Yodel: Strong Metadata Security for Real-Time Voice Calls

David Lazar and Yossi Gilad and Nickolai Zeldovich

MIT CSAIL

Abstract

Yodel is the first system for real-time voice calls that hides metadata (e.g., who is communicating with whom) from a powerful adversary that controls the network and compromises servers. Voice calls require sub-second message latency, but low latency has been difficult to achieve in prior work where processing each message requires an expensive public key operation at each hop in the network. Yodel avoids this expense with the idea of *self-healing circuits*, reusable paths through a mix network that use only fast symmetric cryptography. Once created, these circuits are resilient to passive and active attacks from global adversaries. Creating and connecting to these circuits without leaking metadata is another challenge that Yodel addresses with the idea of *guarded circuit exchange*, where each user creates a backup circuit for the case an attacker tampers with their traffic. We evaluate Yodel across the Internet and it achieves acceptable voice quality with 970 ms of latency for 4.8 million active users.

1 Introduction

Telecom providers retain call records which include the participants and duration of every call. This metadata is used for mass surveillance [9]; for example, the NSA collected 434 million call records of Americans in 2018 [10]. Call metadata is especially problematic for journalists who need to keep their sources confidential [4]. Even if telecoms stop retaining call records, an attacker can monitor or compromise the network to learn about voice calls happening in real-time.

Metadata-hiding systems have made significant progress in scaling up to a large number of users [1, 5, 7, 8, 11, 12, 14], but none of them can support real-time voice communication at that scale. Voice communication requires relatively low latency (a second or two at most) and relatively high bandwidth (a few kilobits per second), which are both at least an order of magnitude beyond what state-of-the-art systems can support. For example, the recent Karaoke [8] design achieves 8 seconds of latency for 4M users with 0.24 kbits/sec of bandwidth for each user. The fastest state-of-the-art systems also leak a small amount of metadata with every message, which is reasonable for text messaging but not well suited to the high message rate of real-time voice calls.

This talk presents Yodel, the first metadata-hiding system for real-time voice communication that defends against a strong adversary that compromises the entire network and might compromise any server. The closest prior metadata-private voice communication system, Herd [3], assumes an adversary that monitors the network but does not focus on compromised servers; other systems based on Tor are vulnerable to traffic analysis attacks by a network adversary [2]. Yodel hides metadata by operating a set of servers that form a mixnet to shuffle user messages. Yodel allows users to set up voice calls with one another, but relies on another (metadata-private) service for the dialing handshake (i.e., notifying a user that someone wants to talk and responding to the call).

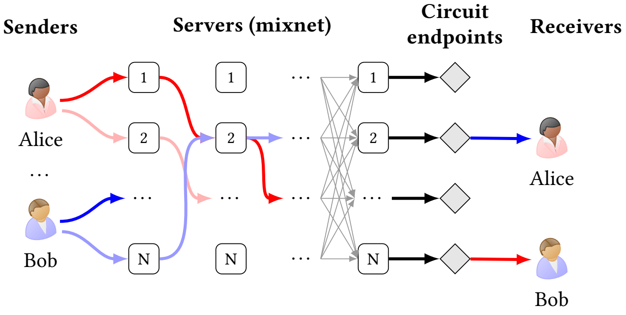

Figure 1: Overview of Yodel's components. Alice and Bob have created two circuits each. The faded arrows are backup circuits, created as part of Yodel's guarded circuit exchange. Alice and Bob are in a voice call, so they are listening on each other's circuits.

Figure 1 shows how users communicate through Yodel at a high level. Users send messages directly to a Yodel server which participates in a mix network with the other servers. The users choose a random sequence of servers to process each of their messages and onion encrypt their messages to ensure that messages follow their chosen paths. An established path through the network is called a *circuit*, and messages (e.g., voice packets) flow from users through circuits to their *endpoints*. Users receive messages by listening on a circuit endpoint, which is a pseudorandom ID that reveals nothing about the sender.

Yodel's servers, labeled 1 through N in Figure 1, shuffle messages to hide which user is sending to which circuit endpoint. The servers shuffle messages in *layers*, indicated by the vertical groups, similar to a parallel mixnet [6, 8]. All paths in Yodel have the same number of layers, which is a system security parameter. At each layer, a server receives messages from all of the servers in the previous layer, decrypts the messages (which are onion-encrypted), shuffles them, and sends the messages to the servers on the next layer. To simplify Figure 1, the server-to-server communication is only shown for the next-to-last layer.

In Figure 1, Alice and Bob are in a voice call and have established two circuits through Yodel, but each of them only listens on one circuit. Alice is sending messages to the circuit endpoint that Bob is listening on, and vice-versa. The adversary sees that Alice and Bob connect to the system, and knows to which circuit endpoints they are listening. However, the mixnet hides which users are sending to which circuit endpoint, so the adversary cannot tell whether Bob is listening on Alice's circuit.

Yodel achieves high performance by addressing the costly public-key cryptographic operations that are typically required in a mixnet. For example, previous systems [1, 6–8, 12, 14, 15] require public-key operations for every message, which becomes a performance bottleneck. Yodel uses a symmetric key circuit through the mixnet to relay messages between two users. Users set up the circuit using public-key cryptography, but individual messages sent over the circuit benefit from low-cost symmetric-key cryptography. Although circuits offer high performance, using them securely required Yodel to address two technical challenges.

The first challenge lies in the fact that circuits are used for multiple messages. Since servers maintain shared keys with each user for the duration of a circuit, a server may be able to learn information about a user over time. For example, if a user is briefly disconnected from the network, a server might observe that no message arrived on a particular circuit, and infer that the circuit belongs to that user. Yodel's key insight is the idea of *self-healing circuits*, which use honest servers to ensure that circuit traffic is maintained despite network interruptions, such as a user's network going offline, or an active attack on any part of the network.

The second challenge lies in connecting to circuits without revealing metadata. In Yodel, if Alice wants to call Bob, Alice sets up a circuit and tells Bob to connect to that circuit to receive Alice's messages. If Alice is not talking to anyone, she connects to her own circuit as a form of cover traffic. However, suppose that Alice calls Bob and doesn't hear back. Alice will connect to her own circuit for cover traffic. Bob might also connect to Alice's circuit because he got the call but the adversary dropped his reply. Now the adversary will observe both Alice and Bob connecting to the same circuit, leaking that they wanted to communicate.

Yodel addresses this challenge using *guarded circuit exchange*, a simple protocol that ensures two honest users never connect to the same circuit. The insight is to have each user establish two circuits: one as a circuit for talking with a buddy and another as a fallback for cover traffic. In Figure 1 the bright red arrow is Alice's circuit she created to chat with Bob, and the faded red arrow is her fallback circuit. In case of any message loss during dialing, each user can safely connect to *either* their cover traffic circuit or the buddy's circuit, without leaking any metadata to the adversary.

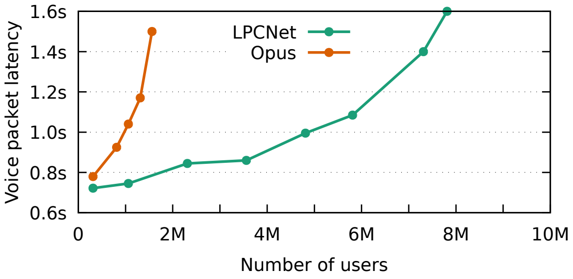

Figure 2: One-way latency for voice packets with a varying number of users and 100 Yodel servers.

2 Implementation

We implemented a prototype of Yodel in Go and ran it on 100 servers across different countries in Europe and North America to evaluate its performance. We evaluated Yodel with two voice codecs: the standard Opus codec and LPCNet [13], a low-bitrate vocoder. Figure 2 shows the preliminary results. The results show that Yodel with LPCNet provides real-time voice communication with a latency of 970 ms from the time a user sends a message to the time their buddy receives it, while supporting 4.8 million users. We find that Yodel provides acceptable voice quality, and we have communicated over Yodel several times to discuss the design of the system.

The full paper describing Yodel and its source code will be available soon at https://vuvuzela.io.

References

[1] S. Angel and S. Setty. Unobservable communication over fully untrusted infrastructure. In *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pages 551–569, Savannah, GA, Nov. 2016.

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

[3] S. L. Blond, D. R. Choffnes, W. Caldwell, P. Druschel, and N. Merritt. Herd: A scalable, traffic analysis resistant anonymity network for VoIP systems. In *Proceedings of the 2015 ACM SIGCOMM*, pages 639–652, London, United Kingdom, Aug. 2015.

[4] S. Humphreys and M. de Zwart. Data retention, journalist freedoms and whistleblowers. *Media International Australia*, 165(1):103–116, 2017. URL https://doi.org/10.1177/1329878X17701846.

[5] A. Kwon, D. Lazar, S. Devadas, and B. Ford. Riffle: An efficient communication system with strong anonymity. In *Proceedings of the 16th Privacy Enhancing Technologies Symposium*, Darmstadt, Germany, July 2016.

[6] A. Kwon, H. Corrigan-Gibbs, S. Devadas, and B. Ford. Atom: Horizontally scaling strong anonymity. In *Proceedings of the 26th ACM Symposium on Operating Systems Principles (SOSP)*, pages 406–422, Shanghai, China, Oct. 2017.

[7] A. Kwon, D. Lu, and S. Devadas. XRD: scalable messaging system with cryptographic privacy. *CoRR*, abs/1901.04368, 2019. URL http://arxiv.org/abs/1901.04368.

[8] D. Lazar, Y. Gilad, and N. Zeldovich. Karaoke: Distributed private messaging immune to passive traffic analysis. In *Proceedings of the 13th USENIX Symposium on Operating Systems*

Design and Implementation (OSDI), pages 711–726, Carlsbad, CA, Oct. 2018.

[9] J. Mayer, P. Mutchler, and J. C. Mitchell. Evaluating the privacy properties of telephone metadata. *Proceedings of the National Academy of Sciences (PNAS)*, 113(20):5536–5541, 2016.

[10] O. of the Director of National Intelligence. Statistical transparency report regarding use of national security authorities (calendar year 2018), Apr. 2019. URL https://www.dni.gov/files/CLPT/documents/2019_ASTR_for_CY2018.pdf.

[11] A. M. Piotrowska, J. Hayes, T. Elahi, S. Meiser, and G. Danezis. The Loopix anonymity system. In *Proceedings of the 26th USENIX Security Symposium*, pages 1199–1216, Vancouver, Canada, Aug. 2017.

[12] N. Tyagi, Y. Gilad, D. Leung, M. Zaharia, and N. Zeldovich. Stadium: A distributed metadata-private messaging system. In *Proceedings of the 26th ACM Symposium on Operating Systems Principles (SOSP)*, pages 423–440, Shanghai, China, Oct. 2017.

[13] J.-M. Valin and J. Skoglund. A Real-Time Wideband Neural Vocoder at 1.6 kb/s Using LPCNet. *arXiv e-prints*, page arXiv:1903.12087, Mar 2019.

[14] J. van den Hooff, D. Lazar, M. Zaharia, and N. Zeldovich. Vuvuzela: Scalable private messaging resistant to traffic analysis. In *Proceedings of the 25th ACM Symposium on Operating Systems Principles (SOSP)*, pages 137–152, Monterey, CA, Oct. 2015.

[15] D. I. Wolinsky, H. Corrigan-Gibbs, B. Ford, and A. Johnson. Dissent in numbers: Making strong anonymity scale. In *Proceedings of the 10th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, Hollywood, CA, Oct. 2012.