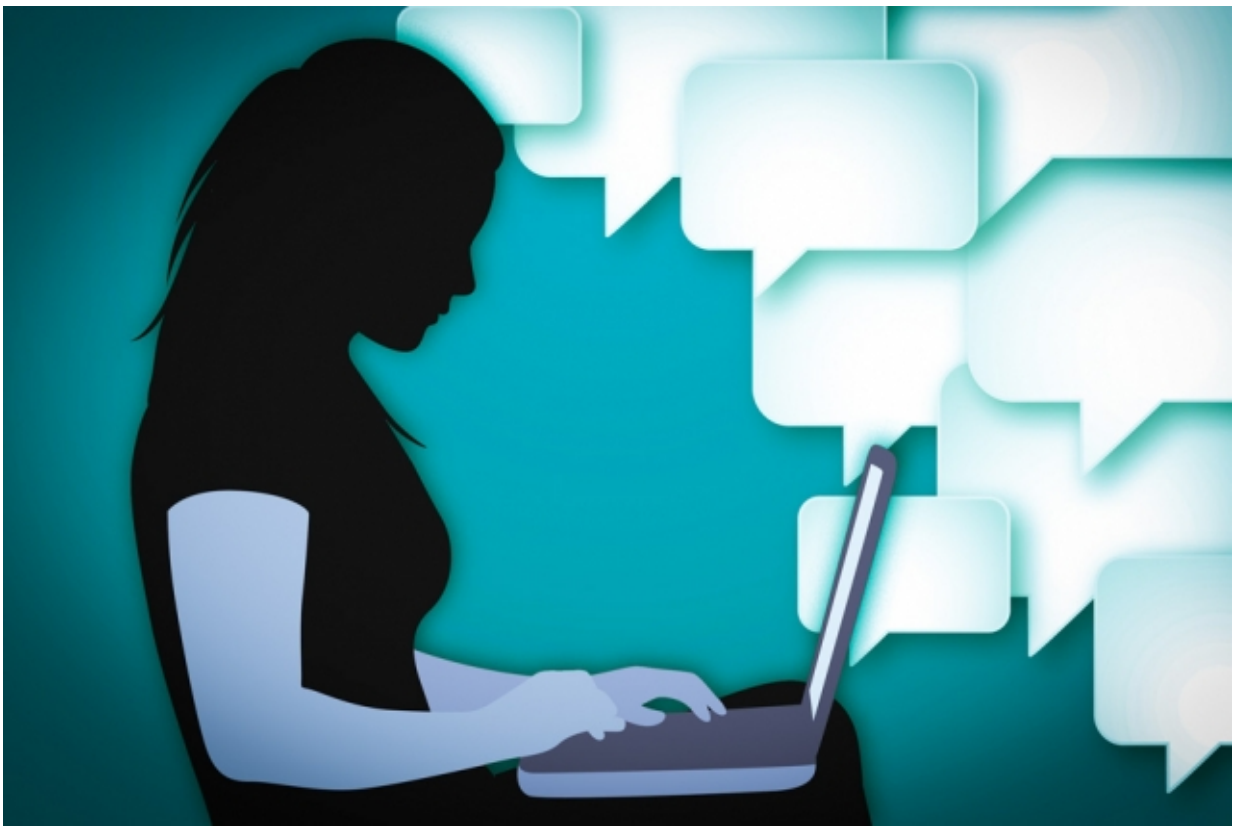
Anonymity network can protect users' identity if all but one of its servers are compromised

July 11 2016, by Larry Hardesty



Researchers at MIT and the École Polytechnique Fédérale de Lausanne have developed a new anonymity scheme that provides strong security guarantees but uses bandwidth much more efficiently than its predecessors. Credit: MIT News

Anonymity networks protect people living under repressive regimes from surveillance of their Internet use. But the recent discovery of vulnerabilities in the most popular of these networks—Tor—has prompted computer scientists to try to come up with more secure anonymity schemes.

At the Privacy Enhancing Technologies Symposium in July, researchers at MIT's Computer Science and Artificial Intelligence Laboratory and the École Polytechnique Fédérale de Lausanne will present a new anonymity scheme that provides strong security guarantees but uses bandwidth much more efficiently than its predecessors. In experiments, the researchers' system required only one-tenth as much time as existing systems to transfer a large file between anonymous users.

"The initial use case that we thought of was to do anonymous file-sharing, where the receiving end and sending end don't know each other," says Albert Kwon, a graduate student in electrical engineering and computer science and first author on the new paper. "The reason is that things like honeypotting"—in which spies offer services through an anonymity network in order to entrap its users—"are a real issue. But we also studied applications in microblogging, something like Twitter, where you want to anonymously broadcast your messages to everyone."

The system devised by Kwon and his coauthors—his advisor, Srini Devadas, the Edwin Sibley Webster Professor of Electrical Engineering and Computer Science at MIT; David Lazar, also a graduate student in electrical engineering and computer science; and Bryan Ford SM '02 PhD '08, an associate professor of computer and communication sciences at the École Polytechnique Fédérale de Lausanne—employs several existing cryptographic techniques but combines them in a novel manner.

Shell game

The heart of the system is a series of servers called a mixnet. Each server permutes the order in which it receives messages before passing them on to the next. If, for instance, messages from senders Alice, Bob, and Carol reach the first server in the order A, B, C, that server would send them to the second server in a different order—say, C, B, A. The second server would permute them before sending them to the third, and so on.

An adversary that had tracked the messages' points of origin would have no idea which was which by the time they exited the last server. It's this reshuffling of the messages that gives the new system its name: Riffle.

Like many anonymity systems, Riffle also uses a technique known as onion encryption; "Tor," for instance, is an acronym for "the onion router." With onion encryption, the sending computer wraps each message in several layers of encryption, using a public-key encryption system like those that safeguard most financial transactions online. Each server in the mixnet removes only one layer of encryption, so that only the last server knows a message's ultimate destination.

A mixnet with onion encryption is effective against a passive adversary, which can only observe network traffic. But it's vulnerable to active adversaries, which can infiltrate servers with their own code. This is not improbable in anonymity networks, where frequently the servers are simply volunteers' Internet-connected computers, loaded with special software.

If, for instance, an adversary that has commandeered a mixnet router wants to determine the destination of a particular message, it could simply replace all the other messages it receives with its own, bound for a single destination. Then it would passively track the one message that doesn't follow its own prespecified route.

Public proof

To thwart message tampering, Riffle uses a technique called a verifiable shuffle. Because of the onion encryption, the messages that each server forwards look nothing like the ones it receives; it has peeled off a layer of encryption. But the encryption can be done in such a way that the server can generate a mathematical proof that the messages it sends are valid manipulations of the ones it receives.

Verifying the proof does require checking it against copies of the messages the server received. So with Riffle, users send their initial messages to not just the first server in the mixnet but all of them, simultaneously. Servers can then independently check for tampering.

Generating and checking proofs is a computationally intensive process, however, which would significantly slow down the network if it had to be repeated with every message. So Riffle uses yet another technique called authentication encryption, which can verify the authenticity of an encrypted message.

Authentication encryption is much more efficient to execute than the verifiable shuffle, but it requires the sender and the receiver to share a private cryptographic key. So Riffle uses the verifiable shuffle only to establish secure connections that let each user and each mixnet server agree upon a key. Then it uses authentication encryption for the remainder of the communication session.

As long as one server in the mixnet remains uncompromised by an adversary, Riffle is cryptographically secure.

"The idea of mixnets has been around for a long time, but unfortunately it's always relied on public-key cryptography and on public-key techniques, and that's been expensive," says Jonathan Katz, director of

the Maryland Cybersecurity Center and a professor of computer science at the University of Maryland. "One of the contributions of this paper is that they showed how to use more efficient symmetric-key techniques to accomplish the same thing. They do one expensive shuffle using known protocols, but then they bootstrap off of that to enable many subsequent shufflings."

"When you use standard encryption on the Internet, you use an expensive public-key crypto system to encrypt a short key, and then you use symmetric-key techniques to encrypt your longer message," Katz adds. "But it's novel in the context of these mixnets. They've been around for 20, 25 years, and nobody has had this insight until now. In the standard context of encryption, you have the honest sender and the honest receiver, and they're defending against an external malicious attacker. Here, you need stronger properties. The issue is that the server that's doing the shuffling might themselves be malicious. So you need a way to ensure that even a malicious server can't shuffle incorrectly."

More information: Albert Kwon et al. Riffle, *Proceedings on Privacy Enhancing Technologies* (2016). DOI: 10.1515/popets-2016-0008 , people.csail.mit.edu/devadas/pubs/riffle.pdf