Anonymity in XIA

Nicolas Feltman, David Naylor

December 11, 2011

1 Introduction

The desire for anonymous communication in the Internet has made it a much-discussed issue in both technical and non-technical circles. Users have identified several reasons for wanting online anonymity; in [2], interviews revealed motivations ranging from protecting personal safety to hiding political activity from governments to unknown fears. In one shocking interview from this study, a Romanian woman describes being abducted after posting personal information about herself on a Web site.

Several standard approaches to anonymity have emerged, normally involving some number of explicitly identified proxies (e.g., Tor). Although none of these methods are perfect, they generally meet most users' needs from a technical standpoint. Where many fall short, however, is in making these systems easy to understand and configure for non-technical users; part of our project addresses this issue.

Our goal is three-fold. First, we consider existing methods for achieving anonymous communication in the context of the eXpressive Internet Architecture (XIA) [1]. As we discuss below, novel features of XIA in many cases allow these existing techniques to be implemented more elegantly. Second, we introduce an extension to the XIA socket API, which provides application developers a dead-easy way to use anonymous communication over XIA. Finally, we consider users. We create a preference pane providing OS level control over the extended API functionality. Anonymization settings are controllable by the user through an intuitive, easily understood GUI.

The rest of this paper is organized as follows: in Section 2 we introduce pertinent features of XIA and review common terminology related to anonymity from the literature. Section 3 examines the use of existing approaches in XIA as well as new ones made possible novel features of the architecture. In Section 4 we compare these approaches to one another in terms of the threats against which they provide protection and we compare the "current" version of each method to its XIA counterpart. Finally, Section 5 presents our implementation and in Section 6 we summarize and conclude.

2 Background

Before we discuss anonymity in XIA, we will first discuss XIA itself to highlight the features we will make use of later. Then, we will bring some precision to the term "anonymity" by presenting a review of the terminology used in previous work.

2.1 XIA

Some features of XIA have implications when it comes to anonymity. We briefly describe the key ideas here and elaborate on their impact on anonymous communication later.

2.1.1 Principal-Based Communication

In contrast with today's host-based Internet, XIA provides a framework for communication among *principals*. The idea of principal-based communication is that users should be able to address packets directly

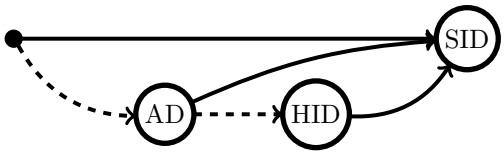


Figure 1: A typical DAG with two fallbacks. Note that higher edges have higher priority; SID is the primary intent, so the network attempts to route it first. If a router does not know the location of SID, or does not understand services, it uses the first fallback and instead routes the packet to SID's AD. Once in the AD, if a router cannot find SID, the packet is routed to HID, the second fallback, before finally being delivered to the service.

to their primary *intent*. For example, a user searching for books on Amazon wishes to communicate with `www.amazon.com`; he doesn't care which particular Amazon server responds to his request.

In this example, `www.amazon.com` can be viewed as a *service* principal. In the case where communication with a particular machine truly is the intent, traditional host-based communication can still be achieved via the *host* principal. Other principals include static *content* (e.g., images) and *autonomous domains* (similar to today's autonomous systems).

2.1.2 DAG-Based Addressing

Addresses in XIA are represented as DAGs (directed acyclic graphs). Using DAGs allows senders to give the network very detailed instructions about how a packet should be routed. DAGs also allow senders to provide *fallback* routes to be used in case the network cannot find the sender's primary intent or does not understand a new principal type. For example, in the scenario above, the user's browser might include the address of a particular Amazon server as a backup in case a packet reaches a router that doesn't know how to find the service `www.amazon.com` directly.

2.1.3 Intrinsic Security

All addresses (or, better put, identifiers) in XIA are *intrinsically secure*; exactly what this means varies among principals. For example, a content identifier (CID) is the cryptographic hash of the content itself, enabling anyone receiving the content to verify its integrity. Hosts and services are required to have a public/private key pair; therefore their corresponding identifiers (HIDs and SIDs) are simply the hashes of public keys. A host can sign any communication it generates with its private key and anyone can publicly verify the signature using the host's ID.

2.2 Anonymity

We adopt terminology proposed by Pfitzmann and Köhntopp [3] to precisely describe various meanings of the term *anonymity*:

Anonymity The state of not being identifiable within a set of subjects, the *anonymity set*.

Unlinkability Two or more items (e.g., subjects, messages, events, actions, etc.) are no more and no less related than they are related to any other item.

Sender Anonymity A particular message is not linkable to any sender and no message is linkable to a particular sender.

Recipient Anonymity A particular message cannot be linked to any recipient and no message is linkable to a particular recipient.

Unobservability The state of messages being indistinguishable from no messages at all.

3 Approach

3.1 Proxies

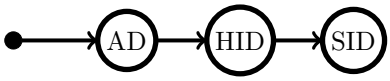
Nico

NOTE to Nico: you should say something about how we're proposing XIA adopt some form of partial DAG encryption, because later I say we talked about it in this section...

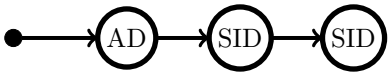
3.2 Temporary Source Addresses

A conceptually simpler approach to achieving anonymity is to use a temporary source address that is unsinkable to you. In today's IP Internet there isn't really a good way to do this. If IP addresses are statically assigned, periodically changing your address is a hassle and may require the involvement of your network administrator, who is unlikely to be willing to assign you a new address every time you want a new identity. Even if addresses are assigned dynamically and you obtain a new one from a DHCP server as often as you'd like, this solution still has major drawbacks. First, most users won't be able to do this from home. A user with a typical home Wi-Fi network can change his computer's IP address to his heart's content, but as his packets pass through his router's NAT, the source address will be overwritten with the public IP assigned to him by his ISP. Further limiting the utility of this approach, your IP address is the source of all traffic you send from your machine, regardless of the application; in some cases, you might want messages from one application to be unlinkable to messages sent by another.

Fortunately, XIA makes the use of temporary source addresses much more feasible. First, consider what normally happens when an XIA application initiates communication with some remote entity. When a host first connects to the network, it contacts a DHCP-like server in its local AD and registers its host ID. Then, when the application opens a socket the operating system will assign the socket an ephemeral service ID (akin to ephemeral ports today). Any packets sent through this socket are given a source address formed form the combination of the host's HID and this ephemeral SID:



Now, suppose we want to use a temporary source address to achieve anonymity. Upon creating the socket, the operating system generates an ephemeral SID, as before, only now it takes the additional step of registering it with the local AD as if it were another host ID via the same mechanism it used to register its actual HID when the machine joined the network. Now packets can be sent with this source address:



Note that we have overcome all of the limitations of the temporary address approach in IP: first, since the operating system can simply generate a new public/private key pair and use it to derive a new SID, there is no need to manually obtain a valid available IP address. Second, as XIA uses 160 bit addresses and is in no danger of running out, there will presumably be no need for NATs, so there is no concern that a router will rewrite our temporary source address and compromise our anonymity. Finally, we no longer have the coarse granularity problem temporary IP addresses posed; each *socket* gets its own temporary address.

Of course, this strategy is not entirely without limitations. Clearly, anyone using this technique must trust his or her local AD not to reveal the mapping between machine and temporary ID; this may not always be reasonable. On the other hand, this can be looked at as a feature rather than a bug. If an AD maintains a log the SIDs registered with it, law enforcement agencies could potentially access them with a subpoena if criminal activity is suspected.

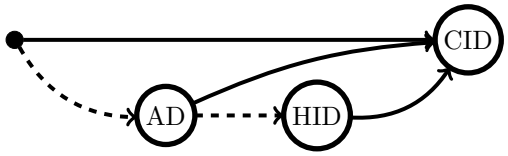
3.3 Principal-Based Traffic Control

A novel approach possible only in XIA is principal-based traffic control. Broadly speaking, a user may to prevent applications from sending packets that are addressed using certain principal types, as the use of a particular principal reveals something about the communication.

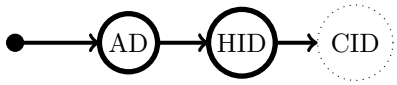
For instance, the presence of a CID makes it clear that a packet's purpose is to retrieve a piece of content. Furthermore, it is conceivable that someone might implement a "reverse lookup" service that does the opposite of the naming service: translates XIDs back into human-readable names. Thus, anyone sniffing our privacy-seeking users's traffic who sees a content request can find out the name of the content from the CID.

Additionally, as new principal types are added to the network in the future, they may similarly reveal information users would prefer to keep private.

In the current XIA design, the operating system would simply drop any packets making use of a disallowed principal (and optionally alert the user). But if a mechanism for partial DAG encryption were adopted, as discussed in §3.1, the system could rewrite DAGs in a way that conceals the use of a particular principal. For example, consider a content request with the following DAG:



Rather than dropping this packet, the DAG could instead be rewritten as follows, where the dotted line indicates that the CID node is encrypted:



4 Comparison

4.1 Services vs. Features

4.2 Threats vs. "Meatures"

5 Implementation

We implemented a single proxy and a version of the API.

5.1 Single Proxy DAG Manipulation

Nico

5.2 In-Network Services Issues

Nico

5.3 OS Integration and API

David

6 Conclusion

References

[1] Ashok Anand, Fahad Dogar, Dongsu Han, Boyan Li, Hyeontaek Lim, Michel Machadoy, Wenfei Wu, Aditya Akella, David Andersen, John Byersy, Srinivasan Seshan, and Peter Steenkiste. XIA: An architecture for an evolvable and trustworthy internet. Technical Report CMU-CS-11-100, Department of Computer Science, Carnegie Mellon University, February 2011.

[2] Ruogu Kang, Sara Kiesler, Peter Kinnaird, Colleen Stuart, and Laura Dabbish. Perceptions about anonymity on the internet. 2011.

[3] Andreas Pfitzmann and Marit Köhntopp. Anonymity, unobservability, and pseudonymity — a proposal for terminology. In Hannes Federrath, editor, *Designing Privacy Enhancing Technologies*, volume 2009 of *Lecture Notes in Computer Science*, pages 1–9. Springer Berlin / Heidelberg, 2001.