Onion Routing for Resistance to Traffic Analysis*

Paul Syverson

Center for High Assurance Computer Systems,
Naval Research Laboratory, Washington, DC, USA
E-mail: syverson@itd.nrl.navy.mil

1 Introduction

Using encryption on a packet-switched network can hide the content of messages, much like placing the messages inside a physical envelope. However, by itself encryption does not hide who is talking to whom, and how often. Onion Routing is a general purpose infrastructure for private communication over a widely shared network such as the internet or the SIPRNET. It provides anonymous connections that are 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. A proof-of-concept prototype for web connections ran at NRL for about two years beginning in 1997, processing more than 1 million web connections per month from more than six thousand IP addresses in twenty countries and in all main top level domains [4]. In the last year the program has been reimplemented from scratch. The resulting codebase is only one third the size of the previous program but at the same time adds several resource management, efficiency, and attack resilience features, which are briefly described below.

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 applications include those for web browsing, email, IRC chat, and remote login (e.g., via SSH).

In the following section we will highlight some of the recent advances made in Onion Routing, followed by a description of how to use Onion Routing to make servers that are resistant both to denial of service and to physical threats. After that, we briefly describe some of the details of how Onion Routing works.

*Some of the material herein appeared previously in [3].

2 Resources, Efficiency, and Robustness

Much of the recent focus of Onion Routing has been on design modifications to make networks more efficient and robust against benign and hostile failures. All Onion Routing connections are via anonymous circuits multiplexed over link connections between onion routers. The bandwidth on the link between onion routers is now throttled. Among other things, this prevents a single onion router from consuming another's entire network bandwidth. We also regulate the flow of data on a single end-to-end circuit by propagating requests for batches of packets back along a route. This controls network congestion and also prevents a single circuit from overwhelming any link. Another addition to the network is router twins. Router twins are simply two or more onion routers that share a private decryption key. This allows a circuit to be established even if an onion router that was designated to be part of that circuit is down or very slow. Finally, we have modified the way application connections correlate to anonymous circuits. Previously, one circuit through the Onion Routing network was made for each application connection initiated at a given proxy. So, for example, every web request required a new Onion Routing circuit. Most of the overhead in Onion Routing is in the setting up of a new circuit. Now, if an application connection request arrives at a proxy, the proxy assigns it a *topic* within an existing anonymous circuit if there is a circuit with space available, otherwise it creates a new anonymous circuit first. In this way the total number of circuits necessary to carry the same application traffic through the system is reduced. Also, the average connection setup overhead for an application connection is reduced. And, attempts to correlate circuit setup with any application connection requests emerging from the network are confounded.

3 Location Protected Servers

We have begun design of a location-protected service infrastructure for survivability and DoS resistance. For exam-

Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX'03)
0-7695-1897-4/03 $17.00 © 2003 IEEE

ple, a large company might have a series of external computers that handle transactions with its customers or suppliers. If these computers are flooded by a network attack, or their building is located and bombed, continued service is now blocked. Akamai solves this problem by purchasing thousands of computers and maintaining them around the world. Our design is a compromise, providing several public computers for transactions in normal conditions, and several computers whose locations are hidden by an anonymous communications network. These hidden computers can continue to provide service to trusted customers or employees in the event of an attack, without revealing their location to anybody — not even those customers or employees. By making the amount of access to these hidden servers proportional to client trust, we can support a variety of "behind the scenes" hierarchical structures for both military and civilian purposes. Since Onion Routing can be overlayed on the same physical infrastructure as existing communications networks, this design protects all the various mission critical servers that can be run on the network against both DoS and physical attack, but without large-scale server redundancy and its associated hardware costs and replication complexity.

4 Onion Routing Overview

Onion Routing operates by dynamically building anonymous circuits within a network of onion routers, similar to real-time Chaum Mixes [1]. 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. A single mix makes it difficult to track a particular message either by specific bit-pattern, size, or ordering with respect to other messages. By routing through numerous mixes in the network, determining who is talking to whom is made even more difficult. Like a mix network, a network of onion routers is designed to be distributed, fault-tolerant, and under the control of multiple administrative domains, so no single onion router, or even small group of cooperating onion routers, can bring down the network. And even cooperating onion routers are confounded in trying to compromise a user's privacy.

Applications interface with the network by means of proxies. A proxy has three logical layers: an optional application specific privacy filter that sanitizes the data streams[1]; 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 circuits. Because this means that it chooses both the route and the keys used for data movement, the onion proxy is the most trusted component in the system.

[1]Privoxy [5] is currently used as an off-the-shelf filter.

4.1 Moving Data through the Network

Onion Routing's anonymous circuits are protocol independent and exist in three phases: circuit setup, data movement, and circuit tear-down. Setup begins when the initiator creates an onion, which defines the path of the circuit through the network. An *onion* is a (recursively) layered data structure that specifies properties of the circuit at each point along the route, e.g., the different symmetric *onion 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 onion keys, 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 circuit is established, data can be sent in both directions. Data from the initiator is repeatedly pre-encrypted using the keys that were specified in the onion. As data moves through the anonymous circuit, each onion router removes one layer of encryption as defined by the onion keys it received in the onion defining the route, so the data arrives as plaintext at the recipient. This layering occurs in the reverse order (using different keys) for data moving backward. Circuit 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. An onion looks different to each onion router along a circuit because of the layered public-key cryptography. Similarly, the layering of symmetric cryptography over the data cells makes them appear different to each onion router. This design resists traffic analysis more effectively than any other deployed mechanisms for general internet communication.

4.2 Overhead

Onion Routing's overhead is relatively small. Connection setup overhead is typically not noticeable in the context of other delays associated with normal web connection setup on the internet. Computationally expensive public-key cryptography is used only during the circuit setup phase. Also, because public-key decryption is much more expensive than encryption, the public-key burden rests mainly upon the onion routers themselves, where the option of dedicated hardware acceleration can be justified. (Our modular design is completely compatible with doing the public-key operations in either hardware or software, and we have used both in our test networks.) Also, the recently introduced innovation, topics, allows several appli-

cation connections to run over a single Onion Routing circuit, thus amortizing and reducing this cost per application connection.

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 circuit and can vary with route length.

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 an encrypted connection 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 circuit. In between these two extremes are multiple configurations of proxies and onion routers, e.g., running on enclave firewalls or at ISPs.

6 Conclusion

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. Additional information about Onion Routing can be found on our web page, [4].

Acknowledgments

Onion Routing has been supported by CNO, DARPA, and ONR. Thanks to Roger Dingledine for helpful comments and input.

References

[1] D. Chaum. "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms", *Communications of the ACM*, v. 24, no. 2, Feb. 1981, pp. 84–88.

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

[3] D. Goldschlag, M. Reed, and P. Syverson. "Onion Routing for Anonymous and Private Internet Connections", *Communications of the ACM*, v. 42, no. 2, Feb. 1999, pp. 39–41.

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

[5] The Privoxy Home Page. `http://www.privoxy.org/`