Introducing Accountability to Onion Routing

Michael Backes*‡, Jeremy Clark†, Peter Druschel‡, Aniket Kate* and Milivoj Simeonovski*
*Saarland University, ‡MPI-SWS, †Carleton University

1 Introduction

Onion routing (OR) protocols provide an overlay network for low-latency anonymous communication [6]. Tor [1], the most widely-deployed onion routing network, serves hundreds of thousands of internet users every day, defending them against network surveillance, censorship, and circumvention. OR protocols achieve anonymity by relaying encrypted TCP streams through a series of independent, geographically-diverse, volunteered *OR nodes*; this obfuscates the true source of the packets in a way that is robust against anyone with access to a partial portion of the network.

While onion routing provides a powerful service to its users, the nature of the technology can sometimes be harmful for the volunteer operators of the OR nodes, more specifically, the exit nodes that serve as the last hop in an OR circuit [2, 3]. Exit nodes route the package to its destination. From the perspective of the destination, and its ISP, the traffic seems to originate from the OR exit node. The fact that the traffic has only been forwarded is not readily apparent. If a user's online communication results in a criminal investigation or a cause of action, the exit node operator may become embroiled in the proceedings, whether as the suspect/defendant or as a third party with evidence (we will henceforth concentrate on criminal actions).

If law enforcement uncovers an IP address related to a criminal act, which is translated to a physical address by an ISP, this in many countries is sufficient for the search and seizure of computer equipment found at the address. From the perspective of law enforcement, there is currently no reliable mechanism to determine that a piece of traffic they are investigating originated from an OR node, short of seizing the computer. The primary contribution of this work is to consider what information an OR exit node can safely log about specific traffic flows and provide to law enforcement, while protecting user anonymity.

1.1 Contribution

1. Exit Node Deniability In this work, we present a novel deniability mechanism called Detra which allows an OR exit node, in cooperation with the network, to issue a cryptographic guarantee that the traffic flows being relayed are indeed on behalf of the OR network, and not originating from the node. To assist in the design of Detra, we propose a concept of pseudonymous signatures, which employ pseudonyms (or half Diffie-Hellman exponents) as temporary public keys (and corresponding temporary secrets) used in the OR circuit construction for signing messages. The pseudonym signatures, in combination with ring signatures formed from the advertised public keys of the online OR routers, allow Detra to provide exit router deniability without hampering the anonymity properties of OR protocols. Further, Detra does not introduce any new infrastructure or external communication, nor does it have an impact on the key-exchange protocol—users utilize their already established values.

2. Selective Traceability While traceability has never been a component of any widely-deployed anonymous communication system, it may become the case that new anonymity networks, or a changing political climate, initiate an interest in providing a complete trace to users who misuse anonymity networks according to laws or terms of service. We show how Detra can be modified to provide a verifiable (backward) traceability mechanism, where selected offending traffic flows output by the network can be traced back to the corresponding user, while maintaining the eventual forward secrecy of the system.

Our mechanisms are generic and can, to the best of our knowledge, be incorporated in known OR networks, remailers, several mix network protocols, and peer-to-peer anonymity networks. In this extended abstract, we cover only the Exit Node Deniability aspects of Detra. For the complete Detra protocol we refer the readers to [4].

2 Exit-Node Deniability

2.1 Preliminaries

An OR infrastructure involves a set of *routers* (or *OR nodes*) that relay traffic, a *directory service* providing status information for OR nodes, and *users*. Users benefit from anonymous access by constructing a *circuit*—a small ordered subset of OR nodes—and routing traffic through it sequentially. The crucial property for anonymity is that an OR node within the built circuit is not able to identify any portion of the circuit other than its predecessor and successor.

Pseudonyms. The authentication challenges employed in all OR circuit construction protocols share the same structure: they are discrete logarithm (DL) exponentiations in some Diffie-Hellman setting. In particular, they can be represented as $\alpha = g^x$, where g is a generator of a cyclic group $\mathbb{G}$ of prime order p and $x \in_R \mathbb{Z}_p$ is a random secret value known only to the user. In the OR literature, these authentication challenges α are known as user *pseudonyms* since they also function as the users' temporary public keys for the recipient OR nodes in key

agreement protocols.

We observe that a pseudonym α and the corresponding secret key x can also be used as a signing key pair in a DL setting. We call signatures that use such $(x, \alpha = g^x)$ as signing key pair *pseudonym signatures.* As pseudonyms are generated independently for every single OR node, and the corresponding secret exponents are random elements of $\mathbb{Z}_p$, they do not reveal the user's identity. Moreover, it also is not possible to link two or more pseudonyms to a single identity. Therefore, pseudonym signatures become particularly useful in our DETRA mechanism, where users utilize them to sign messages without being identified by the verifier.

Ring Signatures. A ring signature scheme enables a signer to sign a message m using all the public keys of a predefined set of signers S and his private key. Consider a set of signers $s_i \in S$. A ring signature for a message m is generated using the public keys of all signers and the private key of any signer from the set. Using such a technique, the verifier can check that message m is signed by a signer within the set S but cannot determine the identity of the signer. In this work, we use an efficient bilinear pairing-based ring signature scheme proposed by Boneh et al. [5].

2.2 The Protocol

We present the exit-node deniability scheme in DETRA. The scheme provides a *cryptographic* proof for exit OR node operators to prove when they output traffic flows from the OR network, as opposed to originating the flow (even while simultaneously operating as a node), without breaking the security properties of the OR protocol.

While maintaining the anonymity, unlinkability and forward secrecy properties of onion routing, for exit-node deniability we also wish to achieve deniability, untraceability and no false accusation. (For the details, we refer to our full paper [4].)

Consider an OR circuit $\langle U \leftrightarrow N_1 \leftrightarrow N_2 \leftrightarrow N_3 \rangle$ of a user U and three OR nodes N_1, N_2, N_3. In the OR protocol pseudonym α_3 for the exit node N_3 is a cryptographic component (generated by the user U) that is known to both N_2 and N_3. Moreover, the pseudonym α_3 and its corresponding secret exponent x_3 can also serve as a signing key pair. DETRA achieves exit-node deniability by combining these characteristics of pseudonym α_3 along with signatures from node N_2. Our deniability scheme works according to the following four steps:

1. **Pseudonym Endorsement:** The middle node N_2, while extending the circuit to N_3, sends the pseudonym α_3 along with its signature on α_3. The exit node N_3 verifies the signature and upon a successful verification, replies to N_2 with an authentication response for the OR key agreement.

   We carefully avoid any conceptual modification of the OR circuit construction protocol; the above signature generation and verification steps are the only adjustments that DETRA makes to this protocol.
2. **Stream Verification:** Once the circuit has been established, the user U can utilize it to send her web stream requests. To open a TCP connection, the user sends a stream request to the exit node N_3 through the circuit. The user U includes a pseudonym signature on the request contents signed with the secret exponent x_3 of α_3. When the stream request reaches the exit node N_3, the exit node verifies the pseudonym signature with α_3. Once the verification is successful, N_3 creates the evidence log (Step 3) and proceeds with the TCP handshake to the destination server. The request is discarded otherwise. This stream verification helps the exit node to prove linkability between its handshakes with the destination server and the pseudonym α_3 it received from the middle node.
3. **Log Generation:** After a successful verification of the stream request, a deniability evidence record for the request is generated. While generating the record, N_3 converts the signature it received from N_2 (on α_3) to a ring signature associated with a larger set S of OR nodes (*e.g.,* a set of all OR nodes other than N_3, those belonging to the country of N_2, *etc.*). DETRA employs these ring signatures to protect the identity of the middle node.The evidence record consists of a signature chain along with its respective messages; *i.e.,* the ring signature on α_3 by a set of OR nodes and a pseudonym signature on stream request contents for the pseudonym α_3.
4. **Deniability Verification:** To check if a malicious stream came out of the OR network and not the exit node N_3, a record originating from N_3 corresponding to the stream request (e.g., IP address, port number, and timestamp) can be used. It can be ensured that the stream indeed came out the OR network by verifying that a member of the set of OR nodes included in the ring signature signed a pseudonym, which in turn signed the stream request. Therefore, deniability verification involving two signature verifications: a pseudonym signature verification and a ring signature verification.

References

[1] Tor Project: Anonymity Online. Online: `https://www.torproject.org/`.

[2] Tor madness reloaded. Online: `http://itnomad.wordpress.com/2007/09/16/tor-madness-reloaded/`, 2007.

[3] Raided for operating a Tor exit node. Online: `http://raided4tor.cryto.net/`, 2012.

[4] M. Backes, J. Clark, P. Druschel, A. Kate, and M. Simeonovski. Introducing accountability to onion routing. `http://crypsys.mmci.uni-saarland.de/Detra.pdf`.

[5] D. Boneh, C. Gentry, B. Lynn, and H. Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In *EUROCRYPT*, 2003.

[6] P. F. Syverson, D. M. Goldschlag, and M. G. Reed. Anonymous connections and onion routing. In *IEEE SSP*, 1997.