wonder what is strategy is best from Eve's perspective. Should she drop all βT packets on just one link, or spread her βT packet drops over multiple links? It turns out the latter case is better for Eve. This follows because the estimator $V_{i-1} - V_i$ is proportional to the number of packets dropped/modified on link $(i-1, i)$; thus, if fewer packets are dropped on each link, the estimator is smaller and therefore closer to Γ, which makes it more likely that Alice will not alarm (and thus fail to catch Eve).

C. Sample parameters and experiments

We now use determine the size of our sketch for the following sample parameters: There are $K = 4$ nodes between Alice and Bob (for a total of 5 links). We suppose the detection threshold is $\beta = 0.01$, the false alarm threshold is $\alpha = \beta/2$ and about $T = 10^6$ packets are sent during an interval. We will require a confidence of $1 - \delta = 99\%$. Applying Theorem V.1, we find that the i^{th} node needs a sketch with $N_i = 2 \times 10^6 i^2$ counters and $5 - \log_2 i$ bits per counter. This sketch is quite large, but recall that Theorem V.1 assumes packet hashing uses a 4-wise independent hash function. Per Theorem III.2, however, we already know that sketches can be smaller when a PRF is used for packet hashing. Indeed, our numerical experiments confirm this. We find that it suffices for node i to

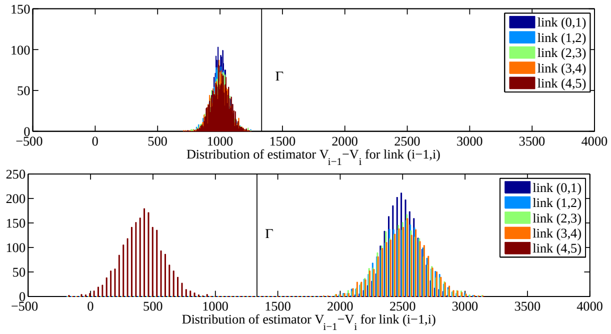

Fig. 4. Distribution of estimator $V_{i-1} - V_i$ for each link $(i-1, i)$ using packet-hashing with a PRF and with $N_i = 800i^2$, $T = 10^6$, $\delta = \beta = 2\alpha = 1\%$ and threshold $\Gamma = 1333$, computed via numerical experiments. $K = 4$ nodes. (Top) The benign case, where each link drops exactly $\alpha/(K+1)$ packets. (Bottom) The malicious case. There is a congestion of rate $\rho = \beta/K^2$ randomly dropping packets on each link, and Eve drops $\beta/K - \rho$ packets on every link except the (4,5) link.

use a sketch with $800i^2$ counters, so that the i^{th} node requires a sketch of size $i^2(10 - \log_2 i) \times 100$B.

Numerical experiments. We preformed numerical experiments for the case where keyed-packet hashing function is instantiated with PRF. We assumed that and each node uses a sketch of size $N_i = 800i^2$. Our numerical experiments operates on synthetic traffic, similar to the experiments in Section III-E; this time, however, we need to simulate packet dropping on each individual link, not just on the entire path. To do this, we model every distinct packet sent during the interval *on a given link* $(i-1, i)$ with fresh pair of uniformly-independent random numbers (j, b) where $j \in [N_i]$ and $b \in \{-1, 1\}$, and use the same technique used in Section III-E to compute the estimator $V_{i-1} - V_i$ on link $(i-1, i)$. The distributions of the resulting estimators are shown in Figure 4, along with the alarm threshold Γ. We present two cases.

Benign case (Figure 4 (Top)). We suppose each link drops exactly $\alpha/(K+1)$ packets. As required, the distribution of the estimator as the $V_{i-1} - V_i$ for each link $(i-1, i)$ is below the threshold Γ, and Alice will not raise an alarm.

Malicious case (Figure 4 (Bottom)). We suppose that there is a congestion of $\rho = \beta/K^2$ randomly dropping packets on each link (per Theorem V.1), and that Eve drops $\beta/K - \rho$ packets on every link except the (4,5) link. As required, the distribution of the estimators for all links except (4,5) are above the threshold Γ, while the estimator for the (4,5) link is well below the threshold Γ. Thus, Alice will correctly raise an alarm and localize all links except the (4,5) links.

D. Implementation issues.

We discuss a few limitations of our secure sketch FL protocol, beyond those related to the computational overhead of packet hashing per Section III-F. First, the protocol requires all traffic being monitored to flow along the same symmetric path. Load balancers, that could split traffic across different paths, complicate its operation; to deal with this, nodes would need to ensure that each sets of flows that take the same path are monitored by a single instance secure sketch FL protocol. Another limitation is that secure sketch FL requires symmetric keys between all nodes on the path and Alice. We do not deny that this is a heavy overhead, but we do note that such keys are *necessary* for any FL protocol that satisfies our security definition (per our lower bounds in [11]), and note that these keys could be established on the fly using traditional key-exchange schemes [19]. Other works in the FL space have also had to grapple with these two important limitations; see [38] for more ideas on how to address them.

VI. Related work

The literature on path-quality monitoring typically deals only with the benign setting; most approaches either have the destination return a count of the number packets he receives from the source, or are based on active probing (ping, traceroute, [42]–[44] and others). However, both approaches fail to satisfy our security definition. The counter approach is vulnerable to attack by an adversary who hides packet loss by adding new, nonsense packets to the data path. Active probing fails when an adversary preferentially treats probe packets while degrading performance for regular traffic, or when an adversary sends forged reports or acknowledgments to mask packet loss. Even known passive measurement techniques, where normal data packets are marked as probes, either explicitly as in IPPM [44] or implicitly as in Trajectory Sampling [21] and PSAMP [16], are vulnerable to the same attacks as active probing techniques if the adversary can distinguish the probe packets from the non-probe packets (*e.g.,* see [23]).

To obtain path-quality monitoring protocols that work in the adversarial setting, we have developed protocols that are more closely related to those used for traffic characterization (see [50] for a survey). On one hand, our protocols are less computationally efficient because they necessarily require the use of keys and cryptographic hash functions to prevent adversaries from biasing measurement results (see [24] for the proof). On the other hand, we used the special structure of PQM to prove new analytical bounds, resulting in provably lower communication and storage requirements than those typically needed in traffic characterization applications (Theorem III.2).

In concurrent work, [36] considered a setting which Alice and Bob are required to sketch adversarially-chosen sets, and then compute metrics on their sets after exchanging sketches over a secure channel; their model maps directly to our PQM model, where Alice and Bob's sets (*i.e.,* packet streams) may be chosen adversarially, and then sketches are exchanged via an authenticated channel. Our work deals with the fact that streams are chosen adversarially by requiring Alice and Bob to compute their sketches using shared secret keys. However, [36] require that sketching is performed *without* any shared randomness.[5] The main advantage of their approach is the reduced key-management overhead. However, this comes at a significant cost; [36] show that any moment estimation protocol for sets of size T requires at least $\Omega(\sqrt{T})$ storage at Alice and Bob. Thus, these protocols are less efficient than our $O(\log T)$-storage *keyed* sketches.

[5] In [24], we argued that PQM protocols require shared randomness; the existence of [36]'s protocol does not contradict this. If we used [36]'s sketching results in a PQM protocol, we would require shared randomness to cryptographically authenticate the report messages (containing the sketches) sent from Bob to Alice.

Our results are also related to work in the cryptography and security literature. Early work in this space, *e.g.,* [20], [41], focused on providing availability guarantees on a *per-packet* basis, resulting in schemes with very high overhead. Later proposals for secure PQM designed to detect when the packet-loss *rate* becomes too high [8], [37], [46], but ours was the first work in this area to provide a formal security model and to prove the security of our protocols in this model. Indeed, one of [37]'s PQM approaches is based on a simple counter (and is therefore vulnerable to the attack described above), while Listen [46] does not use cryptographic operations, and is thus vulnerable to attack by an intermediate node that injects false acknowledgments onto the path. Another earlier work, Stealth Probing [8], is secure in our model but incurs the extra overhead of encrypting all traffic.

Prior to the publication of our work [11], there were a number of proposals for secure FL [6], [7], [9], [37], [40], [49]. The protocol presented here, however, is the only one to exploit the storage and communication savings created by sublinear sketching algorithms; moreover, [6], [9], [40] had a number security flaws that we discussed in detail in [11].

Subsequent to the publication of our work [11], [51] considered FL protocols that are more closely related to the sampling-based FL protocols that we presented in [11] (but omitted from this paper), focusing on optimizing the tradeoffs between communication/storage overhead and the protocol's detection rate (*i.e.,* the number of packets in an interval). In another interesting work, [52] use a small trusted computing base and remote code attestation to circumvent our lower bounds from [11], that argued that in any secure FL protocol, the intermediate nodes and Alice and Bob must all share some secret information. Also subsequently to our work, [4], [13], [53] considered FL in a more stringent setting of multiple paths, similar to the SMT framework; in these protocols, a sender must not only localize a faulty node that is dropping packets, but must also find a path through the network that is guaranteed to deliver its packets. Finally, ICING [38] is a cryptographic network primitive that ensures that packets traverse a known path, selected by the source, in an adversarial setting similar to ours.

VII. Conclusion

We have designed and analyzed an efficient protocol that give accurate estimates of path quality in a challenging environment where adversaries may drop, delay, modify, or inject packets. Our protocols have reasonable overhead, even when compared to previous solutions designed for the *non-adversarial* settings. We combine techniques from sublinear streaming algorithms with simple cryptographic primitives to obtain a protocol that can monitor millions of packets using less than a single kilobyte of storage, and only two small control messages. We have also showed how to compose multiple instances of our PQM protocols to localize the adversary to particular links on a data path. We believe that our secure sketch protocols, and our associated models of their properties, are valuable building blocks for the design of future networks with predictable security and performance guarantees.

Acknowledgments

We thank Eugene Brevdo, Moses Charikar, Nick Feamster, Piotr Indyk, Changhoon Kim, Amir Shpilka, and Yi Wang for useful discussions, and Shai Halevi, Elliott Karpilovsky, Haakon Ringberg, Augustin Soule and the anonymous SIGMETRICS'08 and EUROCRYPT'08 reviewers for comments that have greatly improved the presentation of this work. We are also grateful to UCLA's Institute for Pure and Applied Mathematics's for hosting a semester on Securing Cyberspace in 2006, where much of this work was begun, and to Fabio Manio, Flavio Bonomi, David McGrew, Syam Appala and the rest of Cisco's Advanced Architecture and Research Organization for assistance with the practical aspects of the secure sketch protocol.

Much of this work was performed while S.G., D.X, and B.B were at Princeton University, and E.T. was at MIT. S.G. and J.R were supported by HSARPA grant 1756303. S.G. was supported by NSF grant CNS-0627526. D.X. was supported by an NDSEG Graduate Fellowship and an NSF Graduate Research Fellowship. E.T. was supported by Rothschild Fellowship. B.B. was supported by NSF grants CNS-0627526 and CCF-0426582, US-Israel BSF grant 2004288 and Packard and Sloan fellowships.

References

[1] D. Achlioptas. Database-friendly random projections. In *PODS*, pages 274–281, 2001.

[2] R. Ahlswede and A. Winter. Strong converse for identification via quantum channels. *IEEE Trans. IT*, 48(3):569–579, 2002.

[3] N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. In *STOC*, pages 20–29, 1996.

[4] Y. Amir, P. Bunn, and R. Ostrovsky. Authenticated adversarial routing. In *Theory of Cryptography Conference, TCC*, 2009.

[5] D. Angluin and L. G. Valiant. Fast probabilistic algorithms for hamiltonian circuits and matchings. *Journal of Computer and system Sciences*, 18(2):155–193, 1979.

[6] K. Argyraki, P. Maniatis, O. Irzak, A. Subramanian, and S. Shenker. Loss and delay accountability for the Internet. *ICNP*, 2007.

[7] I. Avramopoulos, H. Kobayashi, R. Wang, and A. Krishnamurthy. Highly secure and efficient routing. In *IEEE INFOCOM*, 2004.

[8] I. Avramopoulos and J. Rexford. Stealth probing: Data-plane security for IP routing. *USENIX*, 2006.

[9] B. Awerbuch, D. Holmer, C. Nita-Rotaru, and H. Rubens. An on-demand secure routing protocol resilient to byzantine failures. In *ACM WiSE*, 2002.

[10] Y. Bai, G. Shou, Y. Hu, and Z. Guo. High performance pipelined architecture of ghash. In *Broadband Network and Multimedia Technology (IC-BNMT), 2010 3rd IEEE International Conference on*, pages 716–720. IEEE, 2010.

[11] B. Barak, S. Goldberg, and D. Xiao. Protocols and lower bounds for failure localization in the Internet. In *IACR EUROCRYPT*, 2008.

[12] D. J. Bernstein. Polynomial evaluation and message authentication. Technical report, http://cr.yp.to/ Document ID: b1ef3f2d385a926123e1517392e20f8c., October 2007.

[13] P. Bunn and R. Ostrovsky. Asynchronous throughput-optimal routing in malicious networks. In *ICALP (2)*, pages 236–248, 2010.

[14] J. L. Carter and M. N. Wegman. Universal classes of hash functions. *JCSS*, 18(2):143–154, 1979.

[15] M. Charikar, K. Chen, and M. Farach-Colton. Finding frequent items in data streams. *Theoretical Computer Science*, 312(1):3–15, 2004.

[16] B. Claise. Packet sampling (psamp) protocol specifications. 2009.

[17] J. Cowie. Rensys blog: China's 18-minute mystery, 2010. http://www.renesys.com/blog/2010/11/chinas-18-minute-mystery.shtml.

[18] J. Daemen and V. Rijmen. A new MAC construction ALRED and a specific instance ALPHA-MAC. In *FSE: Fast Software Encryption*, volume 3557, pages 1–17. Springer, 2005.

[19] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.1. RFC 4346 (Proposed Standard), 2006.

[20] D. Dolev, C. Dwork, O. Waarts, and M. Yung. Perfectly secure message transmission. *J. of the ACM*, 40(1), 1993.

[21] N. G. Duffield and M. Grossglauser. Trajectory sampling for direct traffic observation. *IEEE/ACM Trans. Networking*, 9(3), 2001.

[22] S. Gaudin. Interview with a convicted hacker: Robert Moore tells how he broke into routers and stole VoIP services. *InformationWeek*, September 26 2007.

[23] S. Goldberg and J. Rexford. Security vulnerabilities and solutions for packet sampling. *IEEE Sarnoff Symposium*, 2007.

[24] S. Goldberg, D. Xiao, E. Tromer, B. Barak, and J. Rexford. Path quality monitoring in the presence of adversaries. In *SIGMETRICS*, June 2008.

[25] S. Goldberg, D. Xiao, E. Tromer, B. Barak, and J. Rexford. Path-quality monitoring in the presence of adversaries: The secure sketch protocols. Technical report, Boston University, 2013.

[26] O. Goldreich. *Foundations of Cryptography*. Cambridge University Press, 2007.

[27] K. J. Houle and G. M. Weaver. Trends in denial of service attack technology. Technical report, CERT Coordination Center, October 2001.

[28] R. Impagliazzo and M. Luby. One-way functions are essential for complexity based cryptography. pages 230–235, 1989.

[29] R. Impagliazzo and S. Rudich. Limits on the provable consequences of one-way permutations. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 44–61. ACM, 1989.

[30] Keynote. Keynote launches new SLA services, June 2001. http://investor.keynote.com/phoenix.zhtml?c=78522&p=irol-newsArticle_Print&ID=183745.

[31] H. Krawczyk, M. Bellare, and R. Canetti. HMAC : Keyed-Hashing for Message Authentication. RFC 2104, 1997.

[32] T. Krovetz and W. Dai. VMAC, April 2007. CFRG Working Group, Internet Draft.

[33] K. Levchenko. Chernoff bound. www-cse.ucsd.edu/~klevchen/techniques/chernoff.pdf, 2008.

[34] M. Luckie, K. Cho, and B. Owens. Inferring and debugging path MTU discovery failures. In *Internet Measurement Conference*, 2005.

[35] D. A. McGrew and J. Viega. The security and performance of the galois/counter mode (GCM) of operation. In *INDOCRYPT*, pages 343–355. Springer-Verlag, 2004.

[36] I. Mironov, M. Naor, and G. Segev. Sketching in adversarial environments. In *STOC*, 2008.

[37] A. T. Mizrak, Y.-C. Cheng, K. Marzullo, and S. Savage. Detecting and isolating malicious routers. *Dependable and Secure Computing, IEEE Transactions on*, 3(3):230–244, 2006.

[38] J. Naous, M. Walfish, A. Nicolosi, D. Mazieres, M. Miller, and A. Seehra. Verifying and enforcing network paths with icing. In *CoNEXT*, page 30. ACM, 2011.

[39] A. Nucci. Skype detection: Traffic classification in the dark, 2006. http://www.narus.com/_pdf/news/Converge-Skype-Detection.pdf.

[40] V. N. Padmanabhan and D. R. Simon. Secure traceroute to detect faulty or malicious routing. *ACM SIGCOMM Computer Communication Review*, 33(1):77–82, 2003.

[41] R. Perlman. *Network Layer Protocols with Byzantine Robustness*. PhD thesis, MIT, 1988.

[42] J. Sommers, P. Barford, N. Duffield, and A. Ron. Improving accuracy in end-to-end packet loss measurement. In *ACM SIGCOMM*, 2005.

[43] J. Sommers, P. Barford, N. Duffield, and A. Ron. Accurate and efficient SLA compliance monitoring. In *ACM SIGCOMM Computer Communication Review*, volume 37, pages 109–120. ACM, 2007.

[44] E. Stephan. Ip performance metrics (ippm) metrics registry. 2005.

[45] J. Stone and C. Partridge. When the CRC and TCP checksum disagree. In *ACM SIGCOMM*, 2000.

[46] L. Subramanian, V. Roth, I. Stoica, S. Shenker, and R. H. Katz. Listen and Whisper: Security mechanisms for BGP. In *NSDI*, 2004.

[47] M. Thorup and Y. Zhang. Tabulation based 4-universal hashing with applications to second moment estimation. In *SODA*, pages 615–624, 2004.

[48] E. Vos. List of bad isps shaping p2p traffic - muniwireless, 2009. www.muniwireless.com/2009/01/16/is-your-isp-a-bad-isp/.

[49] E. L. Wong, P. Balasubramanian, L. Alvisi, M. G. Gouda, and V. Shmatikov. Truth in advertising: Lightweight verification of route integrity. In *PODC*, 2007.

[50] J. Xu. Tutorial on network data streaming. In *ACM SIGMETRICS*, 2008.

[51] X. Zhang, A. Jain, and A. Perrig. Packet-dropping adversary identification for data plane security. In *CoNEXT: Conference on emerging Networking EXperiments and Technologies*, December 2008.

[52] X. Zhang, Z. Zhou, G. Hasker, A. Perrig, and V. Gligor. Network fault localization with small TCB. In *IEEE International Conference on Network Protocols (ICNP)*, pages 143–154, 2011.

[53] X. Zhang, Z. Zhou, H.-C. Hsiao, T. H.-J. Kim, A. Perrig, and P. Tague. Shortmac: Efficient data-plane fault localization. *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2012.

□ □ □ □

Appendix A
Interval synchronization

In our secure sketch PQM protocol (Section III-B), the 'Interval End' and 'Report' control messages can be used to synchronize the interval number between Alice and Bob, even if the path between them is subject to variable latency. However, even in the benign case, out-of-order packet delivery at the network layer can cause packets in an interval u to arrive after the 'Interval End' message u (and thus be interpreted by Bob as part of interval $u+1$). Note that out-of-order packet delivery could also occur if Eve deliberately delays packets. To avoid false alarms due to more than αT packets arriving out-of-order before the 'Interval End' control message, we can tune parameter α.

To ensure natural packet reordering does not cause a loss of interval synchronization between the sender and receiver, a good rule of thumb is to ensure that $\alpha T \geq 99^{th}$ percentile of packet lag. Define the packet lag as the number of packets that were sent by the sender after a reordered packet, but were received at the receiver earlier than the reordered packet itself (*e.g.,* if a sender sends the stream 1,2,3,4,5,6,7,8 but the received stream is 1,2,4,5,6,7,3,8, then packet 3 is the reordered packet and packet lag is 4). The value of the packet lag depends on the the class of packets monitored by the PQM protocol. If the packets belong to the same network flow, we can safely assume that packet lag is less than 128 packets, because this is the assumption made in IPSec. Thus, it suffices to take $\alpha T > 1280$. In cases when multiple network flows are monitored with the same PQM instance, then packet lag can be very high (due to load balancing, ECMP, *etc.*); however, we conjecture that even if there is a 10ms difference between the "fast path" used by one group of flows and the "slow path" used by another group of flows, for 1 Gbps flow of traffic, packet lag should be on the order of 10^9 bps/64 bytes/packet $\times 0.01$ sec $= 1.6 \times 10^5$ packets, so we can use $\alpha T > 1.6 \times 10^6$.

Appendix B
Fast packet hashing

Section III-F indicated that the computational cost of packet hashing can be reduced by (1) first mapping packets from from U to a short n_1-bit string using an efficient ε_g-almost universal hash function, and (2) then using a PRF or 4-wise independent hash to map from this n_1-bit string to the sketch. We show this is possible via approaches based on [47].

Preliminaries. Return to the notation of Section III-C, and recall that U is the universe of all possible packets, $\mathbf{v}$ is the characteristic vector of the stream of packets, and $\mathbf{w}$ is the sketch vector of length N. Let $g : U \to \{0,1\}^{n_1}$ be an ε_g-almost universal hash function, as defined in Section III-F. The hash function g maps the packet stream containing elements in U to a new 'intermediate' stream where each element is an n_1-bit string. Let $\mathbf{u}$ be an 'intermediate vector' which is the characteristic vector of this new stream of n_1-bit strings.

Our approach amounts to using the ε_g-almost universal hash g to hash $\mathbf{v}$ the 'intermediate vector' $\mathbf{u}$, and then using a a second-moment estimation scheme to hash $\mathbf{u}$ down to the sketch $\mathbf{w}$. Thus, the second-moment estimation scheme estimates the second moment of $\mathbf{u}$, rather than the real characteristic vector $\mathbf{v}$! We now show that, if ε_g is sufficiently small, this does very little damage, since $\|\mathbf{u}\|_2 \approx \|\mathbf{v}\|_2$.

Theorem B.1. *Given a vector* $\mathbf{v} \in 2^{|U|}$ *and* $\mathbf{u} \in \mathbb{R}^{2^{n_1}}$*. Then if* $g : U \to \{0,1\}^n$ *is an* ε_g*-almost 2-wise independent hash function per equation (14), is used to map* $\mathbf{v}$ *to* $\mathbf{u}$ *according to the algorithm* $u_{g(x)} += v_x$ (i.e., $\forall\, x \in \mathbf{v}$ *the* $g(x)^{th}$ *counter in* $\mathbf{u}$ *is incremented with value* v_x*) then*

$$\Pr\left[\left|\ \|\mathbf{u}\|_2 - \|\mathbf{v}\|_2\ \right| > \delta_1 \|\mathbf{v}\|_2\right] < \delta_2 \tag{21}$$

as long as $|\mathbf{v}|_1 > \frac{\delta_1 \delta_2}{\varepsilon_g}$.

To apply this theorem, recall from Section III-C that $|\mathbf{v}|_1 = A + D$. Thus, for (α,β,δ)-secure PQM we would like (21) to hold when $D = \alpha T$ and $D = \beta T$, with $\delta_1 \ll \varepsilon = \frac{\beta-\alpha}{\alpha+\beta}$. We will conservatively take $|\mathbf{v}|_1 = T$, and $\delta_1 = \frac{\varepsilon}{10}$ and set $\delta_2 = \frac{\delta}{100}$. Then (α,β,δ)-secure PQM require the hash function g to have ε_g as in (15) because $\varepsilon_g < \frac{\varepsilon\delta}{10^3 T} = \frac{\delta}{10^3 T}\frac{\beta-\alpha}{\alpha+\beta}$.

Proof of Theorem B.1. Let v_a be the a^{th} entry of characteristic vector $\mathbf{v}$. Now, start with the observation that

$$\begin{aligned}
\|\mathbf{u}\|_2^2 &= \sum_{g(a)=g(b)} v_a v_b \\
&= \sum_a v_a^2 + \sum_{a\neq b, g(a)=g(b)} v_a v_b \\
&= \|\mathbf{v}\|_2^2 + \sum_{a\neq b} v_a v_b Y_{a,b}
\end{aligned} \tag{22}$$

where we define the random variable $Y_{a,b}$ as

$$Y_{a,b} = \begin{cases} 1 & \text{if } g(a) = g(b), a \neq b, \\ 0 & \text{else.} \end{cases}$$

and from (22) we take the expectation over the randomness in g and find that

$$\begin{aligned}
\mathbb{E}[\ |\ \|\mathbf{u}\|_2^2 - \|\mathbf{v}\|_2^2\ |\ ] &\leq \sum_{a,b} |v_a v_b| \mathbb{E}[|Y_{a,b}|] \\
&\leq \sum_{a,b} |v_a v_b| \cdot \varepsilon_g \\
&= (|\mathbf{v}|_1^2 - \|\mathbf{v}\|_2^2) \cdot \varepsilon_g
\end{aligned} \tag{23}$$

where the first inequality follows from (22), the second inequality follows because per equation (14) the collision probability of g is ε_g.

Now, we would like to ensure that $\|\mathbf{u}\|_2$ provides a good estimate of $\|\mathbf{v}\|_2$. That is, we would like to satisfy (21). Using Markov's inequality, we have

$$\begin{aligned}
\Pr\left[\left|\|\mathbf{u}\|_2^2 - \|\mathbf{v}\|_2^2\right| > \delta_1 \|\mathbf{v}\|_2^2\right] &\leq \frac{\mathbb{E}[\ |\ \|\mathbf{u}\|_2^2 - \|\mathbf{v}\|_2^2\ |\ ]}{\delta \|\mathbf{v}^2\|_2} \\
&\leq \frac{(|\mathbf{v}|_1^2 - \|\mathbf{v}\|_2^2)}{\|\mathbf{v}\|_2^2} \frac{\varepsilon_g}{\delta_1} \qquad \text{(From (23))} \\
&\leq |\mathbf{v}|_1 \frac{\varepsilon_g}{\delta_1}
\end{aligned}$$

And rearranging the last inequality we know that (21) holds as long as $|\mathbf{v}|_1 > \frac{\delta_1 \delta_2}{\varepsilon_g}$ which completes the proof. $\square$

Appendix C
Proof of Lemma V.3

From the statement of the lemma, we have that R_i computes an estimate V_i that (ε_i, δ')-estimates $\|\mathbf{x}_i\|_2^2$ for every $i \in [K]$. That is:

$$\Pr\left[\left|V_i - \|\mathbf{x}_i\|_2^2\right| \leq \varepsilon_i \|\mathbf{x}_i\|_2^2\right] < 1 - \delta' \tag{24}$$

We now prove each item separately.

Link $(i, i+1)$ is good. Since V_i $(\varepsilon_i, \delta_i)$-approximates $\|\mathbf{x}_i\|_2^2$, we can apply (24) to find, that with probability $1 - 2\delta'$,

$$\begin{aligned}
V_{i+1} - V_i &\leq (1+\varepsilon_{i+1})\|\mathbf{x}_{i+1}\|_2^2 + (1-\varepsilon_i)\|\mathbf{x}_i\|_2^2 \\
&\leq (1+\varepsilon_{i+1})(\|\mathbf{x}_{i+1}\|_2^2 - \|\mathbf{x}_i\|_2^2) + (\varepsilon_{i+1} + \varepsilon_i)\|\mathbf{x}_i\|_2^2 \\
&\leq (1+\varepsilon_{i+1})\tfrac{\alpha}{K+1}T + (\varepsilon_{i+1} + \varepsilon_i)\tfrac{i\beta}{K+1}T \\
&= \tfrac{\alpha}{K+1}T\left(1 + \varepsilon_{i+1}(1 + \tfrac{\beta}{\alpha}i) + \varepsilon_i i \tfrac{\beta}{\alpha}\right) \\
&\leq \tfrac{\alpha}{K+1}T\left(1 + (i+1)\varepsilon_{i+1}(1 + \tfrac{\beta}{\alpha}) + i\varepsilon_i(1 + \tfrac{\beta}{\alpha})\right) \\
&= \tfrac{T}{K+1}\tfrac{\beta(2\alpha+\beta)}{\alpha+2\beta} = \Gamma
\end{aligned} \tag{25}$$

where we get the required inequality by putting $\varepsilon_i = \frac{1}{2i}\frac{\beta-\alpha}{2\beta+\alpha}$.

Link $(i, i+1)$ is bad. Again, we apply (24) to find, that with probability $1 - 2\delta'$,

$$\begin{aligned}
V_{i+1} - V_i &\geq (1-\varepsilon_{i+1})(\|\mathbf{x}_{i+1}\|_2^2 - \|\mathbf{x}_i\|_2^2) - (\varepsilon_{i+1} + \varepsilon_i)\|\mathbf{x}_i\|_2^2 \\
&\geq (1-\varepsilon_{i+1})\tfrac{\beta}{K+1}T - (\varepsilon_{i+1} + \varepsilon_i)\tfrac{i\beta}{K+1}T \\
&= \tfrac{T}{K+1}\tfrac{\beta(2\alpha+\beta)}{\alpha+2\beta} = \Gamma
\end{aligned} \tag{26}$$

where we again get the required inequality by putting $\varepsilon_i = \frac{1}{2i}\frac{\beta-\alpha}{2\beta+\alpha}$.

Appendix D
Proof of Lemma V.5

Since Eve occupies M links and causes at least a β-fraction failures, it immediately follows that there exists a link $(i, i+1)$ adjacent to Eve where at least $\frac{\beta}{M}$-fraction of failures, *i.e.,* $D_{i+1} - D_i \geq \frac{\beta}{M}$. Now if the following holds

$$\|\mathbf{x}_{i+1}\|_2^2 - \|\mathbf{x}_i\|_2^2 > \tfrac{\beta}{K+1}T \tag{27}$$

we are done, since link $(i, i+1)$ is adjacent to Eve. Thus, suppose (27) do *not* hold. Then, applying identity (3), we have that

$$\begin{aligned}
\tfrac{\beta}{K+1}T &\geq \|\mathbf{x}_{i+1}\|_2^2 - \|\mathbf{x}_i\|_2^2 \\
&= D_{i+1} - D_i + \|\mathbf{a}_{i+1}\|_2^2 - \|\mathbf{a}_i\|_2^2
\end{aligned}$$

rearranging and then using that fact that $D_{i+1} - D_i \geq \frac{\beta}{M}$ we get

$$\|\mathbf{a}_i\|_2^2 \geq= \beta T(\tfrac{1}{M} - \tfrac{1}{K+1}) \tag{28}$$

Next, consider the next link $(j, j+1)$ that is occupied by Eve and is upstream of link $(i, i+1)$. Now again, if the following holds

$$\|\mathbf{x}_{j+1}\|_2^2 - \|\mathbf{x}_j\|_2^2 > \tfrac{\beta}{K+1}T \tag{29}$$

then we are done, since link $(j, j+1)$ is adjacent to Eve. So, we again suppose (29) does *not* hold. Since Eve does not occupy any links between R_{j+1} and R_i, and only congestion-related loss could have occurred on the links between R_{j+1} and R_i. It follows that $\|\mathbf{x}_{j+1}\|_2^2 \geq \|\mathbf{x}_i\|_2^2 + \rho(i-j-1)$. Since (29) does *not* hold, we can apply identity (3) and the fact that $\|\mathbf{x}_{j+1}\|_2^2 \geq \|\mathbf{x}_i\|_2^2 + \rho(i-j-1) \geq \|\mathbf{a}_i\|_2^2 + \rho(i-j-1)$ and the bound on $\|\mathbf{a}_i\|_2^2$ in (28) to get

$$\|\mathbf{x}_j\|_2^2 > \beta T\left(\tfrac{1}{M} - \tfrac{2}{K+1} - \tfrac{\rho}{\beta}(i-j-1)\right)$$

We continue this argument for all $m \leq M-1$ links that are adjacent to Eve and upstream of link $(i, i+1)$. Finally, arriving at the last such link, which we call link $(e, e+1)$, we have

$$\begin{aligned}
\|\mathbf{x}_{e+1}\|_2^2 &> \beta T\left(\tfrac{1}{M} - \tfrac{m}{K+1} - \tfrac{\rho}{\beta}(i-e-1)\right) \\
&> \beta T\left(\tfrac{1}{M} - \tfrac{M-1}{K+1} - \tfrac{\rho}{\beta}K\right)
\end{aligned}$$

where the last inequality follows by putting $m \leq M-1$ and $i - e \leq K$. Now since by definition Eve does not occupy any links downstream of link $(e, e+1)$, we immediately have that $\|\mathbf{x}_e\|_2^2 = 0$. It follows that link $(e, e+1)$ has

$$\|\mathbf{x}_{e+1}\|_2^2 - \|\mathbf{x}_e\|_2^2 > \beta T(\tfrac{1}{M} - \tfrac{M-1}{K+1} - \tfrac{\rho}{\beta}K) > \tfrac{\beta}{K+1}$$

where the last inequality follows because we put $M \leq \sqrt{(K+1)(1-\frac{\rho}{\beta}K^2)}$. This concludes the proof of this lemma, since link $(e, e+1)$ is adjacent to Eve.

APPENDIX E
SKETCHING WITH PRFS

We now prove Theorem III.2. To do this, we first prove Theorem E.1, and then show how to derive Theorem III.2 from Theorem E.1. Recall from Section III-D that $\mathcal{S}$ is the set of $N \times |U|$ matrices where each column contains a single ± 1 entry in one row, and zeros in all other rows.

Theorem E.1. *For any vector $\mathbf{v} \in \mathbb{Z}^U$, choosing the $N \times U$ matrix S uniformly from $\mathcal{S}$ and setting $\mathbf{w} = S\mathbf{v}$, we have that for all $\varepsilon \in [0,1)$ and all $q, r > N$*

1) *If $\mathbf{v} \in \{-1,0,1\}^U$, and $\|v\|_2^2 \le q$, then for $\eta \in [0, \frac{1}{2}\sqrt{\varepsilon^2+10\varepsilon+9} - \frac{1}{2}(\varepsilon+3))$ and $y \doteq \frac{(1+\varepsilon)(1-\eta)}{(1+\eta)^2} - 1$:*

$$\Pr\left[\|\mathbf{w}\|_2^2 > (1+\varepsilon)q\right] \le 2Ne^{-\frac{\eta^2 q}{3N}} + e^{-\frac{N}{2}(y^2/2 - y^3/3)} \tag{30}$$

2) *If the number of* non-zero *entries in $\mathbf{v}$ is r, then for $\eta \in \left(0, \frac{1}{2-\varepsilon}(3 - 2\varepsilon - \sqrt{5\varepsilon^2 - 14\varepsilon + 9})\right)$and $y \doteq \frac{(1-\eta)^2}{1+\eta}(1-\frac{\varepsilon}{2}) - (1-\varepsilon)$ it follows that*

$$\Pr\left[\|\mathbf{w}\|_2^2 < (1-\varepsilon)r\right] \le 2Ne^{-\frac{\eta^2 r}{3N}} + e^{-N\frac{\varepsilon}{3(1+\eta)}y} \tag{31}$$

Proof of Theorem E.1. Our main observation is that, with high probability, the ± 1 entries of $\mathbf{v}$ are distributed evenly among the coordinates of $\mathbf{w}$. Conditioned on this happening, we can then apply the analysis of [1].

Definitions. We need the following definitions.

- We write v_x for the x^{th} element in $\mathbf{v}$.
- Define for $i \in [N]$ the set $Q_i = \{x \in U \mid h(x) = i\}$ where h is the pseudorandom hash function.
- Define D_i as the number of non-zero entries in $\mathbf{v}$ that hash to the i^{th} bin the sketch $\mathbf{w}$. That is $D_i = |\{v_x | v_x \neq 0, x \in Q_i\}|$.
- Define Y_x as an unbiased ± 1 random variable for each $x \in U$.

Our proof proceeds as follows. We first obtain a bound on D_i for each i. (Note: This bound on D_i gives rise to the awkward bound on T in Theorem III.2 of Section III-D.) When then use the bounds on D_i to prove the first item (30), and then use them to prove the second item (31).

Bounding D_i. Let E_i denote the event that $\exists i \in [N]$ such that $D_i > (1+\eta)q/N$ or $D_i < (1-\eta)q/N$. Then, for $\eta \in [0,1)$, we have that

$$\Pr[E_1] \le N\left(\Pr[D_i > (1+\eta)\tfrac{q}{N}] + \Pr[D_i < (1-\eta)\tfrac{q}{N}]\right) \le N\left(e^{-\frac{\eta^2}{3}\frac{q}{N}} + e^{-\frac{\eta^2}{2}\frac{q}{N}}\right) \tag{32}$$

which is a straightforward application of a union bound followed by the Chernoff bound.[6]

Bounding the first item. Now we condition on $\neg E_1$. Let $\gamma = \frac{1+\varepsilon}{(1+\eta)^2}$ and write:

$$\begin{aligned}
\Pr[\|\mathbf{w}\|_2^2 > (1+\varepsilon)q \mid \neg E_1] &= \Pr[\sum_{i=1}^{N} D_i^2 \left(\frac{1}{D_i}\sum_{x\in Q_i} Y_x v_x\right)^2 > (1+\varepsilon)q \mid \neg E_1] \\
&= \Pr[\sum_{i=1}^{N} \left(\frac{1}{D_i}\sum_{x\in Q_i} Y_x v_x\right)^2 > \gamma\frac{N^2}{q} \mid \neg E_1]
\end{aligned}$$

where first equality comes from expanding $\mathbf{w}$ as $\mathcal{S}\mathbf{v}$ and then multiplying by $\frac{D_i}{D_i}$, and the second equality follows from the fact that conditioning on $\neg E_i$ implies that $D_i \le (1+\eta)q/N$. Next, set $\mathbf{Y}_i$ to be the vector of all Y_x for each $v_x \in \{-1,1\}, x \in Q_i$. Set $\mathbf{u}_i$ the vector with entries $\frac{v_x}{\sqrt{D_i}}$ for each $v_x \in \{-1,1\}, x \in Q_i$. Notice that both $\mathbf{Y_i}$ and $\mathbf{u_i}$ have length D_i, and $\|\mathbf{u}_i\|_2^2 = 1$ so $\mathbf{u_i}$ is a unit vector. Now we write

$$\begin{aligned}
&= \Pr[e^{t\sum_{i=1}^{N}\langle\frac{\mathbf{Y}_i}{\sqrt{D_i}}, \mathbf{u}_i\rangle^2} > e^{t\gamma\frac{N^2}{q}} \mid \neg E_1] \\
&\le e^{-t\gamma\frac{N^2}{q}} \prod_{i=1}^{N} \mathbb{E}[e^{t\langle\frac{\mathbf{Y}_i}{\sqrt{D_i}}, \mathbf{u}_i\rangle^2} \mid \neg E_1]
\end{aligned}$$

[6] We use the following Chernoff bounds. Let X_i be i.i.d indicator variables with mean μ, and let

$$\Pr\left[\sum_{i=1}^{n} X_i \le (1-\gamma)N\mu\right] \le e^{-\gamma^2 N\mu/C_1}$$
$$\Pr\left[\sum_{i=1}^{n} X_i \ge (1+\gamma)N\mu\right] \le e^{-\gamma^2 N\mu/C_2}$$

If $0 < \gamma < 1$ then [5, Fact 4] gives $C_1 = 2$ and $C_2 = 3$. If $0 < \gamma < \frac{1}{2}$ then [2, Thm. 19] gives $C_1 = C_2 = 2\ln 2$.

where the inequality follows from the Markov bound. Now we are ready to apply the result of [1]. We restate equation (2) and Lemma 5.2 of [1] here, using our own terminology.

Lemma E.2 (From [1])**.** *For $t \in [0, D_i/2]$, unit vector $\mathbf{u}_i$ (*i.e.*, $\|\mathbf{u}_i\|_2^2 = 1$) and $\mathbf{Y}_i$ chosen uniformly from $\{1,-1\}^{D_i}$ we have that*

$$\mathbb{E}[e^{t\langle \frac{\mathbf{Y}_i}{\sqrt{D_i}}, \mathbf{u}_i\rangle^2}] \leq \frac{1}{\sqrt{1-2t/D_i}} \tag{33}$$

$$\mathbb{E}[\langle \frac{\mathbf{Y}_i}{\sqrt{D_i}}, \mathbf{u}_i\rangle^2] = \frac{1}{D_i} \tag{34}$$

$$\mathbb{E}[\langle \frac{\mathbf{Y}_i}{\sqrt{D_i}}, \mathbf{u}_i\rangle^4] = \frac{3}{D_i^2} \tag{35}$$

Now, using [1]'s result in (33) we write

$$\begin{aligned}\Pr[\|\mathbf{w}\|_2^2 > (1+\varepsilon)q \mid \neg E_1] &\leq e^{-t\gamma\frac{N^2}{q}} \prod_{i=1}^{N} \mathbb{E}[e^{t\langle \frac{\mathbf{Y}_i}{\sqrt{D_i}}, \mathbf{u}_i\rangle^2} \mid \neg E_1] \\ &\leq e^{-t\gamma\frac{N^2}{q}} \prod_{i=1}^{N} \frac{1}{\sqrt{1-2t/D_i}} \\ &\leq e^{-t\gamma\frac{N^2}{q}} (1 - \tfrac{2t}{(1-\eta)q/N})^{-\frac{N}{2}} \doteq v(t) \end{aligned} \tag{36}$$

where the last inequality (36) follows from conditioning on $\neg E_i$ which implies that $(1-\eta)q/N < D_i$ for all $i \in [N]$. Note that for the result of [1] in (33) to hold, we must have $0 \leq t < D_i/2 \leq \frac{(1+\eta)q}{2N}$ where the last inequality here follows from the fact that $\neg E_i$ implies that $D_i < (1+\eta)q/N$.

Optimizing and bounding t. Next, we optimize $v(t)$ in (36), by finding t such that $\frac{dv(t)}{dt} = 0$.

$$\begin{aligned}\tfrac{dv(t)}{dt} = -\tfrac{\gamma N^2}{q} v(t) + (-\tfrac{N}{2})(-\tfrac{2}{(1-\eta)q/N})(1 - \tfrac{2t}{(1-\eta)q/N})^{-1} v(t) &= 0 \\ \tfrac{\gamma N^2}{q}(1 - \tfrac{2t}{(1-\eta)q/N}) &= \tfrac{N^2}{(1-\eta)q} \\ t &= \tfrac{q}{2N}\left((1-\eta) - \tfrac{(1+\eta)^2}{1+\varepsilon}\right) \end{aligned} \tag{37}$$

where the last equality uses the fact that $\gamma \doteq \frac{1+\varepsilon}{(1+\eta)^2}$. Now recall that for [1]'s result in (33) to hold, we need to ensure that $0 \leq t < \frac{(1+\eta)q}{2N}$. Using (37), we write

$$\begin{aligned} 0 &\leq t \\ 0 &\leq \tfrac{q}{2N}\left((1-\eta) - \tfrac{(1+\eta)^2}{1+\varepsilon}\right) \\ \tfrac{(\eta^2+3\eta)}{1-\eta} &\leq \varepsilon \end{aligned} \tag{38}$$

and we also need

$$\begin{aligned} t &< \tfrac{(1+\eta)q}{2N} \\ \tfrac{q}{2N}\left((1-\eta) - \tfrac{(1+\eta)^2}{1+\varepsilon}\right) &< \tfrac{(1+\eta)q}{2N} \\ -\left(1 + \tfrac{(1+\eta)^2}{2\eta}\right) &< \varepsilon \end{aligned} \tag{39}$$

Now, (39) holds for any $\eta \in [0,1)$. But, we will need to ensure that our choice of $\eta \in [0,1)$ satisfies (38).

Returning now to (36), plug (37) into (36) to get

$$\Pr[\|\mathbf{w}\|_2^2 > (1+\varepsilon)q \mid \neg E_1] \leq \left(e^{-y}(1+y)\right)^{\frac{N}{2}} \tag{40}$$

where we define

$$y \doteq \frac{(1+\varepsilon)(1-\eta)}{(1+\eta)^2} - 1 \tag{41}$$

and solving inequality (38), we find that (40) holds as long as $\eta \in [0,1)$ satisfies

$$0 < \eta < \tfrac{1}{2}\left(\sqrt{\varepsilon^2 + 10\varepsilon + 9} - (\varepsilon+3)\right) \tag{42}$$

Notice from (41) that the bound in (42) this implies that (40) holds for the region $y \in [0, \varepsilon)$. Now, [1] observes that $e^{-y}(1+y) \leq e^{(-y^2/2+y^3/3)}$ for any $y \in (0,1)$. Since for us $y \in (0, \varepsilon)$, and $\varepsilon < 1$ we finally have

$$\Pr[\|\mathbf{w}\|_2^2 > (1+\varepsilon)q \mid \neg E_1] \leq e^{-\frac{N}{2}(y^2/2 - y^3/3)} \tag{43}$$

which decays exponentially in N. □

Bounding the second item. Let r be the number of non-zero entries in $\mathbf{v}$. We will bound $\Pr[\|\mathbf{w}\|_2^2 < (1-\varepsilon)r]$. Define E_1 as before, only this time use r instead of q. Again we condition on $\neg E_1$.

$$\begin{aligned}
\Pr[\|\mathbf{w}\|_2^2 < (1-\varepsilon)r \mid \neg E_1] &= \Pr[\sum_{i=1}^{N} D_i^2 \left(\tfrac{1}{D_i} \sum_{x \in Q_i} Y_x v_x \right)^2 < (1-\varepsilon)r \mid \neg E_1] \\
&= \Pr[\sum_{i=1}^{N} \left(\tfrac{1}{D_i} \sum_{x \in Q_i} Y_x v_x \right)^2 < \tfrac{(1-\varepsilon)}{(1-\eta)^2} \tfrac{N^2}{r} \mid \neg E_1]
\end{aligned}$$

where first equality comes from the expanding $\|\mathbf{w}\|_2^2$and then multiplying by $\frac{D_i}{D_i}$, and the second equality follows from the fact that conditioning on $\neg E_i$ implies that $(1-\eta)r/N < D_i$. Next, we let $c_i^2 = \sum_{x \in Q_i} \frac{v_x^2}{D_i}$. Now observe that $c_i^2 = \frac{1}{D_i} \sum_{x \in Q_i} v_x^2 \geq \frac{1}{D_i} D_i = 1$ since the entries of v are integers (and D_i is the number of non-zero entries in v that are in Q_i). We now multiply by $\frac{c_i}{c_i}$:

$$\begin{aligned}
&= \Pr[\sum_{i=1}^{N} c_i^2 \left(\sum_{x \in Q_i} Y_x \tfrac{v_x}{D_i c_i} \right)^2 < \tfrac{(1-\varepsilon)}{(1-\eta)^2} \tfrac{N^2}{r} \mid \neg E_1] \\
&\leq \Pr[\sum_{i=1}^{N} \left(\sum_{x \in Q_i} Y_x \tfrac{v_x}{D_i c_i} \right)^2 < \tfrac{(1-\varepsilon)}{(1-\eta)^2} \tfrac{N^2}{r} \mid \neg E_1]
\end{aligned}$$

where the inequality follows from the fact that $c_i^2 \geq 1$. We now set $\mathbf{Y}_i$ to be the vector of all Y_x for each $v_x \neq 0, x \in Q_i$. Set $\mathbf{u}_i$ the vector with entries $\frac{v_x}{\sqrt{D_i} c_i}$ for each $v_x \neq 0, x \in Q_i$. Notice that both $\mathbf{Y}_i$ and $\mathbf{u}_i$ have length D_i, and that $\mathbf{u}_i$ is a unit vector, since $\|\mathbf{u}_i\|_2^2 = \frac{1}{D_i c_i^2} \sum_{x \in Q_i} v_x = \frac{c_i^2}{c_i^2} = 1$. We write

$$\begin{aligned}
&= \Pr[\sum_{i=1}^{N} \langle \tfrac{\mathbf{Y}_i}{\sqrt{D_i}}, \mathbf{u}_i \rangle^2 < \tfrac{(1-\varepsilon)}{(1-\eta)^2} \tfrac{N^2}{r} \mid \neg E_1] \\
&\leq e^{t \frac{(1-\varepsilon)}{(1-\eta)^2} \frac{N^2}{r}} \prod_{i=1}^{N} \mathbb{E}[e^{-t \langle \frac{\mathbf{Y}_i}{\sqrt{D_i}}, \mathbf{u}_i \rangle^2} \mid \neg E_1]
\end{aligned}$$

where the first inequality follows from the Markov bound, and we require that $t > 0$. We now follow that analysis in Achiloptas, and expand out the quantity inside the expectation to obtain:

$$\leq e^{t \frac{(1-\varepsilon)}{(1-\eta)^2} \frac{N^2}{r}} \prod_{i=1}^{N} \mathbb{E}[1 - t \langle \tfrac{\mathbf{Y}_i}{\sqrt{D_i}}, \mathbf{u}_i \rangle^2 + \tfrac{t^2}{2} \langle \tfrac{\mathbf{Y}_i}{\sqrt{D_i}}, \mathbf{u}_i \rangle^4 \mid \neg E_1]$$

Now we can apply Achiloptas's results from (34) and (35) to obtain:

$$\leq e^{t \frac{(1-\varepsilon)}{(1-\eta)^2} \frac{N^2}{r}} \prod_{i=1}^{N} \left(1 - \tfrac{t}{D_i} + \tfrac{t^2}{2} \tfrac{3}{D_i^2} \right)$$

and conditioning on $\neg E_1$ gives us:

$$\leq e^{t \frac{(1-\varepsilon)}{(1-\eta)^2} \frac{N^2}{r}} \left(1 - \tfrac{1}{1+\eta} \tfrac{tN}{r} + \tfrac{3}{2(1-\eta)^2} (\tfrac{tN}{r})^2 \right)^N$$

For convience, we'll now let $\tau = \frac{tN}{r}$, and rewrite this as

$$= \left(e^{\frac{(1-\varepsilon)}{(1-\eta)^2} \tau} \left(1 - \tfrac{1}{1+\eta} \tau + \tfrac{3}{2(1-\eta)^2} \tau^2 \right) \right)^N \doteq \nu(\tau)^N \tag{44}$$

Bounding equation (44). We now need to find a choice of $\tau > 0$ that causes (44) to decay with N. It will suffice to find τ that causes $\nu(\tau)$ to decay exponentially, *i.e.,* we want $\nu(\tau) \sim e^{-\chi}$ for some $\chi > 0$. To do this, we start by rewriting $\nu(\tau)$ in the following way:

$$\nu(\tau) = e^{\frac{(1-\varepsilon)}{(1-\eta)^2}\tau}\left(1 - \tfrac{1}{1+\eta}\tau \cdot \left(1 - \tfrac{3}{2}\tfrac{(1+\eta)}{(1-\eta)^2} \cdot \tau\right)\right)$$

Notice that $\nu(\tau)$ is the product of a polynomial and exponential with postive argument (that grows). Notice that the only way we can hope to make $\nu(\tau)$ decay, is if we require the polynomial to decay. To do this, we need to ensure that the expression $(1 - \frac{3}{2}\frac{(1+\eta)}{(1-\eta)^2} \cdot \tau)$ is positive. Thus, we shall choose $\tau = \frac{\varepsilon}{2}(\frac{3}{2}\frac{(1+\eta)}{(1-\eta)^2})^{-1}$. Subsituting in the value for τ gives us:

$$= e^{\frac{1-\varepsilon}{1+\eta}\frac{\varepsilon}{3}}\left(1 - (\tfrac{1-\eta}{1+\eta})^2\tfrac{\varepsilon}{3} \cdot (1 - \tfrac{\varepsilon}{2})\right)$$

The series expansion of an exponential tell us that for any non-negative x we have the identity $1 - x \leq e^{-x}$. Since the quantity $(\frac{1-\eta}{1+\eta})^2\frac{\varepsilon}{3} \cdot (1 - \frac{\varepsilon}{2})$ is non-negative for every $\varepsilon \in (0, 1)$, we can apply this identity here:

$$\begin{aligned} &\leq \exp\left(\tfrac{1-\varepsilon}{1+\eta}\tfrac{\varepsilon}{3} - (\tfrac{1-\eta}{1+\eta})^2\tfrac{\varepsilon}{3} \cdot (1 - \tfrac{\varepsilon}{2})\right) \\ &= e^{-\frac{\varepsilon}{3(1+\eta)}}\exp\left(\tfrac{(1-\eta)^2}{1+\eta}(1 - \tfrac{\varepsilon}{2}) - (1 - \varepsilon)\right) \end{aligned} \tag{45}$$

It follows from (45) that proving that $\nu(\tau)$ decays exponentially amounts to ensuring that

$$y(\eta, \varepsilon) \doteq \frac{(1-\eta)^2}{1+\eta}(1 - \tfrac{\varepsilon}{2}) - (1 - \varepsilon) \geq 0 \tag{46}$$

and, recalling that $\eta, \varepsilon \in (0, 1)$ some algebraic manipulation finds that (46) holds as long as $\eta \in (0, c(\varepsilon))$, where

$$c(\varepsilon) = \tfrac{1}{2-\varepsilon}(3 - 2\varepsilon - \sqrt{5\varepsilon^2 - 14\varepsilon + 9}) \tag{47}$$

This bound on η, despite being ugly, makes sense. Notice that when $\varepsilon = 0$, we have that $\eta = 0$, and when $\varepsilon = 1$, we have $c(\varepsilon) = 1$ so that $\eta \in (0, 1)$. Also, we observe that y monotonically decreases in η, ranging from $y(0, \varepsilon) = \varepsilon$ to $y((c(\varepsilon), \varepsilon) = 0$.[7] We also observe that y monotonically increase in ε, ranging from $y(\eta, 0) = y(0, 0) = 0$ (since $\eta = 0$ when $\varepsilon = 0$), and $y(\eta, 1) = \frac{1}{2}\frac{(1-\eta)^2}{1+\eta}$ (and $\eta \in (0, 1)$ when $\varepsilon = 1$). [8]

Putting everything together, we finally have that as long as $\eta \in (0, c(\varepsilon))$ where $c(\varepsilon)$ is given in (47), then y as given in (46) is such that $y > 0$. Re-writing (44) using (45) and (46) as

$$\Pr[\|\mathbf{w}\|_2^2 < (1 - \varepsilon)r \mid \neg E_1] \leq e^{-N\frac{\varepsilon}{3(1+\eta)}y} \tag{48}$$

we can see that the error decays exponentially in N, as required. □

A simpler statement of the theorem. We now prove Theorem III.2 from Theorem E.1.

Proof of Theorem III.2. We show how to obtain the Theorem III.2 from Theorem E.1. To ensure that the error probability is at most δ in (30) it suffices to set

$$2Ne^{-\frac{\eta^2 q}{3N}} \leq \tfrac{\delta}{2} \tag{49}$$

$$e^{-\frac{N}{2}(y_1^2/2 - y_1^3/3)} \leq \tfrac{\delta}{2} \tag{50}$$

And to ensure that the error probability is at most δ in (31) we need to set

$$2Ne^{-\frac{\eta^2 r}{3N}} \leq \tfrac{\delta}{2} \tag{51}$$

$$e^{-N\frac{\varepsilon}{3(1+\eta)}y} \leq \tfrac{\delta}{2} \tag{52}$$

Bounding N. Referring to (50), we need to choose $N > N_{\min,1}$ where:

$$N_{\min,1} = \frac{4}{y_1^2(1 - y_1/6)}\ln\tfrac{2}{\delta} \tag{53}$$

[7] One can see that when $\eta = 0$, then $y(0, \varepsilon) = \varepsilon$, and a simple check in MATHEMATICA shows that when $\eta = c(\varepsilon)$ as in (47), then $y(c(\varepsilon), \varepsilon) = 0$. By inspection, it follows that y decreases in η.

[8] First consider the case where $\varepsilon = 0$. Now when $\varepsilon = 0$, $c(\varepsilon) = 0$, and the requirement that $\eta \in (0, c(\varepsilon))$ implies that $\eta = 0$. It follows that $y = 0$. Next consider the case where $\varepsilon = 1$, which means that for $\eta \in (0, 1)$, we have that $y(\eta, 0) = \frac{1}{2}\frac{(1-\eta)^2}{1+\eta}$. Now, since the derivative $\frac{dy}{d\varepsilon} = \frac{1+\eta(4-\eta)}{1+\eta} > 0$ for any $\eta \in (0, 1)$, we know that y grow monotonically in ε.

Where recall that $y_1 \doteq \frac{(1+\varepsilon)(1-\eta)}{(1+\eta)^2} - 1$. One can verify that $y_1 \in (0, \varepsilon)$ for any $\eta, \varepsilon \in (0,1)$. To simplify (53), we will now require that $y_1 \geq \varepsilon/2$, which means we can write:

$$\begin{aligned} &\leq \frac{4}{y_1^2(1-\varepsilon/6)} \ln \tfrac{2}{\delta} \\ &\leq \frac{4}{(\varepsilon/2)^2(1-\varepsilon/6)} \ln \tfrac{2}{\delta} \\ &\leq \frac{19.2}{\varepsilon^2} \ln \tfrac{2}{\delta} \end{aligned}$$

where the first inequality follows because $y \leq \varepsilon$, the second follows from $y \geq \varepsilon/2$, and the third follows from $\varepsilon \leq 1$. Now, instead of using the "ugly" expression for $N > N_{\min,1}$ in (53) to bound N, we have "nicer" bound on N that shows the dependence of N on ε, δ as:

$$N \geq \frac{19.2}{\varepsilon^2} \ln \tfrac{2}{\delta} \tag{54}$$

Next, refer to (52), we need to choose $N > N_{\min,2}$ where:

$$N_{\min,2} = \tfrac{3(1+\eta)}{\varepsilon y_2} \ln \tfrac{2}{\delta} \tag{55}$$

Where recall that $y_2 = \frac{(1-\eta)^2}{1+\eta}(1 - \frac{\varepsilon}{2}) - (1-\varepsilon)$. One can see that $y_2 \in (0, \frac{\varepsilon}{2})$ for any $\eta \in (0,1)$. To simplify (53), we will now require that $y_2 \geq \varepsilon/4$ which means we can write:

$$\begin{aligned} &\leq \tfrac{12(1+\eta)}{\varepsilon^2} \ln \tfrac{2}{\delta} \\ &\leq \tfrac{24}{\varepsilon^2} \ln \tfrac{2}{\delta} \end{aligned}$$

where the first inequality follows from our choice of $y_2 \geq \varepsilon/4$ and the second from $\eta \leq 1$. Now we again have "nicer" bound on N (showing it's dependence of N on ε, δ) as:

$$N \geq \tfrac{24}{\varepsilon^2} \ln \tfrac{2}{\delta} \tag{56}$$

Comparing equations (54) and (56) we find that it suffices to choose N satisfying (56).

Bounding η. These nice bounds on N does not come free. To obtain (54), we need to ensure that $y_1 > \varepsilon/2$. We write

$$\begin{aligned} \tfrac{\varepsilon}{2} &\leq y_1 \doteq \tfrac{(1+\varepsilon)(1-\eta)}{(1+\eta)^2} - 1 \\ \tfrac{1+\frac{\varepsilon}{2}}{1+\varepsilon} &\leq \tfrac{1-\eta}{(1+\eta)^2} \end{aligned} \tag{57}$$

Now since $\frac{1+\frac{\varepsilon}{2}}{1+\varepsilon} \leq \left(\frac{1-\eta}{1+\eta}\right)^2 \leq \frac{1-\eta}{(1+\eta)^2}$ it follows that (57) holds if

$$\tfrac{1+\frac{\varepsilon}{2}}{1+\varepsilon} \leq \left(\tfrac{1-\eta}{1+\eta}\right)^2 \tag{58}$$

Next, to obtain (56) we need ensure that $y_2 > \varepsilon/4$, so we write

$$\tfrac{\varepsilon}{4} \leq y_2 \doteq \tfrac{(1-\eta)^2}{1+\eta}(1 - \tfrac{\varepsilon}{2}) - (1-\varepsilon) \tag{59}$$

and a similar argument show that (59) holds as long as

$$\tfrac{1-\frac{3\varepsilon}{4}}{1-\frac{\varepsilon}{2}} \leq \left(\tfrac{1-\eta}{1+\eta}\right)^2 \tag{60}$$

Bounding q, r. Referring to (49) and (51), we observe that is suffices to choose

$$q, r \geq \tfrac{3N}{\eta^2} \ln \tfrac{4N}{\delta} \tag{61}$$

Notice that this bound relies on both N, and η. We bounded N in (56). To minimize q, r, we want to chose η as large as possible, subject to the constraints in (58) and (60). Thus, it suffices to chose η such that

$$\left(\tfrac{1-\eta}{1+\eta}\right)^2 = \max\left(\tfrac{1+\frac{\varepsilon}{2}}{1+\varepsilon}\ ,\ \tfrac{1-\frac{3\varepsilon}{4}}{1-\frac{\varepsilon}{2}}\right) \tag{62}$$

and this completes our proof of Theorem III.2. $\square$