ID-INT: Secure Inter-Domain In-Band Telemetry

Lars-Christian Schulz
OVGU Magdeburg
Magdeburg, Germany
lschulz@ovgu.de

David Hausheer
OVGU Magdeburg
Magdeburg, Germany
hausheer@ovgu.de

***Abstract*—In-band network telemetry (INT) is a powerful tool for gathering status information from network components in a distributed and timely way. Until now, INT has mostly been deployed in data center environments or single operator WANs, because it lacks mechanisms for authentication and is not widely standardized. SCION is a novel, path-based Internet architecture providing strong resilience and security properties. In this paper, we propose Inter-domain In-band Network Telemetry (ID-INT) as a protocol extension for SCION. ID-INT leverages SCION's public key infrastructure to authenticate telemetry data while augmenting SCION's end host path control with real-time network information. Promising applications of ID-INT include intra-AS path tracing, congestion control, SLA verification, and carbon-aware routing. We implement ID-INT in the open-source SCION stack and provide a proof of concept for an AS-hosted telemetry collection service. We show that cryptographically authenticated ID-INT can be fully implemented in the SCION router's fast-path with no measurable impact on router performance. If optional encryption is employed in addition to authentication, router throughput drops by no more than 13% even if every packet carries telemetry.**

***Index Terms*—In-band Network Telemetry, SCION, WAN**

I. Introduction

Network monitoring and measurement is an integral part of any network operator's toolkit. In order to meet the demands of modern real-time applications, constant monitoring of the network's status and performance is required. Traditionally, networks have been monitored through active measurements using probe packets, e.g., using the well-known *ping* and *traceroute* commands, or through passive traffic monitoring at routers. Passive monitoring is usually employing sampling techniques as observing every single packet is costly.

With the advent of SDN, programmable data planes, and P4, a new network monitoring paradigm emerged in the form of push-based network telemetry. Telemetry-enabled devices push network measurements to a central controller, instead of waiting for the controller to poll monitoring data. Fully programmable network devices like Intel's Tofino [1] enable to push telemetry one step further by offloading the collection of telemetry metadata entirely to the data plane. Noticeably, the INT specification [2] was developed as a standardized way to exchange telemetry information between network entities. The INT framework is related to a number of earlier systems all based around the idea of embedding telemetry instructions and is some cases metadata as well in packet headers [3], [4]. INT has in turn inspired research on advanced in-band telemetry protocols like ML-INT for optical networks [5] and probabilistic approaches like PINT [6]. All these systems have in common that they can only be deployed in networks under shared administrative control. Additionally, security and privacy aspects have largely been ignored, precluding Internet-wide deployment.

The SCION Internet architecture [7] has been developed to address the lack of security-by-design in today's Internet based on the Border Gateway Protocol (BGP). BGP's design limitations have caused numerous outages. SCION provides a public key infrastructure for authenticating network entities and allows multiple roots of trust to coexist. Another core feature of SCION is that it is a path-based routing protocol. End hosts include the AS-level forwarding path in packet headers to eliminate uncertainties in traditional routing. The same property also allows end hosts to send traffic to a specific destination over multiple parallel paths to increase reliability and aggregate bandwidth. SCION has been successfully deployed in both research [8] and commercial networks [9] and already reaches hundreds of thousands devices. A challenge of the end host routing approach is to provide sufficient information for making routing decisions to hosts. Current solutions (cf. [10]–[12]) are based on control plane messages and cannot provide real-time feedback from routers to hosts. Therefore, SCION path selection is mostly based on end-to-end measurements, which become challenging as the number of available paths grows with the number of SCION ASes.

In order to address the absence of real-time telemetry in SCION and INT's lack of an authentication infrastructure and inter-operator compatibility, we introduce Inter-Domain In-band Network Telemetry (ID-INT). ID-INT relies on SCION's Dynamically Recreatable Key (DRKey) system to provide efficient message authentication in the data plane and in turn allows SCION end host to make informed routing decisions.

This work is structured as follows: We continue with a brief description of SCION in section II and provide an overview of related work in and outside the SCION ecosystem in section III. ID-INT's design is presented in section IV. section V provides details on our prototype implementation which we evaluate for throughput and overhead in section VI, before we discuss potential extensions to the protocol in section VII. Finally, section VIII gives an outlook on a wide range of applications, while section IX concludes this paper.

978-3-903176-66-9 ©2024 IFIP

II. Background

In-band network telemetry has had great success in single-operator networks in applications like network debugging [13] and congestion control (cf. [14], [15]). Bringing INT to multi-operator networks like the Internet is expected to have similar benefits. However, Internet-wide INT poses a number of new challenges, since operators generally distrust each other. Telemetry data needs to be secured from man-in-the-middle attacks and spoofing attempts. The kind of telemetry exported by switches has to be standardized and universally suitable for many applications. Operators must be able to ensure telemetry data is only disclosed to authorized parties. Moreover, the routing paths are generally outside the control of INT users, which could lead to misleading measurements as packets takes different paths.

A. SCION

Faced with the challenges outline above, an adoption of INT in the current Internet seems unlikely. In contrast, SCION [7] is a future Internet architecture which takes great steps to resolve the lack of trust in the Internet by introducing an explicit trust infrastructure. Autonomous Systems (ASes) are grouped in Isolation Domains (ISDs) that share common jurisdictions. Route hijacking attacks are prevented using cryptographic signatures on all route announcements. Most importantly, SCION end host are in charge of selecting which AS-level path their packets take on the Internet. End host routing solves the otherwise unpredictable routes of INT packets. As hosts must make routing decisions themselves, they have to probe available paths to determine the most suitable ones for a given application. Currently, they mostly rely on end-to-end metrics like round-trip time (RTT) and packet loss to determine the quality and congestion status of a path. With INT, routers could provide accurate and timely information on link status to the end host which should result in much faster convergence to the set of best paths.

A comprehensive introduction to SCION is available by Chuat et al. [7], here we briefly sketch the aspects of SCION relevant to ID-INT. ID-INT provides in-band telemetry to SCION and in turn relies on SCION's control plane and data plane packet format to function. SCION is a routing protocol and Internet architecture replacing today's BGP routing. The ASes forming the Internet are separated in ISDs each with their own consortium of Core-ASes that manage and disseminate root certificates (the TRC — Trust Root Certificate) that are used the sign AS's public keys which authenticate all control plane messages between ASes. SCION discovers path segments by disseminating beacons on inter-AS links. The path segments are uploaded to path servers from which end hosts can retrieve them. Path segments often have a high degree of overlap as beacons flow over all possible links. The amount of path overlap is increased even further, because path segments can be assembled in different ways to form the final forwarding path.

On the data plane, SCION hosts encode the fully assembled path in the packet headers. SCION supports different path types that offer different features such as bandwidth reservations (Colibri [16], Hummingbird [17]) and protection from DDoS attacks (EPIC [18]). Here we discuss only the default SCION path format that provides basic path validity checks at border routers. The AS-level forwarding path consists of a list of *hop fields*. Hop fields are grouped by the path segment they originate from by up to three *info fields* preceding them in the header. Other SCION path types that follow the same general model are also compatible with ID-INT, which currently includes all paths types we are aware of.

B. DRKey

ID-INT takes advantage of SCION's dynamic-recreatable-key (DRKey) system. DRKey allows routers to derive symmetric keys from local secrets using a fast pseudo-random function such as AES. Individual keys can be generated between ASes, between as AS and a host, or even between hosts [7, §3.2]. The keys are not suitable for end-to-end communication as they are known to the AS (i.e., the ISP) of the two end hosts, but they can be used to encrypt and authenticate data produced within one AS such that a host in another AS can decrypt and validate the data. Therefore, it is ideal for securing ID-INT from manipulation by third party ASes that might want to shift blame for an SLA violation to another AS or try to infer topology information that was not supposed to be disclosed to them.

III. Related Work

In-band network telemetry has previously been combined with source routing in the INT-path [19], NetView [20], and NetVision [21] systems. In contrast to the these approaches, ID-INT does not introduce a new routing label stack and instead builds on the SCION Internet architecture. By reusing SCION, we can guarantee deployability of the routing system in inter-domain networks.

Previous inter-domain telemetry systems disseminate measurements in the control plane instead of in-band on the data plane. NetQuery [22] allows operators to validate network properties like the availability of backup links. It exports router configurations such as routing tables to a logically centralized knowledge plane through sanitizers that run operator-authorized commands on behalf of external applications. Trusted computing (i.e., TPMs) is used to prevent sending false information.

FABRID [10] allows SCION hosts to select intra-AS path policies via the inclusion of policy indices in the SCION path. The authors propose using remote attestation using TPMs to strengthen an AS' claims that one of its offered policies has the declared characteristics. Attestation in FABRID cannot guarantee, however, that a packet was actually forwarded by the routers claimed in the path policy. Using ID-INT, FABRID-enabled hosts can verify the intra-domain path of their packets to obtain additional assurance of the correct path policy.

SCION provides path metadata of path segments including latency, bandwidth, and geographical information in its path construction beacons [7, §4.1]. The bandwidth and latency information only represents a theoretical optimum however and does not reflect the current state of the network. As paths often remain valid for many hours, there is also the risk of metadata becoming outdated without a client noticing. GLIDS [11] addresses the path selection problem for latency-sensitive applications by disseminating the propagation delay between border routers to SCION hosts. Since latency information is available right away, hosts using GLIDS do not need to probe potentially hundreds of paths to find the lowest latency. The authors also show that GLIDS can be used as source of information for congestion control algorithms. GLIDS can benefit from ID-INT in two ways. Firstly, ID-INT can be used as source of one-way latency measurements that GLIDS disseminates in the control plane. Secondly, GLIDS only provides a lower bound for latency — not the true latency taking processing and queuing into account.

CIRo [12] enables carbon-aware routing in SCION by forecasting carbon-intensity on inter-domain paths and disseminating the forecast to SCION hosts for path selection. Forecasts cover a range of 24 hours with one hour resolution and are updated in regular intervals. ID-INT includes energy and carbon-emission metadata that can be used as input data to CIRo's forecasts or, if exported to end hosts directly, can be used for making carbon-aware routing adjustments on a very fine-grained timescale.

Tabaeiaghdaei et al. recently proposed Debuglets [23] as network debugging architecture for SCION. Debuglets are programs deployed at border routers that can perform active network measurements on behalf of remote operators. They perform end-to-end measurements between border router interfaces to narrow down the location of a problem and to detect misbehavior. Since no special headers are needed, debuglets can mimic any traffic type and avoid special treatment of measurements packets, but they do require the operator conducting the measurements to obtain measurement slots via a blockchain which limits reaction time. In contrast, ID-INT delivers telemetry within one RTT and is suitable for applications that require real-time feedback such as congestion control. As debuglets have the some capabilities as SCION end hosts, probing with ID-INT packets could be added to debuglets in the future.

The Path Oracle [24] was proposed as a way to share end-to-end path quality measurements between different SCION hosts in the same AS. Hosts donate their measurements using the oracle's API and in turn can retrieve summarized throughput scores of paths to other ASes. As proposed, the path oracle does not have a mechanism to verify the validity of end-to-end measurements and relies an heuristic filtering of improbable reports. The path oracle could benefit from incorporating ID-INT, as ID-INT measurements can be verified at the Oracle without interference from the reporting host. As the oracle is focused on multi-path bandwidth optimization, ID-INT measurements would enable it to recommend sets of paths that do not share common bottlenecks. The oracle concept is similar to our telemetry collector and the projects will be merged in the future.

The work of Pan et al. [25] on privacy-preserving telemetry uses homomorphic encryption between the data collector in the data plane and a centralized controller to protect user privacy by enabling anomaly detection in the control plane without exposing clear text connection data. In contrast, ID-INT applies fast symmetric authentication and encryption directly in the data plane to protect telemetry from on-path attackers.

TCP-INT [15] implements INT as a TCP extension header for telemetry-based congestion control (CC). TCP-INT shares the notion of aggregation functions that reduce the amount of transmitted telemetry with ID-INT, albeit with per-path instead of per-AS aggregation. Another similarity is the combination of end host and network telemetry in the same header. TCP-INT has also been implemented in eBPF on end hosts and in P4 on routers. By interfacing with the TCP congestion control system it is possible to run telemetry-based CC in the Linux kernel [26]. Other works in the field of host-based INT using eBPF include two unrelated systems called Host-INT [27], [28].

IV. ID-INT Design

In-band network telemetry works by embedding requests for telemetry data (often called "instructions") into the headers of regular data packets. When a router receives an INT-tagged packet, it captures the requested telemetry information at the moment the packet is processed and sends the telemetry data to a telemetry sink. The P4.org Application Working Group defines three applications modes of INT [2]. INT-XD (export data) avoids packet tagging by matching packets against a watch list, INT-MX (embed instructions) tags data packets with instructions but does not modify packets on-the-fly, instead sending telemetry directly to a central monitoring system. Neither of these modes are suitable for the inter-domain case as we cannot assume a monitoring plane shared by all operators. Additionally, implementing INT-XD is more difficult in SCION as responding to a packet requires a path in the reverse direction. Reversing paths is too complex an operation to support in every SCION router's fast-path. This leaves us with INT-MD (embed data) which embeds both instructions and telemetry data in packet headers.

ID-INT is built on the ideas of the P4 INT-MD specification. Despite the existence of INT-MD as a standardized INT protocol for P4-enabled devices, we choose not to adapt INT-MD directly to SCION. We developed ID-INT as a new protocol, because INT-MD lacks authentication for the reported data which allows man-in-the-middle attacks on the telemetry. There is also no way to encrypt confidential telemetry data crossing foreign ASes. Additionally, INT-MD lacks telemetry aggregation features which have proven useful in, e.g., TCP-INT [15], and help to reduce the data plane overhead.

A. ID-INT Protocol Specification

There are two modes of operation of ID-INT, (a) between end hosts, and (b) between border routers, called Host-ID-INT and Infrastructure-ID-INT, respectively. Figure 1 shows

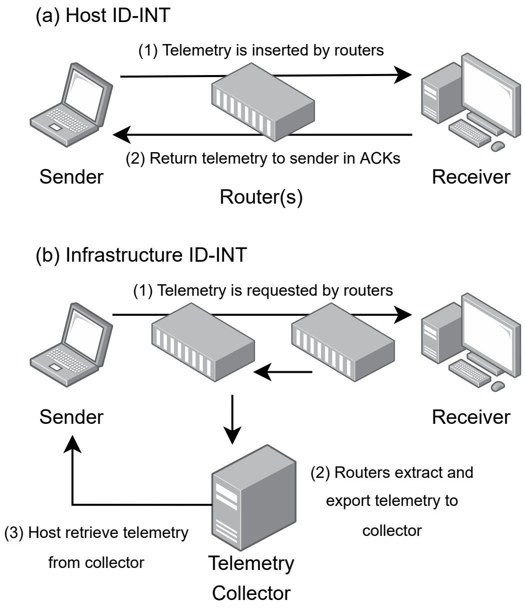
(a) Host ID-INT
(1) Telemetry is inserted by routers
(2) Return telemetry to sender in ACKs
Sender
Receiver
Router(s)
(b) Infrastructure ID-INT
(1) Telemetry is requested by routers
Sender
Receiver
(2) Routers extract and export telemetry to collector
(3) Host retrieve telemetry from collector
Telemetry Collector

Fig. 1. ID-INT telemetry can (a) be delivered directly to hosts or (b) is used by the AS to build a database of path quality measurements.

the two cases. In Host-ID-INT, SCION hosts request telemetry from routers which insert them in the packet headers and deliver the original packet along with the telemetry data to the intended destination host. The destination host is then free to make use of the telemetry data and/or return it to the sender in an application specific way, e.g., an ACK packet. Host-ID-INT is suitable for applications like congestion control and provides the freshest possible information. The second mode is Infrastructure-ID-INT, where telemetry probes are inserted between data packets by the border routers themselves. Probes are terminated at the last border router on the path and telemetry data stored in a telemetry collector. The collector provides an API for hosts to retrieve recent telemetry on different paths to aid in path selection. There is some overlap between both modes, as hosts can employ Host-ID-INT and export the captured data to the telemetry collector sharing it with other hosts.

ID-INT is an extension to SCION carried in a new header following the SCION routing path. SCION's extension mechanism is modeled after IPv6. SCION headers contains a next header field indicating the type of the next header. The chain of headers ends when a layer 4 header such as UDP or TCP is reached. If the next header is not a transport header, it can be a hop-by-hop (HBH) or an end-to-end (E2E) extension header. If all are present, the headers must be in the order SCION-HBH-E2E-L4. The HBH and E2E extension headers are containers for type-length-value (TLV) encoded options. It would be natural to represent ID-INT as a HBH extension header, but unfortunately the maximum length of each extension is 253 bytes, which is insufficient for longer paths, as ID-INT adds between 8 and 64 bytes per hop. Furthermore, implementation of ID-INT as HBH extension is difficult on hardware as headers become deeply nested. Another extension point of the SCION header is the ability to store different path types. Adding ID-INT as a path type would not scale however, as we would have to introduce an ID-INT variant of all existing path types. It is preferable to keep ID-INT and the path type orthogonal. Hence, we treat ID-INT as a new standalone header that if present is inserted between the HBH and E2E extension headers.

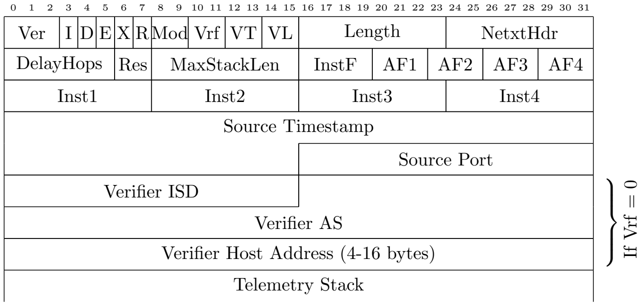

Fig. 2. ID-INT Main Header.

The ID-INT main header[1] is between 20 and 44 bytes long. It is shown in Figure 2. The main header contains instructions for each router on what telemetry data to add to the packet and how to merge it with existing data. Metadata is selected by four instruction bytes (Inst1 to Inst4) which each request a 2, 4, 6, or 8 byte data item. To save space, multiple routers in the same AS can aggregate their telemetry in a single entry, which is controlled by the aggregation functions AF1 to AF4 for each metadata slot separately. The instruction bytes are all equal and can be requested in any order, although some implementations of ID-INT may have additional restrictions such as only supporting a maximum of two instructions or imposing a limit on the maximum amount of bytes added per hop. Besides the instruction bytes, there is a 4-bit instruction bitmap (InstF) which requests one of the following "default" items in addition to the regular metadata:

1) An AS-wide unique **Node ID** identifying the router (4 bytes)
2) A **Node Count** of the number of routers that have updated aggregated data (2 bytes)
3) Ingress and egress **interface IDs** to distinguish different ports of the same router (2 bytes each)

Besides the instructions and obligatory Length and NextHdr fields required for integration in the SCION header stack, ID-INT contains two fields that help to avoid exceeding the MTU before the packet reaches its destination. Routers do not insert telemetry while DelayHops is larger than zero and decrements the field when the packet leaves their AS. When pushing additional telemetry data to the telemetry stack would cause the overall length to exceed MaxStackLen, the MaxHdrSizeExceeded (X) flag is set and no additional data is inserted. The remaining flags indicate whether the INT header

[1] full specification: https://github.com/netsys-lab/id-int-spec

was inserted by a router (I, Infrastructure Mode), whether the packet should not be delivered to an end host (D, Discard), and whether encryption is requested (E, Encrypt).

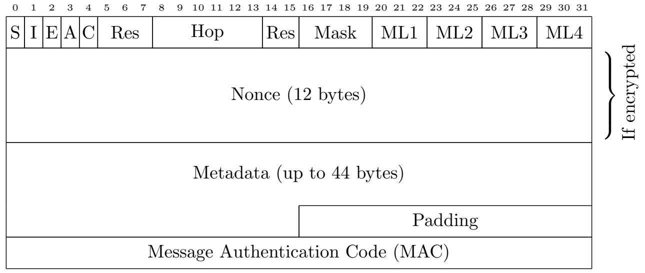

Fig. 3. ID-INT Metadata Header.

The source timestamp and the verifier address (ISD, ASN, and Host Address) are involved in authentication as described in Section IV-E. Following the main header is the telemetry stack consisting of entries formatted as shown in Figure 3. Each stack entry contains a hop index (Hop) corresponding to the index of the hop field that entry relates to. If the entry corresponds to multiple hop fields due to aggregation (subsection IV-C), the index is the first hop field the entry relates to. Telemetry data is concatenated to a byte string and padded to a size divisible by four in order to align telemetry entries at 4-byte boundaries. The overall length and presence of specific data items is indicted by a bitmask (Mask) for the four default metadata and with a length field for each of the remaining four metadata slots (ML1-ML4). As we expect partial telemetry reports to be rather common, i.e., reports that only provide a subset of the requested metadata, encoding report length in every entry individually instead of marking unavailable metadata with a sentinel value like in INT-MD saves space. Flags indicate whether the entry was produced by the INT source (S), whether it was modified at an AS-ingress (I) or AS-egress (E) border router, and whether the entry contains aggregated data (A). If the Encrypted (E) flag is set, the entry has been encrypted with the nonce given in the Nonce field. Every entry also must contain a 4-byte MAC over all flags, length fields, the nonce, and metadata.

B. Available Metadata

Because ID-INT works in multi-operator networks, we need to define a standardized set of metadata instructions. The four "default" types of metadata node ID, node count, ingress and egress interface are necessary in order to localize which router and interface is providing a report. Hence, this metadata type can be combined with any other metadata instruction and does not take up one of the four instruction slots in the main header. The "regular" metadata instructions can be grouped into static and dynamic information. Static metadata is rarely changing and includes general information about the router (e.g., software vendor and version) Constantly changing dynamic metadata is the main focus of ID-INT. It includes device- and port-level statistics as well as queue and buffer utilization. Table 4 gives some metadata examples.

Type	Size	Unit
Device-level Metadata		
CPU/Memory usage	1+1 B	1/255
Forwarding energy per packet.	4 B	nJ
CO_2 emission per packet	4 B	pg
Port Statistics		
Ingress timstamp	6 B	ns
Egress timestamp	6 B	ns
Ingress link RX util.	4 B	kbit/s
Egress link TX util.	4 B	knit/s
Queue/Buffer Utilization		
Queue ID	4 B	
Instantaneous queue length	4 B	packets

Fig. 4. Example of available ID-INT metadata.

Noteworthy examples of supported metadata are ingress and egress timestamps for latency calculations and the link utilization on both ingress and egress ports that can be used for load balancing. The queue ID and queue length are useful for congestion control applications. Moreover, we support green routing by including the amount of energy that was required to forward the packet and the resulting carbon emissions.

C. Telemetry Aggregation

As described above, every ID-INT-capable node adds an entry to the telemetry stack. Sometimes this is undesirable, because the telemetry stack would become too large or because an AS is unwilling to share detailed information about its internal links. In both cases, it is still useful to obtain a condensed summary of the conditions experienced by packets along the entire path, e.g., to detect and locate a bandwidth bottleneck. To this end, ID-INT provides three metadata aggregation modes that aggregate multiple metadata values into a limited number of stack entries.

The aggregation modes allow the ID-INT source to request either a maximum of one telemetry stack entry per entire AS summarizing all border routers and internal nodes, two entries per AS distinguishing between ingress and egress routers, or three entries per AS allowing separate entries for the ingress border router, a summary of internal routers, and an entry for the egress border router. Of course, aggregation can also be disabled, resulting in an entry for every ID-INT node. In practical terms, the "aggregation" of metadata is achieved by first deciding whether a new stack entry must be added or whether the metadata values of the current router should be aggregated into the values written by the previous hop as determined by the role of the node (ingress, egress, or internal) and the requested mode. If the data is to be merged, one of five functions is applied to the already present value and the new metadata. The available functions are: `First`, which will keep the value written be the previous hop. `Last`, which always overwrites the old value with the current router's measurements. `Minimum` and `maximum`, that take the minimum or maximum of the previous and the new value, respectively. And `sum` to accumulate the metadata values as a sum. The aggregation function is selected independently for each of the four instruction words.

Metadata aggregation enables measurements such as finding which AS along a path is the most congested by selecting the maximum aggregation function for the egress port's queue length. As SCION can only control the AS-level path, it is enough to aggregate data for an entire AS into a single entry. Comparing the maximum queue length of all ASes on the path, reveals which AS is congested. If desired, the aggregation mechanism allows finding the router with the highest or lowest value by requesting the node ID metadata, the four "default" metadata are only updated if the telemetry stack entry as a whole is updated, therefore, they represent the last router that has written a value because it was smaller or larger than the previous measurement. The sum aggregation function also enables to calculate averages by dividing the sum by the Node Count metadatum.

D. Source Authentication

Telemetry data is sensitive information to AS operators, therefore, we expect that only certain metadata, i.e., timestamps will be made available to every host requesting it. More sensitive data such as the state of internal routers may only be offered to trusted partners. In order to confidentially identify who originated an ID-INT packet, we need secure source authentication. The ID-INT header does not contain fields for source authentication, as many SCION path types already require source authentication in order to function. If a source authentication tag is not already a part of the routing path, SCION offers a packet authentication option header (SPAO) [7, §3.3] that uses DRKey and a MAC to authenticate the packet source. When ID-INT is combined with source authentication, the contents of the ID-INT main header must be included in the calculation of the authentication tag in order to prevent tempering with the telemetry request.

E. Metadata Authentication and Encryption

Beyond source authentication, ID-INT protects the integrity of telemetry data using a 4-byte MAC attached to every entry on the telemetry stack. The MAC is computed over the entire stack entry including its header, the metadata, padding if necessary, and the MAC of the previous entry. By including the previous MAC, we chain the stack entries together up to the first telemetry entry which serves as anchor and includes additional fields. The first entry on the telemetry stack, the *source metadata*, is mandatory to be written by the INT source, even if no actual telemetry is included. It includes the ID-INT main header in it's MAC. Updateable fields are zeroed when the MAC is computed. In order to detect duplicate reports, the ID-INT main header contains a 48-bit timestamp and a port identifier that together should be unique for every request packet.

Authentication with a MAC requires a shared symmetric key between the communication parties, which in our case are the INT source originating the ID-INT header and the *verifier*. The verifier may be the destination host of the packet, the source host, or a third party identified by the full SCION ISD-AS-Host address triple. Before the MAC is created or checked, an INT node has to derive a secret key shared with the verifier. SCION's DRKey subsystem enables to create such a shared key with any other host for which there is a SCION address. DRKey is uniquely suitable for establishing keys in the data plane as it uses a fast pseudo random function (PRF) to derive host specific keys from secrets shared between ASes. This allows DRKey to generate keys for signing data at AS infrastructure components that can be verified by a SCION host in another AS.

In addition to metadata authentication, ID-INT also supports metadata encryption. If encryption is enabled, each telemetry stack entry also contains a random 96-bit nonce. We mandate that border routers choose nonces at random, as using a counter would require keeping state per ID-INT source. The length of the nonce has to balance space overhead with a sufficiently low probability of nonce collisions. For 2^{32} ID-INT messages, the probability of a collision is approximately $1.16 \cdot 10^{-10}$, at 2^{48} this probability rises to 0.4, so care must be taken to renew keys often enough.

V. Implementation

We implemented ID-INT in the SCION reference router and the primary SCION host library SNET. In order to collect ID-INT measurements, we introduce a telemetry collector capable of parsing telemetry reports, storing them in a time-series database, and offering a REST API retrieving current and historical telemetry data.

A. Choosing Cryptographic Primitives

In order to implement ID-INT, we need to select cryptographic functions for data encryption and authentication. Since ID-INT must work on hardware as well as software routers, we base our choice on the most challenging platforms — P4-programmable switches such as Intel Tofino. Despite the limited environment, AES-128 [29], [30], Ascon-AEAD [31], and HalfSipHash [32] are available for Tofino. We decided on AES for encryption, because Ascon as the main alternative requires many more packet recirculations. We choose AES-CBC MACs for authentication to avoid implementing multiple different primitives, as the same algorithm is required by DRKey. In order to parallelize the encryption operation in ID-INT, we employ AES in counter mode. The MAC and encryption step are combined to form an AES-CCM cipher.

B. Telemetry Collector

We implemented an ID-INT telemetry collector taking INT reports from end hosts in Google protobuf format and storing them in a database for later analysis. The collector has three essential functions: (1) The information contained in the SCION packet headers does not explicitly name every AS on the path. To save space, paths are only encoded as a sequence of ingress and egress interface IDs. In order to map these IDs back to AS identities, the collector has to match the IDs against a database of known paths retrieved and updated from SCION path servers. Additionally, the collector infers internal topology information of the observed ASes, in order to correctly map telemetry data to internal links. (2) The telemetry data itself (latencies, link utilization, etc.) has to be stored for later analysis. (3) The collector provides an API for other end hosts to retrieve historical information on potential paths to use for their routing decisions. The first function of the collector is implemented with the help of a PostgreSQL relational database storing a graph structure of known ASes, routers, links, and paths. The second function is fulfilled by an InfluxDB time-series database. We choose InfluxDB as there is a rich ecosystem of analysis tools available for it. Finally, the client API is designed as REST API offering endpoints aggregated telemetry data from certain routers or links. We note that, other INT collectors can be used with ID-INT as well, but might require some modifications to make use of the path information SCION inherently provides. Figure 5 shows a flow chart of the collector's operation.

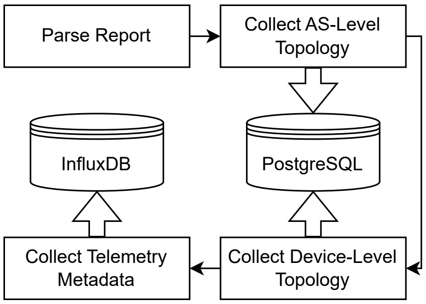
Parse Report
Collect AS-Level Topology
InfluxDB
PostgreSQL
Collect Telemetry Metadata
Collect Device-Level Topology

Fig. 5. ID-INT collector report processing pipeline.

VI. Evaluation

We evaluated the ID-INT collector and ID-INT-enabled border router on servers equipped with AMD Epyc 7543P CPUs interconnected using Nvidia ConnectX NICs with link speeds of at least 100 Gbit/s in order to avoid bottlenecks. Evaluation traffic for the border router was generated using an Intel Tofino 2 switch.

In our testing, the telemetry collector was able to process approximately 50,000 reports/second with numbers fluctuating slightly depending on the size of the reports. We found that the collector was bottlenecked by the mapping of raw telemetry reports to topology nodes stored in a relational database. Higher performance would likely be possible if reports were written directly to Influx DB. The latency of the collector, which we define as the time from receiving a report to the data being reflected in the databases, was measured at around 2.3 seconds, due to aggressive batching of database operations.

We measured the packet processing speed of the ID-INT-enabled SCION border router to assess the overhead on the router side. The measurements were carried out with UDP/SCION packets containing a payload of 1000 bytes. We varied the amount of requested telemetry data between 8 and 42 bytes which is the maximum possible. Our initial results showed a 3% loss in packet throughput compared to no-INT traffic when only telemetry authentication was requested. If telemetry is also encrypted, throughput diminished by up to 33%. We traced most of the performance impact to the standard cryptography library used throughout SCION's code. In order to improve performance, we reimplemented specialized versions of the cryptographic functions used by ID-INT in C, making use of AES-NI instructions. As SCION is implemented in Go, the functions must be called through Go's foreign function interface (FFI) cgo. The improved results in Figure 6 show that ID-INT with custom cryptography via cgo has no measurable impact on the border router throughput as long as only authentication is required. Encryption still has a performance impact of 9% to 13% depending on ID-INT payload size.

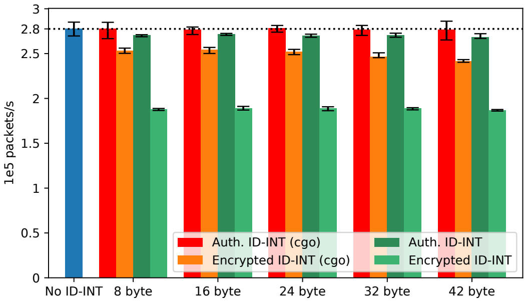

Fig. 6. Packet throughput of the ID-INT-enabled SCION border router for different amounts of requested metadata compared to packets without ID-INT headers. Throughput is shown for ID-INT with only authentication and with additional encryption. Results marked (cgo) make use of AES-NI instructions via Go's cgo FFI. Error bars show 20% and 80% quantiles over 10 repetitions.

While packet throughput is the most important metric for routers, ID-INT also affects applications goodput. The more telemetry headers are inserted in the packet headers, the less space remains for the application payload disincentivizing applications to send ID-INT requests with every data packet which will further reduce the burden on routers. We leave an evaluation of ID-INT applications and application goodput to future work.

VII. Discussion

In this section we discuss the deployment of ID-INT and potential extensions to the protocol and its implementations.

A. Incremental Deployment

ID-INT does not require all routers on the path to be able to fully parse the ID-INT header. SCION routers only need to be able to recognize the ID-INT option and compute the header length from the length field and length of the optional verifier address, in order to access the transport header (typically UDP). As the transport header is only strictly required by the last border router on a path, it is sufficient if SCION edge routers recognize ID-INT. Pure transit routers, that do not allow packets to enter an AS, can remain unchanged. Nevertheless, the value of deploying ID-INT increases with every router that can satisfy telemetry requests. We recognize that not all AS operators wish to share arbitrary telemetry information with any client. If a router does understand ID-INT, but is unwilling to provide telemetry data to the requester, it should append an empty but correctly MACed stack entry. ID-INT availability can also be encoded in SCION's path metadata. As paths that offer ID-INT are more valuable, we expect that ID-INT-enabled paths will attract more customers and become more and more prevalent as time progresses.

B. Hardware Implementation

This work focuses on introducing the ID-INT protocol and the implementation of a proof-of-concept in the SCION software stack. However, ID-INT is carefully designed for implementation in Intel Tofino switches as well [33].

C. Broadcast Authentication

ID-INT uses pairwise symmetric keys to authenticate telemetry requests and data. An alternative architecture worth considering would be to use a broadcast authentication scheme such as TESLA [34] to derive keys in the border routers. Broadcast authentication can be seen as an alternative to the third-party verifier option we have in ID-INT, extended in scope to not just one receiver but potentially many. Not relying on just one verifier would be advantageous for use cases such as SLA verification as broken SLAs could be proven even to entities that do not trust a common telemetry collector service. On the other hand, TESLA would not be suitable to use cases such as congestion control, as there is a substantial delay before a captured massage can be verified.

D. Remote Attestation using TPMs

Some routes equipped with a trusted platform module (TPM) offer remote attestation of the running software to ensure that they are not compromised. As remote attestation relies on expensive asymmetric cryptography, we cannot include it in the ID-INT data plane directly. Remote attestation is useful, however, to assure ID-INT telemetry consumers that the switches employed by an AS operator run a certified and correct implementation of ID-INT, thus strengthening trust in the reported data.

VIII. Applications

We envisage ID-INT to be useful for both traditional INT tasks such as network debugging and as input to congestion control algorithms, as well as for new applications specific to SCION. We give a brief overview below.

Intra-AS Path Tracing. SCION only controls the AS-level path. Routing within ASes is up to the operator and might be unpredictable or use ECMP to combine multiple paths. ID-INT can make the internal routing transparent, or if such detailed information is kept confidential, can be used to detect whether a network overload is happening at the ASes edges or within its core. Particularly interesting is the combination of ID-INT with FABRID a policy based intra-domain path selection system for SCION [10], where ID-INT can be used to obtain more up-to-date information than available in FABRID's relatively static policies.

Path Selection. SCION encodes the AS-level path in the packet header when the packet is created at the network edge. Host usually have a choice of at least a few dozen paths to reach any given destination. To select the most suitable path or paths for multi-path protocols, the end host only gets static path information such as the number of hops, the link capacity, and the path MTU from the SCION control plane. The actual current conditions of the path can only be determined by sending traffic over a path and evaluating metrics like RTT and packet loss. Testing every path in this manner is time consuming and inefficient. If ID-INT is available, the status of every INT-enabled AS on the path is available within one RTT. Additionally, ID-INT does not only offer congestion and latency information for the entire path, but for every router on the path. If the bottleneck router is included in more than one path, the sender can immediately establish a lower bound for the quality of those paths as well, to quickly shrink the search space.

Congestion Control. Once the initial path(s) have been selected, connection oriented protocols such as TCP and QUIC must change the size of their congestion window to avoid overloading the network. INT has been proposed as a source of feedback to congestion control (CC) and shows promising results [14], [15], [26]. ID-INT enables the implementation of INT CC on Internet-scale.

SLA Verification. When end users (e.g., a power plant) and Internet Service Providers (ISPs) or different ISPs enter a contract, they typically sign a Service Level Agreement (SLA) that specifies a set of service level objectives (SLOs) that must be maintained by the provider [35]. It is in the best interest of the contract parties to continuously monitor SLOs, to ensure the service is provided correctly. While not all parts of an SLA can be monitored using INT, characteristics such as latency, latency variance, bandwidth and routing hardware capabilities are in scope for ID-INT.

Carbon-Aware Routing. SCION's path awareness enables end hosts to select paths causing the lowest carbon emissions. AS power consumption and carbon emission can clearly not be measured end-to-end, a system for dissemination of the environmental impact each hop has is required. To this end, CIRo (Carbon-aware Inter-domain Routing) was designed to broadcast carbon-intensity forecasts in SCION routing messages [12]. We include power consumption and carbon emission metrics in ID-INT to provide instantaneous metrics to augment CIRo's 24h forecasts.

IX. Conclusion

We introduced ID-INT as an in-band telemetry extension to the SCION routing architecture. ID-INT leverages SCION's public key infrastructure to ensure trustworthiness of INT data even in multi-operator networks. We designed ID-INT to incorporate as many features as possible from other INT-type protocols to support a wide-range of use cases. The broad applicability of ID-INT will pave the way for more intelligent path selection, better accountability, and more efficient congestion control.

Our preliminary results show that ID-INT can be integrated in the existing SCION software stack easily. Packet throughput of the open-source SCION router is not affected by ID-INT as long as only telemetry authentication is requested. Telemetry encryption incurs a performance penalty of 9% to 13%, but could likely be optimized further.

We conclude by remarking that in-band telemetry is a natural fit to SCION's transparency-focused approach to inter-domain routing. We expect ID-INT to mesh well with many existing and future SCION projects. Especially promising is the ability to use ID-INT for path selection that we will explore further in future work.

References

[1] Intel, "Open Tofino," 2022. [Online]. Available: https://github.com/barefootnetworks/Open-Tofino

[2] P4.org Application Working Group, "In-band Network Telemetry (INT) Dataplane Specification Version 2.1," 2020. [Online]. Available: https://p4.org/specs/

[3] N. Handigol, B. Heller, V. Jeyakumar, D. Mazières, and N. McKeown, "I Know What Your Packet Did Last Hop: Using Packet Histories to Troubleshoot Networks," in *NSDI'14*. USA: USENIX Association, 2014, p. 71–85.

[4] V. Jeyakumar, M. Alizadeh, Y. Geng, C. Kim, and D. Mazières, "Millions of Little Minions: Using Packets for Low Latency Network Programming and Visibility," in *SIGCOMM '14*. New York, NY, USA: ACM, 2014, p. 3–14.

[5] B. Niu, J. Kong, S. Tang, Y. Li, and Z. Zhu, "Visualize Your IP-Over-Optical Network in Realtime: A P4-Based Flexible Multilayer In-Band Network Telemetry (ML-INT) System," *IEEE Access*, vol. 7, pp. 82 413–82 423, 2019.

[6] R. Ben Basat, S. Ramanathan, Y. Li, G. Antichi, M. Yu, and M. Mitzenmacher, "PINT: Probabilistic In-Band Network Telemetry," in *SIGCOMM '20*. New York, NY, USA: ACM, 2020, p. 662–680.

[7] L. Chuat, M. Legner, D. Basin, D. Hausheer, S. Hitz, P. Müller, and A. Perrig, *The Complete Guide to SCION*. Springer, 2022.

[8] OVGU Magdeburg and ETH Zurich. (2024) SCION Education, Research and Academic Network. [Online]. Available: https://sciera.readthedocs.io/

[9] Anapaya Systems AG. (2024) Secure, resilient, and controlled connectivity with SCION. [Online]. Available: https://www.anapaya.net/

[10] C. Krähenbühl, M. Wyss, D. Basin, V. Lenders, A. Perrig, and M. Strohmeier, "FABRID: Flexible Attestation-Based Routing for Inter-Domain Networks," in *USENIX Conference on Security Symposium*, ser. SEC '23. USA: USENIX Association, 04 2023.

[11] C. Krähenbühl, S. Tabaeiaghdaei, S. Scherrer, M. Frei, and A. Perrig, "Toward global latency transparency," in *IEEE/IFIP Networking, recent results track*, 2024.

[12] S. Tabaeiaghdaei, S. Scherrer, J. Kwon, and A. Perrig, "Carbon-Aware Global Routing in Path-Aware Networks," in *Future Energy Systems*, ser. e-Energy '23. New York, NY, USA: ACM, 2023, p. 144–158. [Online]. Available: https://doi.org/10.1145/3575813.3595192

[13] Y. Zhu, N. Kang, J. Cao, A. Greenberg, G. Lu, R. Mahajan, D. Maltz, L. Yuan, M. Zhang, B. Y. Zhao, and H. Zheng, "Packet-Level Telemetry in Large Datacenter Networks," in *SIGCOMM '15*. New York, NY, USA: ACM, 8 2015, p. 479–491.

[14] Y. Li, R. Miao, H. H. Liu, Y. Zhuang, F. Feng, L. Tang, Z. Cao, M. Zhang, F. Kelly, M. Alizadeh, and M. Yu, "HPCC: High Precision Congestion Control," in *SIGCOMM '19*. New York, NY, USA: ACM, 2019, p. 44–58.

[15] G. Jereczek, T. Jepsen, S. Wass, B. Pujari, J. Zhen, and J. Lee, "TCP-INT: lightweight network telemetry with TCP transport," in *SIGCOMM '22 Poster and Demo Sessions*, ser. SIGCOMM '22. New York, NY, USA: ACM, 2022, p. 58–60.

[16] G. Giuliari, D. Roos, M. Wyss, J. A. García-Pardo, M. Legner, and A. Perrig, "Colibri: A Cooperative Lightweight Inter-Domain Bandwidth-Reservation Infrastructure," in *CoNEXT '21*. New York, NY, USA: ACM, 2021, p. 104–118.

[17] G. Giuliari, M. Legner, A. Perrig, J.-P. Smith, and K. Wüst, "Hummingbird: A Flexible and Lightweight Inter-Domain Bandwidth-Reservation System," 2023. [Online]. Available: https://arxiv.org/abs/2308.09959

[18] M. Legner, T. Klenze, M. Wyss, C. Sprenger, and A. Perrig, "EPIC: Every Packet is Checked in the Data Plane of a Path-Aware Internet," in *USENIX Conference on Security Symposium*, ser. SEC'20. USENIX Association, 2020.

[19] T. Pan, E. Song, Z. Bian, X. Lin, X. Peng, J. Zhang, T. Huang, B. Liu, and Y. Liu, "INT-path: Towards Optimal Path Planning for In-band Network-Wide Telemetry," in *INFOCOM 2019*. IEEE, 2019, pp. 487–495.

[20] Y. Lin, Y. Zhou, Z. Liu, K. Liu, Y. Wang, M. Xu, J. Bi, Y. Liu, and J. Wu, "NetView: Towards On-Demand Network-Wide Telemetry in the Data Center," in *ICC 2020*. IEEE, 2020, pp. 1–6.

[21] Z. Liu, J. Bi, Y. Zhou, Y. Wang, and Y. Lin, "NetVision: Towards Network Telemetry as a Service," in *2018 IEEE 26th International Conference on Network Protocols (ICNP)*. IEEE, 2018, pp. 247–248.

[22] A. Shieh, E. G. Sirer, and F. B. Schneider, "NetQuery: a knowledge plane for reasoning about network properties," *SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 4, p. 278–289, aug 2011.

[23] S. Tabaeiaghdaei, F. Costa, J. Kwon, P. Bamert, Y.-C. Hu, and A. Perrig, "Debuglet: Programmable and Verifiable Inter-domain Network Telemetry," in *Proceedings of the IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2024.

[24] T. Krüger, C. Beck, and D. Hausheer, "Path Oracle: Improving Performance of Path-Aware Applications in SCION," in *CNSM '23*, 2023, pp. 1–5.

[25] X. Pan, S. Tang, S. Liu, J. Kong, X. Zhang, D. Hu, J. Qi, and Z. Zhu, "Privacy-Preserving Multilayer In-Band Network Telemetry and Data Analytics: For Safety, Please do Not Report Plaintext Data," *Journal of Lightwave Technology*, vol. 38, no. 21, pp. 5855–5866, 2020.

[26] J.-T. Hinz, V. Addanki, C. Györgyi, T. Jepsen, and S. Schmid, "TCP's Third Eye: Leveraging eBPF for Telemetry-Powered Congestion Control," in *Proceedings of the 1st Workshop on eBPF and Kernel Extensions*, ser. eBPF '23. New York, NY, USA: ACM, 9 2023, p. 1–7. [Online]. Available: https://doi.org/10.1145/3609021.3609295

[27] Intel, "Host-INT for packet-telemetry," 2012. [Online]. Available: https://github.com/intel/host-int

[28] T. Osiński and C. Cascone, "Achieving End-to-End Network Visibility with Host-INT," in *ANCS '21*. ACM, 2021, p. 140–143.

[29] X. Chen, "Implementing AES Encryption on Programmable Switches via Scrambled Lookup Tables," in *SPIN '20*. New York, NY, USA: ACM, 2020, p. 8–14.

[30] L.-C. Schulz, R. Wehner, and D. Hausheer, "Cryptographic Path Validation for SCION in P4," in *EuroP4 '23*. New York, NY, USA: ACM, 12 2023, p. 17–23.

[31] A. Bhatnagar, X. Z. Khooi, C. H. Song, and M. C. Chan, "P4EAD: Securing the In-band Control Channels on Commodity Programmable Switches," in *EuroP4 '23*, ser. EuroP4 '23. New York, NY, USA: ACM, 12 2023, p. 41–48.

[32] S. Yoo and X. Chen, "Secure keyed hashing on programmable switches," in *SPIN '21*. New York, NY, USA: ACM, 8 2021, p. 16–22.

[33] R. Wehner, T. John, L.-C. Schulz, and D. Hausheer, "Secure In-Band Network Telemetry for the SCION Internet Architecture on Tofino," in *EuroP4 '24*. ACM, 2024.

[34] A. Perrig, R. Canetti, D. Song, P. D. Tygar, and B. Briscoe, "Timed Efficient Stream Loss-Tolerant Authentication (TESLA): Multicast Source Authentication Transform Introduction," RFC 4082, jun 2005. [Online]. Available: https://www.rfc-editor.org/info/rfc4082

[35] A. Akbari-Moghanjoughi, J. R. de Almeida Amazonas, G. Santos-Boada, and J. Solé-Pareta, "Service Level Agreements for Communication Networks: A Survey," 2023. [Online]. Available: https://arxiv.org/abs/2309.07272