# Homomorphic Routing: Private Data Forwarding in the Internet

Francesco Tusa*
University of Westminster
London, United Kingdom
f.tusa@westminster.ac.uk

David Griffin
University College London
London, United Kingdom
d.griffin@ucl.ac.uk

Miguel Rio
University College London
London, United Kingdom
miguel.rio@ucl.ac.uk

## ABSTRACT

We propose a new private routing and packet forwarding scheme for the Internet—Homomorphic Routing (HR)—that enables endpoints to communicate with one another without divulging source or destination addresses to the routers or service providers along the path. This is achieved via homomorphic encryption, whereby domains can match encrypted address ranges with encrypted destinations of packets without the need of decryption. Compared to approaches such as source or onion routing, HR is a hop-by-hop solution that allows current BGP-like decisions and traffic engineering techniques to remain largely unchanged, while per-flow state need not be maintained by routers. Preliminary performance evaluation shows that HR implies a tolerable computational overhead compared to plain text operations. Through aggregation we can compress inter-domain routing rules to around 5% of those required for current IPv6 and we can organize encrypted forwarding rules so that matching can be achieved in logarithmic time.

**ACM Reference Format:**
Francesco Tusa, David Griffin, and Miguel Rio. 2023. Homomorphic Routing: Private Data Forwarding in the Internet . In *2nd ACM SIGCOMM Workshop on Future of Internet Routing & Addressing (FIRA '23), September 10, 2023, New York, NY, USA*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3607504.3609287

## 1 BACKGROUND AND MOTIVATION

Despite the success of public key cryptography in ensuring that data exchanged between communicating endpoints remains confidential, today's IP packet forwarding requires that IP addresses—and hence the identities of the endpoints—are conveyed as plaintext. This implies that any network domain along the path (or an eavesdropper) may determine who is communicating with whom, or the service being accessed by a user, even if—thanks to end-to-end encryption—they can not identify *what* is being communicated. This is a fundamental privacy issue at the core of today's Internet.

Overlays such as virtual private networks (VPNs) and onion routing services can protect the confidentiality of users and hide the identity of communicating endpoints from intervening networks and their routers. However, these techniques create an encapsulation overhead, detour traffic through potentially inefficient paths, and bypass the traffic routing policies of network operators. All of which can reduce performance for both users and network providers.

Ideally, the packet forwarding paradigm would make tying a particular user to a particular service as difficult as possible whilst maintaining the ability of providers to apply routing and traffic engineering policies. Homomorphic Routing (HR) achieves this under the assumption that domains are honest-but-curious, meaning that they implement the HR protocol correctly without additional collusion through side-channels or flow tagging but are curious to know the identities of the communicating endpoints.

## 2 INTRODUCTION

Our framework mirrors today's inter-domain architecture with Autonomous Systems (ASs)—*domains* in this paper—exchanging reachability information to make routing decisions. However, in our case, both the routing information exchanged and the destination address of a packet are encrypted. Fig. 1 presents an overview of HR. There are five main entities: Range Announcers (RAs) generate and propagate encrypted ranges on behalf of a domain; Route Calculators (RCs) calculate encrypted forwarding tables and announce routes to neighbor domains; Encrypted Packet Forwarders (EPFs) forward packets between domains; clients initiate private connections with servers; and, finally, a trusted third party—the Homomorphic Encryption Parameters Service (HEPS)—distributes relevant security parameters to the above entities.

Our system works at layer 3 (L3). However, we distinguish the initial packet of an HR flow from subsequent data packets to minimize the number of packets that trigger homomorphic matching and path calculation operations. Only INIT packets are matched using HR, all subsequent data packets are forwarded according to pre-computed paths conveyed in packet headers. In the following, and referring to Fig. 1, we present a first, high-level pass through the main steps of system operation before going into more detail in Section 3.

**Step 1:** The domain's RA contacts the HEPS in order to obtain an encrypted and signed version of an address range (step 1a in Fig. 1). It propagates the encrypted range to neighboring domains following the BGP-like overlay (step 1b). Within each domain, the RC selects the best routes, based on the ranges' attributes and local policies, and populates local forwarding tables in the EPF (step 1c).

**Step 2:** A client wishing to communicate with another endpoint obtains homomorphic encryption (HE) parameters from the HEPS (step 2a). It uses them to encrypt the address of the destination, which is conveyed in an INIT packet for the flow (step 2b). The EPFs at each domain use HE operations to match the encrypted destination with the encrypted ranges in the forwarding rules created in step 1c to identify the next hop. The packet is forwarded to the next domain (steps 2c, 2d) until the final one forwards the INIT packet to the destination host (step 2e). As the INIT packet progresses along the

*Also with University College London.



*FIRA '23, September 10, 2023, New York, NY, USA*

ACM ISBN 979-8-4007-0276-1/23/09.
https://doi.org/10.1145/3607504.3609287

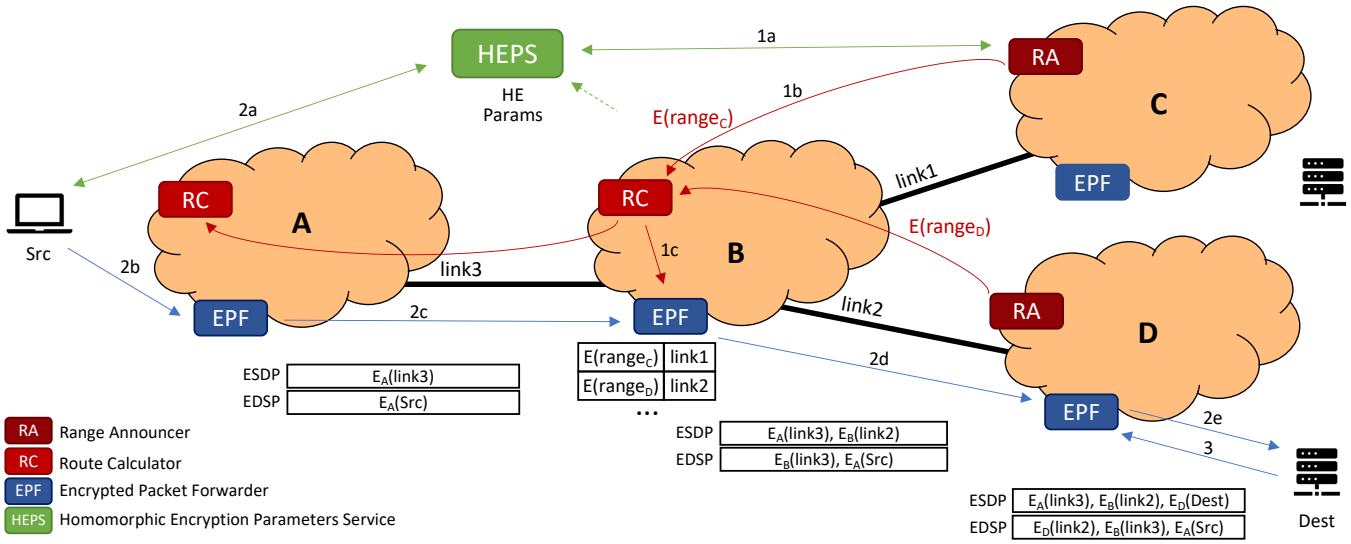


**Figure 1: System Operation**

path, an encrypted pair of forward and reverse paths are constructed for use during the data transfer phase of the flow.

**Step 3:** An INIT reply is sent back to the client to complete flow establishment.

**Step 4:** Data packets follow the encrypted paths constructed in step 2, which are conveyed in the headers of the data packets themselves. This prevents HE matching operations from being performed on every data packet and per-flow state to be maintained by intermediate domains.

**Step 5:** If inter-domain routes change during the data transfer phase of the flow—due to link failures, for example—a subsequent INIT packet is required to signal the hop-by-hop recalculation of the encrypted forward and reverse paths.

The rest of the paper is organized as follows: in the next section we present HR in more detail; in Section 4 we show preliminary evaluation results and in Section 5 we discuss key implementation considerations; we finish with related work in Section 6 and conclusions in Section 7.

## 3 SYSTEM OPERATION

### 3.1 Introduction to Paillier Homomorphic Encryption

The security mechanisms discussed in this paper are inspired by the Paillier cryptosystem: an HE scheme whereby the encrypted sum $m_1 + m_2$, of two messages $m_1$ and $m_2$, can be computed from the individual encrypted values of $m_1$ and $m_2$. Further details are omitted in this paper but can be found in [17]. In the original Paillier cryptosystem, the tuple $(\lambda, \mu)$ is the *private key* and the tuple $(n, g_p)$ is the *public key*. However, it can be proved that $\mu$ does not need to be private since it is hard to decrypt an encrypted message by only knowing $\mu$ [16]. Hence, $\mu$ can be made public while achieving the same security guarantees as the unmodified Paillier cryptosystem – the new public and private keys become $(n, g_p, \mu)$ and $\lambda$ respectively.

Like in [16], the above keys are used differently from the original Paillier cryptosystem. Only those holding the private key can encrypt, whereas the decryption is performed via the public key. Because $\lambda$ is now utilized during encryption, together with $n$ and $g_p$, the resulting computational complexity is higher than in the original Paillier. On the other hand, $\lambda$ is no longer used for the decryption, reducing the associated computational cost. All the HE operations of the original Paillier cryptosystem are still available, as this modification only shifts computational overhead from decryption to encryption.

Confidential information is encrypted in a special way—called *blinding*—using the private key. Blinded values are semantically secure as two blinding operations performed on the same plaintext produce different blinded values. The blinding operation is devised so that the original plaintext cannot be obtained by solely decrypting the blinded value with the public key. However, when two blinded values are multiplied, some of the blinding parameters cancel out due to the HE properties of Paillier. A randomized difference between the original values is obtained via decryption of the product. It is utilized here, as in [16], to determine whether $plaintext_1 \geq plaintext_2$ or $plaintext_1 < plaintext_2$, without learning the original plaintexts.

In HR each range is described by two blinded values: *low*, the lowest IP address in the range and *high*, the highest IP address in the range. Ranges generated by RAs, and destination addresses specified by clients, are both blinded using the private key. RCs use the public key and the blinded ranges in order to build forwarding tables. This is done via the HE operations discussed in Section 3.2, which perform the previous inequality checks on the blinded *low* and *high* values of the ranges. EPFs use the same public key and the blinded ranges in the forwarding tables to perform HE matching with blinded destination addresses for packet forwarding. At forwarding time, the destination address (*dst*), specified within an INIT packet, matches a given range when $dst \geq low \wedge dst \leq high$. Thanks to homomorphic encryption, these inequalities can be checked without learning the original plaintext values of *low*, *high* and *dst*.

The HEPS in a trusted third party in our system that takes care of distributing the required security parameters [16] among clients, RAs, and RCs/EPFs, such that data confidentiality and privacy are preserved when performing the above HE operations. While the public key is made accessible by the HEPS to RCs/EPFs, private parameters are not divulged and blinding operations are performed by the trusted third party on behalf of both clients and RAs.

RAs authenticate with the HEPS and send the plaintext versions of *low* and *high* that define a range to be blinded. The HEPS checks the range validity and adds a digital signature before returning the blinded range's values to an RA. By strictly following this approach, authorized RA must interact with the HEPS to blind every range, hence the risk of malicious RAs injecting invalid ranges into the system is mitigated. To overcome the need for clients to interact with the HEPS to blind every destination address *dst*, we adopt the solution proposed in [15]. Clients are initialized with parameters provided by the HEPS that they use in subsequent *self-blinding* operations, without needing further interaction with the HEPS.

Since the private key is kept secret by the HEPS, it can not be leaked by malicious clients or RAs; this prevents the security of the system from being compromised. With the solution in [15], blinding is devised so that HE checks could only be performed in our system between blinded ranges and destination addresses at forwarding time. Hence, here we extend that design so that an additional pair of blinded values is calculated for *low* and *high*, which is used to support HE operations that compare different blinded ranges in order to populate the forwarding tables.

## 3.2 STEP 1: Routing

An RC receives blinded ranges announced by neighboring domains, accompanied with *plaintext* attributes, the most important being AS path length. It first checks the range's validity by verifying the digital signature added by the HEPS. Then, similarly to BGP, the RC stores the received ranges in its routing information base (RIB), which will subsequently be used: i) to create forwarding rules in the EPF of the local domain—the forwarding information base (FIB), and, ii) to populate the announcements table (AT) whereby ranges are propagated to neighbor domains.

HR deviates from the traditional network prefixes and slash notation in Classless Inter-Domain Routing (CIDR) [12]. Instead, each prefix is converted into a range specified by its lower and upper bounds. This allows forwarding decisions to be made based on HE matching of blinded addresses with blinded ranges; also, it supports much more compression than CIDR aggregation, which requires every subnetwork to be included in the prefix for the larger network. To minimize the size of the RIB, FIB and AT, we specifically rely on the homomorphic operations explained next.

*3.2.1 Range Comparison, Aggregation and Splitting.* The RC needs to perform range comparisons and undertake *aggregation* and *split* operations, all of which can be done homomorphically. Given two ranges, $range_1(low_1, high_1)$ and $range_2(low_2, high_2)$ the following *homomorphic operations* are defined.

**Cover:** $range_1$ covers $range_2$ if $low_1 \leq low_2 \wedge high_1 \geq high_2$.
**Intersection:** $range_1$ intersects $range_2$ if $low_1 < low_2 \wedge low_2 < high_1 \wedge high_1 < high_2 \vee low_2 < low_1 \wedge low_1 < high_2 \wedge high_2 < high_1$.
**Contiguity:** $range_1$ is contiguous with $range_2$ if $low_2 > high_1 \wedge low_2 - 1 < high_1 + 1 \vee low_1 > high_2 \wedge low_1 - 1 < high_2 + 1$.
**Disconnection:** $range_1$ is disjoint from $range_2$ if $high_1 + 1 < low_2 \vee high_2 + 1 < low_1$.
**Aggregation:** If $range_1$ covers $range_2$ then the aggregate is $range_a(min(low_1, low_2), max(high_1, high_2))$. Although *max* and *min* functions cannot be performed directly with HE operations, we know the higher and lower values from the range intersection checks performed previously.
**Split:** $range_1(low_1, high_1)$ can be split at *mid* where $low_1 < mid < high_1$ into two ranges: $range_{1_a}(low_1, mid)$ and $range_{1_b}(mid+1, high_1)$ or $range_{1_a}(low_1, mid-1)$ and $range_{1_b}(mid, high_1)$, depending on whether we are splitting to the left or the right of *mid*.

*3.2.2 Routing Information Base Creation.* The RC makes routing decisions by selecting the best next hop for a range. If there are multiple options for a range, then a priority is assigned according to the range's attribute values, as in BGP today. The attributes are unencrypted to allow comparisons and tie-breaking decisions to be made easily. We assume that the full AS path is not passed between domains but only the AS path length, so that domains cannot infer the identity of the blinded ranges.

For instance, a new range received by the RC that is *cover*ed by an existing range with the same attribute values can be discarded; if a received range *cover*s an existing one with the same attribute values, then *aggregation* can be performed in order to obtain a single range. Conversely, if a new range overlaps an existing range through *cover* or *intersection* relationships, but they have different attribute values, they should be *split* so that the common sub-ranges with worse attribute values can be discarded. Table 1 describes the complete RIB insertion process, which is performed when a new range announcement is compared with existing range entries. RIB entries will be aggregated as much as possible so that contiguous ranges with the same attributes are covered by a single entry.

The RIB is eventually transformed into the FIB, for packet forwarding, and the AT, for announcing to neighboring domains, as explained in the two following subsections.

*3.2.3 Forwarding Table Creation.* When converting the RIB into forwarding rules in the EPF's FIB, a significant amount of further aggregation can be done. The EPF is only concerned with the *result* of the RC's route prioritization and selection process, i.e., the next hop for a given range, rather than any details of range attributes. Hence, all contiguous ranges for the same next hop, irrespective of priority, may be aggregated as a single rule in the FIB.

Since the resulting set of rules will not intersect one another, they can be ordered in a data structure that allows for logarithmic search (e.g., a binary tree); this reduces the quantity of HE matching tests to be done on incoming INIT packets at HR flow creation time.

*3.2.4 Range Announcement.* A similar aggregation process can be done when converting the RIB into the ranges to be announced to neighboring domains. The outgoing next-hop is not of concern to the neighbor, only the AS path length and Multi-Exit Discriminator (MED) attributes are important. All ranges with the same AS path length are candidates for aggregation in the AT, irrespective of the next-hop. Whenever the AT is modified following the arrival of a

| | Cover | Intersection | Contiguity | Disconnection |
|---|---|---|---|---|
| Same next hop | If same path attributes then (1) Insert aggregate range (2) Remove old range, else Insert incoming range | | | Insert incoming range |
| Different next hop | (1) Split into 3 parts (2) Choose best ranges (3) Aggregate them if possible (4) Insert new range(s) (5) Remove old range | (1) Split common part (2) Create two contiguous ranges adding common part to the best one and insert them (3) Remove old range | Insert incoming range | |

**Table 1: Overview of Routing Information Base (RIB) Creation**

range announcement from a neighboring domain, the AT updates are propagated to the domain's other neighbors.

## 3.3 STEP 2: INIT Packet Creation and Forwarding

*3.3.1 INIT Packet Creation.* A destination address in HR consists of two parts: i) *Enc_HE(dst)*, the address blinded by HE, which is used by all domains along the path to route the packet to the destination domain; ii) *Enc_PK(dst)*, the address encrypted with the public key of the destination domain, which is used only in the destination domain to route the packet to the correct host.

A client, wishing to communicate with another endpoint using HR, first needs to retrieve the destination address through DNS or another system that ensures the confidentiality of queries. Given the destination address, the destination's domain ID needs to be identified to retrieve the public key of that domain, which will be used to encrypt the address to create *Enc_PK(dst)*. Mechanisms to ensure the privacy of name resolution and for public key distribution are out of scope of this paper.

Next, the client obtains the HE parameters from the HEPS for blinding the destination address, i.e., to create *Enc_HE(dst)*. HE parameters can be cached by clients to create self-blinded destination addresses for multiple flows without needing to interact with the HEPS each time.

*3.3.2 Rule Matching.* When an INIT packet arrives at a domain, the EPF *match*es the blinded destination address with the blinded address ranges in its FIB to find the next-hop decided by the RC's routing process. A blinded destination address *Enc_HE(dst)* matches a blinded range, $range_i(low_i, high_i)$, if $low_i \leq Enc\_HE(dst) \leq high_i$. As previously discussed, honest-but-curious EPFs cannot decrypt blinded destination addresses as they perform a *match*.

When the packet reaches the destination domain, *Enc_PK(dst)* is decrypted using the domain's private key and the packet can be forwarded normally through the domain's interior gateway protocol routing system. Although the destination domain is able see the plaintext version of *dst*, it will not know the source address; hence, the privacy of the communicating parties is preserved.

*3.3.3 ESDP/EDSP Calculation.* The encrypted destination address in a flow's INIT packet is fully matched homomorphically against forwarding rules, as described above. Subsequent data packets are forwarded according to Encrypted Source-Destination Paths (ESDPs) and Encrypted Destination-Source Paths (EDSPs). These are sequences of hops where each element consists of the encrypted identifier of the outgoing link from the domain using a private symmetric key of the domain undertaking the forwarding. This is similar to the concept of Packet-Carried Forwarding State in the SCION architecture [2]. The ESDP and EDSP pair is constructed hop-by-hop as the INIT packet is forwarded along the sequence of domains along the path during flow initialization (steps 2c, 2d and 2e in Fig. 1). In this way HE matching is not required for packets in the data-transfer phase of the flow and neither is per-flow state required to be maintained by any of the domains due to the encrypted path being encoded in the ESDP/EDSPs within the packets themselves.

During flow initialization, once the domain has determined the next hop, it encrypts the egress identifier (which only needs to be locally significant to that domain) with a private symmetric key. It uses a session ID constructed from a hash of the encrypted destination, *hash(Enc_PK(dst), Enc_HE(dst))*, as the initialization vector (IV) for encrypting the next hop. The resulting cyphertext is written as the next element of the ESDP—with a similar operation being done for the next hop in the reverse direction to add an element to the EDSP. By the time the INIT packet reaches the destination host, both the ESDP and the EDSP will be fully constructed.

## 3.4 STEP 3: INIT Reply

Now that destination has fully constructed ESDP and EDSP addresses, it can already send packets to the source using the EDSP. The first packet returned is the INIT-ACK, which is used to send the ESDP to the source for use in subsequent data packets. Note that, as they are fully constructed in step 2, the ESDP and EDSP addresses in the INIT-ACK do not need to be further processed by the domains on the return path.

## 3.5 STEP 4: Data Transfer

The same session ID is carried in all data packets along with a hop counter. When a data packet arrives at the ingress link of a domain, the EPF will use the hop counter as the index into the ESDP/EDSP to retrieve the encrypted next hop, which it will decrypt using its private symmetric key and the session ID as a flow-specific IV to determine the next hop. The IV is used to reduce the probability that flows using the same egress link have the same cyphertext for the next hop in their ESDP/EDSP.

## 3.6 STEP 5: Path Changes

ESDP/EDSPs are constructed in a distributed manner per flow whenever INIT packets are processed. This usually happens once per flow during initialization: all subsequent data packets will use the same ESDP/EDSP. However, if routes change mid-flow then we have several cases to consider.

Firstly, if a route changes, e.g., due to a better path being found, the domain that has updated its route needs to signal the change to all flows using the egress link that the affected route uses, so that source nodes can reinitialize their flows. This will happen by following a similar procedure to step 2, which will cause a new path and a new ESDP/EDSP pair to be calculated by the domains along the path. The flow/session state can be maintained by the endpoints to preserve session and application continuity even though new ESDP/EDSPs are being used. Because we have private routing, and the source and destination addresses are not known by intermediate domains, it is not possible for them to directly signal the source, using ICMP, for example. We propose that the signaling is achieved by setting a flag in the data packets the domain forwards. If these are in the forward direction of the flow then the destination host will maintain the state of the flag in packets it sends in the reverse direction. Whenever a source receives a packet with a route change flag, it can reinitialize the flow. It should be noted that this re-initialization is optional. A source node can choose to maintain the previous—potentially suboptimal—path for the duration of the flow.

Another reason for a domain signaling a re-initialization is if it changes its private symmetric encryption keys, which it may do periodically to increase security. Finally, if a link along the path fails, then the signaling mechanism we suggest will not work due to the failure of connectivity or unavailability of one of the next hops on one or both of the ESDP/EDSP. In this case the source can re-initialize the flow after a timeout on expected incoming data packets or on ACKs on data sent in the forward direction.

## 4 PRELIMINARY EVALUATION

The partial homomorphic Paillier cryptosystem [17] offers better performance than existing fully HE schemes [22], even though it only supports homomorphic modular additions. Moreover, the modified version [15] utilized in this paper exhibits further improved performance thanks to the shift of computation from HE matching to encryption. This is well suited to routing scenarios since it removes the more computationally intensive operations from the routers, eliminating a potential bottleneck in the routing system for deployment of HR on a real network infrastructure.

The performance impact of HR on a real today's network domain was assessed as follows. First, HR was implemented to measure how long it takes for a blinded address to be matched against blinded ranges in forwarding rules. The HE operations to check an address against a rule takes 0.5ms for a key of 2048 bits, when executed on a laptop using a state-of-the-art Apple M1 Pro CPU. Then, to measure the quantity of rules that need to be checked by an EPF, the forwarding table of a major core domain[1] was retrieved. It consists of 142,000 IPv6 rules, which were compressed to 8,140 rules in the FIB by utilizing our routing and HE aggregation scheme. With logarithmic search that equates to around 10 checks on average to determine the next hop for an INIT packet: an overhead of approximately 5ms for a domain to initialize an HR flow.

The above computational overhead is reasonable, especially as this is only required for the INIT packet, which we believe makes HR feasible for deployment today. However, several further optimizations could be put in place. Firstly, dedicated hardware like GPUs [14] could be used. In addition, the Pareto distribution of the popularity of destinations [11] could be considered to organize the forwarding rules in the EPF's FIB. For example, an additional binary tree of the most popular ranges could be checked first by the EPF at forwarding time.

[1] AS AS3257, retrieved from IP address 213.200.64.94

## 5 DEPLOYMENT CONSIDERATIONS

Although HR could be implemented as a clean slate architecture, it is designed to be retro-fitted in current IPv6 and BGP without significant extensions. In this section we look at how the components of HR can be built into the current Internet architecture.

**Inter-domain routing and TE:** The ideas in this paper address inter-domain operations. These operations mirror current BGP with the exception that Network Layer Reachability Information fields in update messages are encrypted. We believe this could be implemented with minor changes to today's BGP protocol to accommodate encrypted ranges rather than IP prefixes.

Most BGP Traffic Engineering (TE) techniques can be used without change: path prepending can be implemented by increasing the AS path length attribute in range updates. MED can be used as it is today: domains can check that blinded ranges they receive over separate links cover the same range but with different MED attribute values that can be used in priority calculations. Hot potato routing to the closest next hop can also be implemented when a tie-break is needed. LOCAL-PREF cannot be handled in precisely the same way because a domain does not know the ranges being advertised. But domains are still able to implement local preferences for identical ranges announced by neighbors on different links.

Anycast routing by BGP [20] remains available. If the same range or sub-range is received by an RC it simply chooses the best one according to its priority calculations on range attributes, independently of its origin, as in today's BGP.

**Intra-domain routing and TE:** The processing of encrypted INIT and data packets introduces some additional processing overhead; the former more than the latter. We do not prescribe how a domain should process HR packets internally but we advise that not all routers in the domain incur this cost. In most cases the ingress router should make inter-domain forwarding decisions on behalf of the domain—possibly with escalation to an SDN controller—and direct the packet to its egress router for the appropriate next hop domain. This could be achieved with tunnels, MPLS, segment routing or extension headers. Most intra-domain TE techniques are applicable with HR since intra-domain routing remains untouched.

**INIT and data packets:** Although the encrypted destination address fits in the source and destination address fields of IPv6 headers in INIT packets, we believe that a better way of incorporating blinded addresses is with an extension header, since this offers more flexibility. Data packets can be implemented with standard IPv6 packets: the ESDP or EDSP can be conveyed in the source and destination address fields and hence 256 bits are available to encode multiple domains and the destination address to be used by the final domain. If more bits are required an extension header will be used. The session ID can be implemented in the flow label field. The hop counter only requires a small number of bits to indicate the index into the ESDP/EDSP which can also by implemented as part of the flow label field.

**HR flows versus TCP/IP flows:** An HR flow is a L3 association between two blinded locators. It does not necessarily map to a single five-tuple TCP/IP flow. Clients can multiplex several TCP/UDP flows on the same HR flow, which could be kept open even if clients temporarily disconnect.

**HEPS implementation:** Our design does not constrain the deployment of the HEPS. Trusted organizations operating as today's Certification Authorities (CAs) and Internet registries are good candidates to implement HEPS functionality. Several organizations could jointly participate in deploying the HEPS, in a similar way to today's Public Key Infrastructures (PKI) and to the way certificate chains are built. Alternatively, if a single organization deployed the HEPS, its components could be distributed for improved performance, scalability and resilience.

Clients need to contact the HEPS only once and can subsequently perform self-blinding of destination addresses for all future sessions. RAs only have to contact the HEPS to generate a blinded and signed version of a new range whenever a domain is assigned new IP addresses. This happens relatively infrequently, in the order of 100s of times per day globally [3]. Cloud techniques leveraging a serverless Function as a Service (FaaS) paradigm [23] could be used to implement the HEPS as a distributed system, whose components can be scaled dynamically according to the number of incoming requests. This would avoid depending upon a single point of failure and prevent bottlenecks that could introduce unnecessary latency.

**Security:** Most of the security infrastructure of the current Internet can remain untouched. However, the lack of visibility of source addresses reduces accountability of senders, which may make Denial-of-Service (DoS) easier. We believe that most current DoS mitigation techniques can be applied. The only change being that those that depend upon blacklisting malicious IP addresses will blacklist malicious ESDPs instead. Additional techniques to mitigate INIT flood-attacks require further investigation.

## 6 RELATED WORK

The most well-known alternatives to HR are VPNs and onion routing systems like ToR [10, 19]. VPNs require extra infrastructure in the network and rely on a trust relationship between the client and the VPN provider. ToR allows for more flexibility in path choice but bypasses provider network decisions and incurs an overhead for all packets in a given flow. Even in solutions where onion routing is implemented at the network layer [8] most of these problems remain.

There have been several proposals in the literature to enhance privacy in routing. Some protect the privacy of BGP announcements [1, 13] but not the data path itself. HE was also used by [4, 18] but in the context of small IoT network's rather than in the wide area Internet. Other alternative privacy-enhancing proposals include [7, 21] but with none of the advantages of HE for native L3 hop-by-hop routing across multiple network domains.

We are not the first to propose removing source addresses from IP packets [6]. Hop-by-hop private routing as we have proposed for HR could be complemented with privacy-focused source routing [24], which increases application or user choice of paths but reduces the scope of TE by network providers.

The use of INIT packets for HR flow establishment borrows ideas from RSVP [5] and connection-oriented protocols like ATM [9] but our system does not need per-flow state to be established in the routers.

## 7 CONCLUSIONS AND FUTURE WORK

HR enables private communications over the Internet without revealing the addresses of flow endpoints to intermediate network domains or to eavesdroppers. While many current solutions to flow privacy are based on overlays, HR is a hop-by-hop L3 solution that allows network providers to remain in control of routing policies and TE strategies. HR does not require per-flow state to be maintained in routers and we have shown that although HE has a computational overhead the overall performance impact of encrypted routing and forwarding is not significant. Our intention with this paper is to initiate a wider discussion on how flow privacy can be implemented and deployed in the Internet. Future work includes investigating options for HE matching operations in firmware or dedicated hardware; and privacy-preserving name resolution techniques to reduce the possibility of the exposure of intended communications endpoints prior to flow establishment.

## ACKNOWLEDGEMENTS

The authors would like to acknowledge the support of Huawei Technologies Co., Ltd. Francesco Tusa is partially supported by the HARPOCRATES EU research project (No. 101069535).

## REFERENCES

[1] Gilad Asharov, Daniel Demmler, Michael Schapira, Thomas Schneider, Gil Segev, Scott Shenker, and Michael Zohner. 2017. Privacy-Preserving Interdomain Routing at Internet Scale. *Proceedings on Privacy Enhancing Technologies* 2017 (07 2017).

[2] David Barrera, Raphael M. Reischuk, Pawel Szalachowski, and Adrian Perrig. 2015. SCION Five Years Later: Revisiting Scalability, Control, and Isolation on Next-Generation Networks.

[3] The APNIC Blog. 2023. BGP in 2022 – the routing table. "https://blog.apnic.net/2023/01/06/bgp-in-2022-the-routing-table/". Accessed: June 2023.

[4] Carlos Borrego, Marica Amadeo, Antonella Molinaro, and Rutvij H. Jhaveri. 2019. Privacy-Preserving Forwarding Using Homomorphic Encryption for Information-Centric Wireless Ad Hoc Networks. *IEEE Communications Letters* 23, 10 (2019), 1708–1711.

[5] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. 1997. Resource ReSerVation Protocol (RSVP).

[6] M. B. Braun and J. Crowcroft. 2014. *SNA: Sourceless Network Architecture*. Technical Report. University of Cambridge, Computer Laboratory.

[7] C. Chen and Perrig Adrian. 2017. PHI: Path-Hidden Lightweight Anonymity Protocol at Network Layer. *Proceedings on Privacy Enhancing Technologies* 2017 (2017), 100 – 117.

[8] C. Chen, D. E. Asoni, D. Barrera, G. Danezis, and A. Perrig. 2015. Hornet: High-speed onion routing at the network layer. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. 1441–1454.

[9] Martin De Prycker. 1993. *Asynchronous Transfer Mode. Solutions for Broadband ISDN*. Prentice Hall.

[10] R. Dingledine, N. Mathewson, and P. Syverson. 2004. *TOR: The second generation onion router – Naval Research Lab Washington DC, Tech. Rep*. Technical Report.

[11] Marwan Fayed, Lorenz Bauer, Vasileios Giotsas, et al. 2021. The Ties That Un-Bind: Decoupling IP from Web Services and Sockets for Robust Addressing Agility at CDN-Scale. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference* (Virtual Event, USA). Association for Computing Machinery, 433–446.

[12] V. Fuller and T. Li. 2006. *RFC 4632 - Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan*. Technical Report.

[13] Debayan Gupta, Aaron Segal, Aurojit Panda, Gil Segev, Michael Schapira, Joan Feigenbaum, Jenifer Rexford, and Scott Shenker. 2012. A new approach to interdomain routing based on secure multi-party computation. *Proceedings of the 11th ACM Workshop on Hot Topics in Networks, HotNets-11*, 37–42.

[14] Toufique Morshed, Md Momin Al Aziz, and Noman Mohammed. 2020. CPU and GPU Accelerated Fully Homomorphic Encryption.

[15] Mohamed Nabeel, Stefan Appel, Elisa Bertino, and Alejandro Buchmann. 2013. Privacy Preserving Context Aware Publish Subscribe Systems. In *Network and*

*System Security*, Javier Lopez, Xinyi Huang, and Ravi Sandhu (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 465–478.

[16] Mohamed Nabeel, Ning Shang, and Elisa Bertino. 2012. Efficient privacy preserving content based publish subscribe systems. *Proceedings of ACM Symposium on Access Control Models and Technologies, SACMAT* (06 2012).

[17] Pascal Paillier. 1999. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In *Advances in Cryptology — EUROCRYPT '99*, Jacques Stern (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 223–238.

[18] Palmieri, Calderoni P., L., and D. Maio. 2017. An Anonymous InterNetwork Routing Protocol for the Internet of Things. *Journal of CyberSecurity and Mobility* 6, 2 (2017), 127–146.

[19] The Tor Project. 2022. The Tor Project – Privacy and Anonymity Online. "https://www.torproject.org". Accessed: June 2023.

[20] Sandeep Sarat, Vasileios Pappas, and Andreas Terzis. 2006. On the Use of Anycast in DNS. In *Proceedings of 15th International Conference on Computer Communications and Networks*. 71–78.

[21] F. Shirazi, M. Simeonovski, M. R. Asghar, M. Backes, and C. Diaz. 2018. A survey on routing in anonymous communication protocols. *ACM Computing Surveys (CSUR)* 51, 3 (2018), 1–39.

[22] Vasily Sidorov, Ethan Yi Fan Wei, and Wee Keong Ng. 2022. Comprehensive Performance Analysis of Homomorphic Cryptosystems for Practical Data Processing.

[23] Vikram Sreekanti, Chenggang Wu, Xiayue Lin, Johann Schleier-Smith, Joseph Gonzalez, Joseph Hellerstein, and Alexey Tumanov. 2020. Cloudburst: stateful functions-as-a-service. *Proceedings of the VLDB Endowment* 13 (08 2020), 2438–2452.

[24] Francesco Tusa, David Griffin, and Miguel Rio. 2021. Private Routing in the Internet. In *2021 IEEE 22nd International Conference on High Performance Switching and Routing (HPSR)*. 1–6.