Onion Routing for Anonymous and Private Internet Connections

David Goldschlag* Michael Reed† Paul Syverson†

January 28, 1999

1 Introduction

Preserving privacy means not only hiding the content of messages, but also hiding who is talking to whom (traffic analysis). Much like a physical envelope, the simple application of cryptography within a packet-switched network hides the messages being sent, but can reveal who is talking to whom, and how often. Onion Routing is a general purpose infrastructure for private communication over a public network [8, 9, 4]. It provides anonymous connections that are strongly resistant to both eavesdropping and traffic analysis. The connections are bidirectional, near real-time, and can be used for both connection-based and connectionless traffic. Onion Routing interfaces with off the shelf software and systems through specialized proxies, making it easy to integrate into existing systems. Prototypes have been running since July 1997. As of this article's publication, the prototype network is processing more than 1 million Web connections per month from more than six thousand IP addresses in twenty countries and in all six main top level domains. [7]

Onion Routing operates by dynamically building anonymous connections within a network of real-time Chaum Mixes [3]. A Mix is a store and forward device that accepts a number of fixed-length messages from numerous sources, performs cryptographic transformations on the messages, and then forwards the messages to the next destination in a random order. A single Mix makes tracking of a particular message either by specific bit-pattern, size, or ordering with respect to other messages difficult. By routing through numerous Mixes in the network, determining who is talking to whom becomes even more difficult. Onion Routing's network of core onion-routers (Mixes) is distributed, fault-tolerant, and under the control of multiple administrative domains, so no single onion-router can bring down the network or compromise a user's privacy, and cooperation between compromised onion-routers is thereby confounded.

*Divx, Herndon, VA USA, david.goldschlag@divx.com

†Center for High Assurance Computer Systems, Naval Research Laboratory, Washington, DC USA, {reed, syverson}@itd.nrl.navy.mil

2 Application Support via Proxies

Onion Routing can be used with applications that are proxy-aware, as well as several non-proxy-aware applications, without modification to the applications. Currently supported protocols include HTTP, FTP, SMTP, rlogin, telnet, NNTP, finger, whois, and raw sockets. Proxies are under development for Socks5, DNS, NFS, IRC, HTTPS, SSH, and Virtual Private Networks (VPNs). A proxy has three logical layers: an optional application specific privacy filter that sanitizes the data streams; an application specific proxy that translates the data streams into an application independent format accepted by the Onion Routing network; and lastly, an onion proxy that builds and manages the anonymous connections. Because it builds and manages the anonymous connections, the onion proxy is the most trusted component in the system. Likewise, to build onions and hence define routes the onion proxy must know enough of the topology and link state of the network, the public certificates of nodes in the network, and the exit policies of nodes in the network. This information is distributed securely within the network automatically as new nodes come on-line or as the information changes.

3 Moving Data through the Network

Onion Routing's anonymous connections are protocol independent and exist in three phases: connection setup, data movement, and connection tear-down. Setup begins when the initiator creates an onion, which defines the path of the connection through the network. An onion is a (recursively) layered data structure that specifies properties of the connection at each point along the route, e.g. cryptographic control information such as the different symmetric cryptographic algorithms and keys used during the data movement phase. Each onion router along the route uses its public key to decrypt the entire onion that it receives. This operation exposes the cryptographic control information, the identity of the next onion router, and the embedded onion. The onion router pads the embedded onion to maintain a fixed size, and sends it to the next onion router. After the connection is established, data can be sent in both directions. Data from the initiator is repeatedly pre-encrypted using the algorithms and keys that were specified in the onion. As data moves through the anonymous connection, each onion-router removes one layer of encryption as defined by the cryptographic control information in the onion defining the route, so the data arrives as plaintext at the recipient. This layering occurs in the reverse order (using different algorithms and keys) for data moving backward. Connection tear-down can be initiated by either end, or in the middle if needed.

All information (onions, data, and network control) are sent through the Onion Routing network in uniform-sized cells. All cells arriving at an onion-router within a fixed time interval are mixed together to reduce correlation by network insiders. Likewise, the long-standing connections between onion-routers can be padded and bandwidth-limited to foil external observers. An Onion looks different to each onion-router along a connection because of the layered public-key cryptography. Similarly, the layering of symmetric cryptography over the data phase cells makes them appear different to each onion-router. This design resists traffic analysis more effectively than any other deployed mechanisms for Internet communication.

4 Overhead

Onion Routing's overhead is relatively small. Connection setup overhead is typically much less than one second and appears to be no more noticeable than other delays associated with normal web connection setup on the Internet. Computationally expensive public-key cryptography is used only during this connection setup phase. Also, because public key decryption is much more expensive than encryption, the public key burden is mainly placed upon the onion routers themselves, where dedicated hardware acceleration can be justified. The data movement phase uses only secret-key (symmetric) cryptography, which is much faster. Furthermore, since the symmetric encryption can be pipelined, data throughput can be made as fast as ordinary link or end-to-end encryption. Data latency is affected by the number of onion-routers along the connection and can vary with route length and the duration of the Mix cycles.

5 Network Architectures that Shift Trust

Proxies, onion-routers, and other components can be run in a variety of distributed configurations. This allows Onion Routing to mesh well with a wide variety of operational and policy environments. At one extreme, proxies can run remotely. If a user makes a secure connection (e.g., encrytped or withing a firewall) to a trusted remote proxy, Onion Routing's protection can be utilized without installing any software or inducing local computational overhead. At the other extreme, all trusted components can run locally, providing maximum protection of anonymity and privacy against non-local components, even those participating in a connection. In between these two extremes are multiple configurations of proxies and onion routers, running on enclave firewalls or at ISPs.

By shifting trust in this way, Onion Routing can also complement other services like the Anonymizer [1] and LPWA [6]. The Anonymizer uses a central, trusted intermediary to provide sender anonymity (i.e., hide the identity of the sender from the receiver). If Onion Routing is used for privacy, an Anonymizer can run as a filtering proxy on the user's desktop (or the enclave firewall, or the user's ISP) to add sender anonymity. Security is improved because the filtering executes on a machine the user trusts, and communication leaving that machine will resist traffic analysis. Such security in depth removes the central point of failure for network traffic anonymity. LPWA provides various psuedonymy-based services (described elsewhere in this issue). Like Onion Routing it is designed to handle email in addition to HTTP. And, like Onion Routing, it can be configured so that trusted functions are performed at various locations [2]. However, communication between and from these points is not itself anonymous or resistant to traffic analysis. This makes LPWA and Onion Routing especially natural complements.

6 Extensions

A natural extension to Onion Routing is the introduction of reply onions. Reply onions allow connections to be made back to an anonymous sender through the Onion Routing network long after the original connection existed. Reply Onions could be used to send anonymous replies in response to a previously received anonymous email. They could also enable novel applications such as anonymous publishing (anonymous URLs) similar to the Rewebber project [5].

7 Conclusion

In summary, Onion Routing is a traffic analysis resistant infrastructure that is easily accessible, has low overhead, can protect a wide variety of applications, and is flexible enough to adapt to various network environments and security needs. The system is highly extensible, allowing for additional symmetric cryptographic algorithms, proxies, or routing algorithms with only minor modifications to the existing code base. Instructions for accessing the Onion Routing network can be found on our web page along with additional background, pointers to publications, and contact information [7].

References

[1] The Anonymizer. `http://www.anonymizer.com/`

[2] D. Bleichenbacher, E. Gabber, P. Gibbons, Y. Matias, and A. Mayer. “On Secure and Pseudonymous Client-Relationships with Multiple Servers”, to appear in *Proc.* 3^{rd} *USENIX Electronic Commerce Workshop* , August 1998.

[3] D. Chaum. “Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms”, *Communications of the ACM*, v. 24, n. 2, Feb. 1981, pp. 84-88.

[4] D. Goldschlag, M. Reed, P. Syverson. “Hiding Routing Information”, in *Information Hiding*, R. Anderson, ed., LNCS vol. 1174, Springer-Verlag, 1996, pp. 137–150.

[5] I. Goldberg and D. Wagner. “TAZ Servers and the Rewebber Network: Enabling Anonymous Publishing on the World Wide Web”, *First Monday*, vol. 3 no. 4, April 1998.

[6] The Lucent Personalized Web Assistant. `http://lpwa.com/`

[7] The Onion Routing Home Page. `http://www.onion-router.net/`
Conference:

[8] M. Reed, P. Syverson, and D. Goldschlag. "Anonymous Connections and Onion Routing", *IEEE Journal on Selected Areas in Communications*, vol. 16 no. 4, May 1998, pp. 482–494.

[9] P. Syverson, M. Reed, and D. Goldschlag. "Private Web Browsing", *Journal of Computer Security*, vol. 5 no. 3, 1997, pp. 237–248.