BlindBox: Deep Packet Inspection for Encrypted Traffic – Public Review

Michael Walfish
NYU

Today, network administrators use middleboxes to scan and filter traffic. Yet senders and receivers sometimes want to hide their traffic from entities in the path (for example by communicating over an encrypted channel, using HTTPS), which is at odds with middlebox scanning. As the saying goes, we can't have our cake and eat it too. Or can we?

Enter Blindbox. Its authors begin with the intuition that cryptography could be helpful because, in principle, one can compute over encrypted data. But there is a vast gulf between that intuition and a system that solves the problem. The authors cross this gulf by knitting together, for the first time in an implemented system, searchable encryption (in which one party can detect whether the encrypted traffic of another party matches given patterns) and secure two-party computation (by which two parties can agree on the output of a function while keeping their inputs private). The protocols are deft and novel, and require inventiveness at every turn. For example, the authors devise a new technique that I'll call "tripwire decryption" (wherein a middlebox gets to decrypt a flow only when the flow matches a pattern). They develop a clever searchable encryption scheme (which improves on, and owes a debt to, a protocol of Song, Wagner, and Perrig). They do much more besides (constructing a sophisticated cryptographic synthesis is hard; doing it while being compatible with a given problem domain is even harder.)

The authors implement their protocols and produce an HTTPS variant. They apply this protocol to several applications: intrusion detection, parental filtering, and exfiltration prevention. The end result is impressive, and a strong example of theory meeting practice: if the end-points and middlebox are Blindbox-aware, then traffic is hidden from the middlebox, the middlebox's logic is hidden from the end-points, and the middlebox can apply that logic to the encrypted traffic of the sender and receiver. To top off all of this work, Blindbox comes with a proof of security.

Although it pushes the envelope very far, Blindbox is still limited by the underlying cryptographic machinery (which the authors are very clear about.) For example, to support regular expressions, Blindbox requires "flow decryption when a suspicious keyword is observed in the flow"; for some users, this gives the middlebox too much power. Also, setting up connections costs tens of seconds for thousands of patterns. (On the other hand, applying the rules is astonishingly fast, owing in part to the authors' algorithmic innovations and in part to hardware support for AES encryption.) In addition, there is substantial bandwidth overhead for some flows, and the scheme does not yet apply to binary data (video, images, etc.). All of this points to future work in this area.

One lingering question is whether all of this work is worth it. A lot of Blindbox's complexity stems from keeping middlebox rules cryptographically hidden from end-points. The authors explain in Section 2.2.1 of the paper why some parties may prefer this model. However, the program committee (PC) believes that it may be acceptable to expose rules to end-points, as is the case today in some IDS setups (HIDS, etc.). The PC looks forward to the community's discussion of this aspect of the work's motivation.

One answer is that Blindbox is a first step. The authors' larger goal is to retain in-network support for all middleboxes while providing privacy. This is an ambitious and worthwhile goal. Indeed, although the technical work in this paper was reason enough for the PC to accept the paper, what makes the paper even more exciting is the possibilities that it points to.