Friends or Rivals: Insights from Integrating HIP and i3

Andrei Gurtov, Anthony D. Joseph[*]
Helsinki Institute for Information Technology

November 1, 2004

Abstract

The Host Identity Protocol (HIP) uses cryptographic host identities to provide secure and efficient end-to-end communication without requiring a distributed key authority. However, HIP hosts can be vulnerable to DoS attacks and require some infrastructure to support simultaneous mobility of end points. The Internet Indirection Infrastructure (i3) overlay network can be used to provide these desirable properties for HIP control packets. However, with the introduction of network shortcuts in i3 where two hosts can communicate directly, a question arises as to whether i3 can completely replace HIP. Is the end-to-end security provided by HIP a strong enough benefit compared to using shortcuts in i3? Is it worthwhile to consider using a general Distributed Object Location and Routing (DOLR) or Distributed Hash Table (DHT), such as Tapestry or Chord, instead of i3 as a control plane for HIP? We discuss these questions in the paper. We also present implementation experiences with HIP-i3 integration and show initial performance results comparing the throughput of i3 and HIP.

1 Introduction

The proposal for Host Identity Indirection Infrastructure (Hi3) [5] recommends using the Internet Indirection Infrastructure (i3) [8] to relay Host Identity Protocol [6] handshake packets, thereby serving as a control plane. In this paper, we compare the pros and cons of using plain i3, Hi3, and HIP with a DHT. We use the following evaluation criteria:

- Denial of Service (DoS) resistance,
- efficient routing,
- support for mobility,
- infrastructure cost,
- end-to-end security and privacy,
- accountability,
- trust model,
- fault-tolerance,
- management overhead, and
- inverse mapping support.

The basic HIP protocol provides efficient and secure end-to-end connectivity. If the Host Identity Tags (HIT) and IP addresses of end points are known, it can work without additional infrastructure, thus having no issues with infrastructure cost, accountability, trust, management overhead, or fault-tolerance. Basic HIP provides limited DoS protection by making the initiator solve a computationally substantial puzzle before creating state in the responder. Mobility of one end point at a time is supported, but there is no way to perform the reverse mapping support. HIP with a rendezvous server enables mobility of both end points, while preserving accountability and the trust model, since the rendezvous server is chosen by the responder.

The advantages of i3 include better DoS protection, support for simultaneous mobility, and higher fault-tolerance when using a DHT with data replication. Disadvantages of i3 include reliance on an extensive infrastructure, server scalability, use of UDP, and lack of traffic encryption. Since i3 is an overlay network on top of the Chord DHT, it makes the infrastructure fairly complex. There is limited experience with widespread i3 deployment, thus it is difficult to assess how scalable the servers are. The latency of relayed control

[*] Anthony is a Nokia Visiting Fellow currently on sabbatical from the University of California, Berkeley.

traffic will mostly be affected by forwarding and network delays. However, relaying all control and data traffic through i3 infrastructure would likely prove burdensome. By mutual agreement, the client and the server could use i3 only for initial contact and afterwards exchange the data directly. Alternatively, both control and data traffic latency issues could be partially addressed by the use of network *shortcuts* [4]. Although the i3 implementation could run on both UDP and TCP, currently only UDP is supported because maintaining many TCP connection between servers is challenging. As a consequence, some i3 features may be disabled in the wide area, because UDP packets often do not traverse through firewalls and Network Address Translation devices[1].

The basic i3 system does not provide data encryption, although it could be implemented as an add-on feature. i3 also lacks encryption and privacy for control packets. When a public infrastructure is used, i3's extensive infrastructure requirements bring other serious security issues including the possibility of malicious or misbehaving i3 nodes that do not forward correctly and a lack of trust of arbitrary i3 servers from end points. Note that Secure-i3 [1] introduced several constraints on the structure of triggers to prevent misuse of triggers by third parties and formation of loops in the topology. Finally, diagnosing problems in a distributed Internet system is always challenging, and the added indirection introduced by i3 further complicates the situation.

2 Integrating HIP and i3

A combination approach helps address some of the separate shortcomings of HIP and i3. The advantages of using i3 as a control plane for HIP in Hi3 include protection from DoS attacks, solving the double-jump problem, and providing an initial rendezvous service. By hiding parties' IP addresses until the HIP handshake partially authenticates them Hi3 provides additional protection against DoS attacks. Although some DoS protection could be provided by a HIP rendezvous server, the client's IP address is revealed to a server in the first control packet. Simultaneous mobility of both hosts in i3 is supported by sending update control packets via i3 when end-to-end connectivity is lost. Hi3 inherits the challenges of the extensive i3 infrastructure, including trust, accountability, and cost issues.

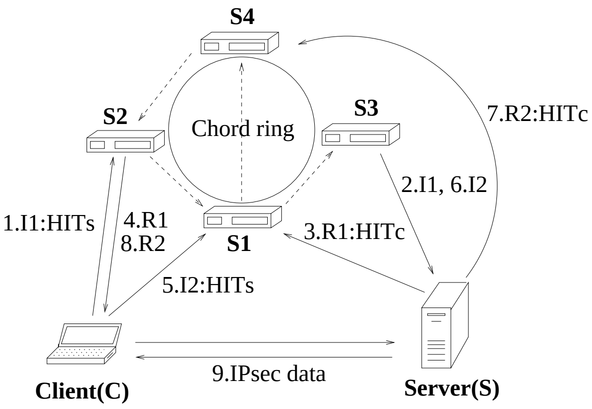

Figure 1: The Hi3 protocol and trigger structure.

Figure 1 shows the setup of HIP connections in Hi3. The client C sends an I1 packet to the address of a random i3 server it happens to have, S2 in this case. The public trigger for the server's S HIT (HITs) is stored in S1 server, and the packet is forwarded from S2 to S1 via Chord. The client obtains the correct i3 server for future contacts to the recipient server, S1. For security, the server S also registers a private trigger that happens to reside on server S3. Therefore, S1 forwards I packets to S3 that in turn delivers them to S. A similar procedure is followed by S to send an R1 reply packet to C. C first contacts S1 that informs it on the correct location of C's public trigger, S4. From S4 packets are forwarded to C's private trigger on S2. The consequent I2-R2 exchange occurs in a similar manner, except that packets are sent straight to i3 servers keeping the public triggers, S1 and S4 respectively. When the base HIP exchange is completed, further communication occurs directly between C and S using IPsec payload.

We used a Boeing Linux HIP implementation [3] to create an Hi3 prototype. Our in-house HIP implementation for the Linux kernel [2] supports only IPv6 at the moment, although an IPv4 port is forthcoming. The Hi3 prototype currently supports only the basic HIP exchange over i3; more advanced Hi3 features such as location privacy using in-path middle boxes and multicast have not yet been implemented. The implementation is 500 lines of code and took about a month to complete. In the implementation, HIP control packets including the IP header are tunneled through i3 servers. IP addresses are not yet hidden in the prototype; some effort is required to fix difficulties with accepting HIP packets with changing IP addresses in the HIP implementation.

[1] To be fair, we note that not all firewalls block UDP traffic and many NAT devices will support reverse flows of UDP traffic. This problem equally affects HIP if IPsec traffic is not forwarded.

In a preliminary evaluation, we compared Hi3 with i3 without shortcuts. The client and server reside in a single LAN and use i3 servers on PlanetLab. TCP throughput between the client and the proxy over i3 (measured with `ttcp` running using i3 proxies [4]) was 80 KByte/second, and the round-trip time (measured with `ping`) was 280 ms. With Hi3, TCP throughput was 5 MByte/second with the round-trip time below 1 ms. These results are heavily influenced by lack of i3 shortcuts, and by the fact that all i3 servers were located in the US.

More extensive wide area evaluation would require access to i3 shortcuts code, root access to hosts outside of Europe (to run an i3 proxy and HIP daemon), and adding a European node to the set of i3 servers on PlanetLab. We have requested these actions from the i3 authors.

We have also experimented with a single locally running i3 server collocated with a data receiver. In this case, throughput was 10 MByte/s and the round-trip time was below 1 ms. A likely reason for higher throughput than with HIP is the data encryption overhead with IPsec. However, even with a local i3 server, the host may experience long round-trip times in i3, since its triggers could be still located on a distant node.

3 Resolving HITs with a DHT

A different approach to a complex Hi3 system would be to use a DHT just to provide the mapping between HITs and IP addresses. The main functionality would be forwarding lookups for a set of IP addresses for a given HIT. A reverse lookup from an IP address to HIT (similar to reverse DNS) provides additional functionality, for example, for security purposes.

From among many existing DHTs [9, 10] we have to choose one that is best suited for this purpose. The main criteria to consider include short key search time and short update time. Network churn is relatively less important, because we expect the resolution infrastructure to be provided by a stable set of servers. Availability of a stable open-source implementation is important. Furthermore, we need a data-storing DHT that can return not just the location of the data, but also the data itself, a set of IP addresses in our case. The Bamboo DHT [7] appears to meet the criteria well. We are planning to develop a prototype for a HIT resolution service based on Bamboo running on PlanetLab.

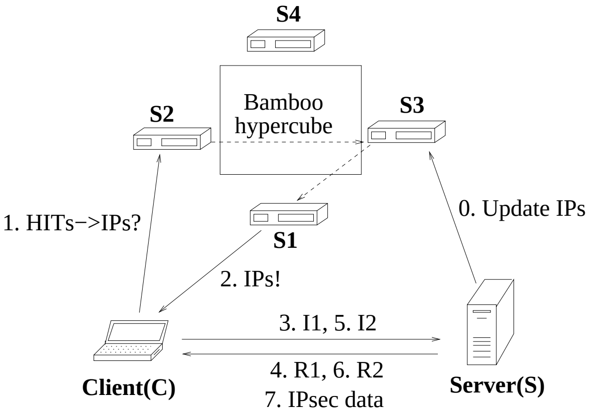

Figure 2: Resolution of a HIT to IP addresses using a DHT.

Figure 2 shows how HIP could use a DHT for resolving a HIT to IP addresses. Before the HIP exchange between the client C and the server S can happen, the client has to discover the current set of IP addresses where S could be reached. C sends a query to a random DHT server it knows about, S2. The query is routed internally in the DHT to the node responsible for the part of address space including HITs, S1. That node sends the IP addresses of S to C. S can change its IP address set by sending an update request to any of DHT servers that will route it to S1. The update should be secured by authenticating the infrastructure server and S with public keys. Furthermore, as with i3, due to malicious or misbehaving nodes, the infrastructure may not always be trusted to route the user traffic correctly.

The advantage of using a DHT instead of i3 is reduced complexity and hence higher scalability of the infrastructure servers. However, additional DoS protection is lost because hosts communicate directly with each other. In addition, whether a DHT can provide sufficiently short update times to support fast mobility is an open question. This question is also valid for i3 with or without shortcuts, since i3 uses a Chord DHT underneath. A related problem that needs more attention is determining when to timeout a transmission and fallback to full routing, and does that impact the performance of interactive (e.g., VoIP, chat) applications.

In addition to IP addresses, a DHT can return a public key of a HIP host, which would allow the initiator to verify the host's HIT for security purposes, such as logging. The returned public key may not always be trusted, but it makes an attack more difficult because it requires cooperation between a malicious host and the infrastructure.

4 Conclusions

In this short paper, we compared the use of plain i3, Hi3, and HIP with a DHT. Hi3 combines best capabilities of i3 and HIP. The efficient and secure end-to-end data plane is inherited from HIP, while the control plane from i3 provides DoS protection, initial rendezvous, and simultaneous mobility of end points. Our initial performance measurements show significant improvements in throughput and latency of Hi3 over plain i3. We plan a more extensive evaluation including i3 shortcuts and running HIP hosts on PlanetLab. Authors hope that HIP and i3 could be used as complementary systems.

We are also studying the use of a DHT for resolution between HITs and IP addresses. We have identified the Bamboo DHT as a suitable candidate and a prototype implementation is ongoing. The use of DHTs offers more simplicity over i3, but they lack the advanced capabilities of i3, such as DoS protection.

5 Acknowledgments

Authors thank Tom Henderson, Ion Stoica, Karthik Lakshminarayanan, Miika Komu, and Pekka Nikander.

References

[1] D. Adkins, K. Lakshminarayanan, A. Perrig, and I. Stoica. Towards a more functional and secure network infrastructure. Technical Report UCB/CSD-03-1242, UCB, 2003.

[2] C. Candolin, M. Komu, M. Kousa, and J. Lundberg. An implementation of HIP for Linux. In *Proc. of the Linux Symposium*, July 2003.

[3] T. Henderson, J. Ahrenholz, and J. Kim. Experience with the Host Identity Protocol for secure host mobility and multihoming. In *Proc. of the IEEE Wireless Communications and Networking Conference (WCNC'03)*, Mar. 2003.

[4] J. Kannan, A. Kubota, K. Lakshminarayanan, I. Stoica, and K. Wehrle. Supporting legacy applications over i3. Technical Report UCB/CSD-04-1342, UCB, May 2004.

[5] P. Nikander, J. Arkko, and B. Ohlman. Host Identity Indirection Infrastructure (Hi3). In *Proc. of The Second Swedish National Computer Networking Workshop 2004 (SNCNW2004)*, Nov. 2004.

[6] P. Nikander, J. Ylitalo, and J. Wall. Integrating security, mobility, and multi-homing in a HIP way. In *Proc. of Network and Distributed Systems Security Symposium (NDSS'03)*, Feb. 2003.

[7] S. Rhea, D. Geels, T. Roscoe, and J. Kubiatowicz. Handling churn in a DHT. In *Proc. of the USENIX Annual Technical Conference*, June 2004.

[8] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana. Internet indirection infrastructure. In *Proc. of ACM SIGCOMM'02*, Aug. 2002.

[9] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proc. of ACM SIGCOMM'01*, Aug. 2001.

[10] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. Kubiatowicz. Tapestry: A resilient global-scale overlay for service deployment. *IEEE Journal on Selected Areas in Communications*, 22(1):41–53, Jan. 2004.