Vuvuzela: Scalable Private Messaging Resistant to Traffic Analysis

Jelle van den Hooff, David Lazar, Matei Zaharia, and Nickolai Zeldovich
MIT CSAIL

1 Introduction

Many users would like their communications over the Internet to be private, and for some users, such as reporters, lawyers, or whistleblowers, privacy is of paramount concern. Encryption software can hide the *content* of messages, but adversaries can still learn a lot from *metadata*—which users are communicating, at what times they communicate, and so on—by observing message headers or performing traffic analysis. For example, if Bob repeatedly emails a therapist, an adversary might reasonably infer that he is a patient, or if a reporter is communicating with a government employee, that employee might come under suspicion. Recently, officials at the NSA have even stated that "if you have enough metadata you don't really need content" [8: ¶7] and that "we kill people based on metadata" [5]. This suggests that protecting metadata in communication is critical to achieving privacy.

State-of-the-art private messaging systems fall into two broad categories, but neither can protect metadata for large numbers of users. On the one hand are systems that provide strong, provable privacy guarantees, such as Dissent [10] and Riposte [2]. Although these systems can protect metadata, they either rely on broadcasting all messages to all users, or use expensive cryptographic constructions such as Private Information Retrieval (PIR) to trade off computation for bandwidth [9]. As a result, these systems have scaled to just 5,000 users [10] or hundreds of messages per second [2].

On the other hand, scalable systems like Tor, Pond [6], OTR, and mixnets provide little protection against powerful adversaries that can observe and tamper with network traffic. To the extent these systems try to protect metadata, they require a large number of users to provide any degree of privacy, so as to increase the anonymity set for each user, and even then they are susceptible to traffic analysis. Adding cover traffic to try to obscure which pairs of users are communicating has been shown to be expensive and to yield only limited protection against a passive observer over time [3, 7], while adversaries that can actively disrupt traffic (*e.g.,* inject delays) can gain even more information [1].

We propose Vuvuzela, the first system to provide scalable point-to-point text messaging while guaranteeing metadata privacy, achieving orders of magnitude more messages per second than previous systems. Vuvuzela ensures that no adversary will learn which pairs of users are communicating, as long as just one out of N servers is not compromised, even for users who continue to use Vuvuzela for years.[1] Vuvuzela uses only simple, fast cryptographic primitives, and preliminary experiments suggest that it can scale to millions of users and tens of thousands of messages per second.

[1] Vuvuzela cannot hide the fact that a user is connected to Vuvuzela's network, but we expect that users will simply run the Vuvuzela client in the background at all times to avoid revealing the timing of their conversations.

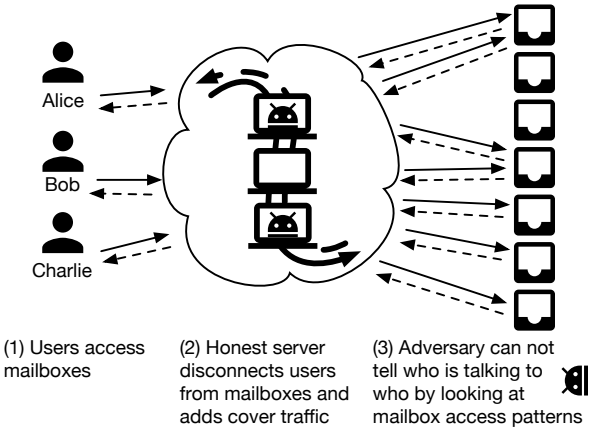

Figure 1: Overview of Vuvuzela's conversation protocol.

Threat model. Vuvuzela's design assumes an adversary that controls all but one of the Vuvuzela servers (users need not know which one), controls an arbitrary number of clients, and can monitor, block, delay, or inject traffic on any network link. Two users communicating through Vuvuzela should have their communication protected if their two clients, and any one server, are uncompromised. We also assume that the Vuvuzela servers' public keys are known to all users, and that two users who wish to communicate know each other's public keys. Separate mechanisms are needed to let users discover each other's keys, but we consider these orthogonal to the private communication problem in this paper.

2 System Design

Vuvuzela works by routing user messages through a chain of servers, as shown in Figure 1, where each of the servers adds cover traffic to mask the communication patterns of users. Unlike prior systems, Vuvuzela's design enables cover traffic to scale to millions of users, and allows us to prove strong guarantees about the level of privacy provided by cover traffic. We achieve this using two key techniques.

First, Vuvuzela's protocols are carefully structured to reveal only a small, well-defined set of observable variables to an adversary. For instance, Vuvuzela's conversation protocol, used for sending user messages, exposes just two variables: the total number of users engaged in a conversation, and the total number of users not engaged in a conversation. It does *not* reveal the users in either group. This is significantly smaller than the number of variables exposed by previous systems, and enables Vuvuzela to focus on minimizing the useful information that an adversary can learn from these variables.

Second, Vuvuzela adopts ideas from differential privacy [4] to state precise privacy guarantees, and to bound information leakage over time by adding noise to the observable variables with cover traffic. Vuvuzela ensures that regardless of whether

any given user is active or not, the value of every observable variable has near equal probability of being observed by an adversary. This means that the adversary cannot learn who, if anyone, a given user is talking to. Somewhat counter-intuitively, the amount of cover traffic required is constant—independent of the number of users or messages—and we find that it is manageable in practice. Adding noise to achieve differential privacy is tractable for the small number of variables exposed by Vuvuzela, but it was not feasible for prior systems that expose many distinct variables.

Communicating via mailboxes. To minimize observable variables, Vuvuzela does not let users communicate directly, but instead uses a mailbox design, where servers never have to initiate connections back to clients. The way two users communicate in this protocol is reminiscent of a "dead drop." Users pick some mailbox as a meeting spot, and perform an *exchange* operation on that mailbox. The exchange operation places a new message in a mailbox, and retrieves whatever message was placed there by another user. If two users perform an exchange on the same mailbox, they receive each others' messages. This protocol repeats in *rounds*, which we expect to be on the order of tens of seconds. Thus, if Alice and Bob want to communicate, then each round, each of them exchanges the message they want to send (if any) with the mailbox, and each will receive the other's message as a result.

Masking observable variables. The above mailbox approach forms the basis of our protocol, but still allows an adversary to observe three sets of variables, which we briefly describe how to eliminate or mask:

1. *Which users participated in the protocol each round.* To eliminate this variable, all users always perform an exchange with a mailbox, even if they are not in an active conversation.
2. *Which mailbox each user accessed.* To eliminate the observable connection between the sender of a message, the mailbox that the message is placed in, and the eventual recipient of the message, the Vuvuzela servers form a mixnet that shuffles the messages (Figure 1). As long as one server is uncompromised, adversaries cannot link each message to a mailbox. To ensure that the adversary cannot learn anything from the IDs of the mailboxes accessed, each pair of communicating clients chooses a pseudo-random mailbox ID each round, using a shared secret based on their public keys and the round number.
3. *How many mailboxes were accessed by two users.* While it may seem that unlinking users from mailboxes is enough to achieve privacy, an adversary can still learn something from statistics about messages exchanged. E.g., if the adversary suspects that Alice and Bob are talking, she can try knocking Bob offline and seeing whether the total number of mailboxes with two accesses decreases. To mask this variable, each server adds *cover traffic* messages to random mailboxes (some paired, some not). We use results from differential privacy to set the amount of traffic so that, if at least one server is honest, its traffic is enough to mask the activity of any single user.

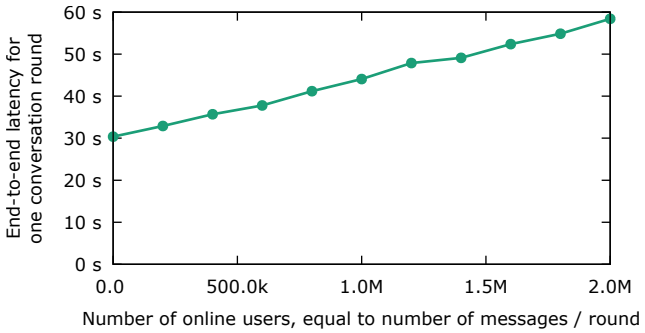

Figure 2: Performance of Vuvuzela's conversation protocol with three servers. Every user sends a message every round; two-thirds of them are active conversations, and one-third is not (but each of these inactive users still sends a message to a random mailbox).

3 Preliminary Results

To understand whether this design is viable, we implemented a prototype of Vuvuzela, and ran it on Amazon's 36-core EC2 servers (since our prototype is CPU-heavy). With 1 million simulated users, we achieved a throughput of 15,000 messages per second and a latency of 44 seconds, as shown in Figure 2.

To achieve these results, the amount of cover traffic we needed to add was equivalent to about 300,000 users per Vuvuzela server, or half the traffic in this case. However, this amount is constant regardless of the number of users, allowing Vuvuzela to scale further. We can further increase throughput by using multiple physical machines per Vuvuzela server.

We believe these results are encouraging, since they indicate Vuvuzela can scale to a reasonable number of users, and its latency may be acceptable for email-like messaging or chat.

References

[1] A. Back, U. Möller, and A. Stiglic. Traffic analysis attacks and trade-offs in anonymity providing systems. In *Proc. of the Workshop on Information Hiding*, pages 245–257, Pittsburgh, PA, Apr. 2001.

[2] H. Corrigan-Gibbs, D. Boneh, and D. Mazières. Riposte: An anonymous messaging system handling millions of users. In *Proc. of the 36th IEEE Security and Privacy*, San Jose, CA, May 2015.

[3] G. Danezis. Measuring anonymity: a few thoughts and a differentially private bound. In *Proc. of the DIMACS Workshop on Measuring Anonymity*, May 2013.

[4] C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9 (3-4):211–407, 2014.

[5] M. Hayden. The price of privacy: Re-evaluating the NSA. Johns Hopkins Foreign Affairs Symposium, Apr. 2014. https://www.youtube.com/watch?v=kV2HDM86XgI&t=17m50s.

[6] A. Langley. Pond, 2015. https://pond.imperialviolet.org/.

[7] N. Mathewson and R. Dingledine. Practical traffic analysis: Extending and resisting statistical disclosure. In *Proc. of the Privacy Enhancing Technologies Workshop*, pages 17–34, May 2004.

[8] A. Rusbridger. The Snowden leaks and the public. *The New York Review of Books*, Nov. 2013.

[9] L. Sassaman, B. Cohen, and N. Mathewson. The Pynchon gate: A secure method of pseudonymous mail retrieval. In *Proc. of the Workshop on Privacy in the Electronic Society*, Arlington, VA, Nov. 2005.

[10] D. I. Wolinsky, H. Corrigan-Gibbs, B. Ford, and A. Johnson. Dissent in numbers: Making strong anonymity scale. In *Proc. of the 10th OSDI*, Hollywood, CA, Oct. 2012.