Vuvuzela: Scalable Private Messaging Resistant to Traffic Analysis

Jelle van den Hooff, David Lazar, Matei Zaharia, Nickolai Zeldovich

MIT CSAIL

Symposium on Operating Systems Principles (SOSP), 2015

Motivation

- Encryption systems hide only content of messages
- Protection of metadata is critical for privacy
- Strong, provable privacy guarantees XOR scalability

"If you have enough metadata, you don't really need content." Stewart Baker

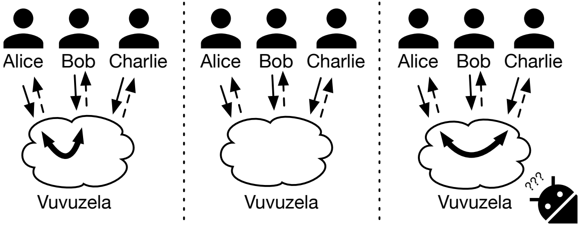
"We kill people based on metadata." Michael Hayden

Vuvuzela — Goals

- Private point-to-point messaging
- Scalable (millions of users, tens of thousands of messages per second)
- Limited amount of information about communication patterns over time
- No availability guarantees

Vuvuzela — Goals

- Private point-to-point messaging
- Scalable (millions of users, tens of thousands of messages per second)
- Limited amount of information about communication patterns over time
- No availability guarantees



Vuvuzela gives Alice differential privacy: any event observed by the adversary has roughly equal probability in all worlds.

Strong, active attacker that. . .

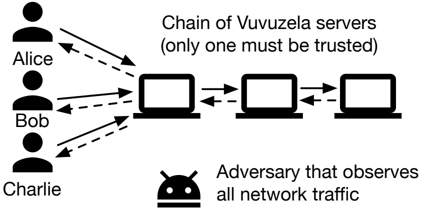
- controls all but one (any) of the Vuvuzela servers
- controls an arbitrary number of clients
- monitors/blocks/delays/injects traffic on any network link

Vuvuzela — Assumptions/Prerequisites

- Standard cryptography: encryption, key exchange, signatures, hashes
- Established public keys for Vuvuzela servers and users (PKI)
- Bug-free implementation

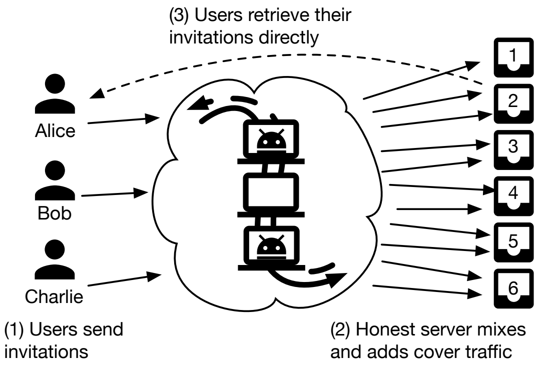
Overview

- Single chain of Vuvuzela servers
  - Users connect to first server
  - Last server hosts dead drops
  - Mix messages and randomly add fakes
- Fixed-rate, fixed-size encrypted messages → fixed number of conversations/client
- Two protocols: dialing + conversion



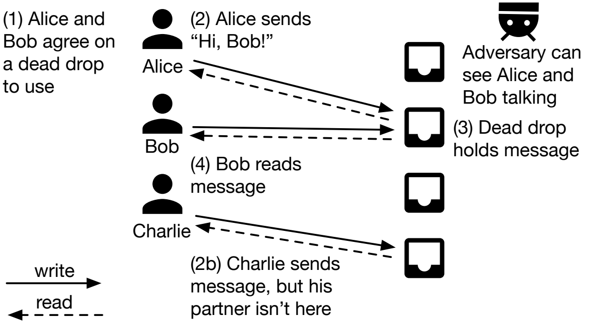
Dialing Protocol

- Dialing round every 10 minutes
- m large invitation dead drops
- Invitation for user with public key pk stored in $H(pk) \mod m$
- Special *no-op* dead drop



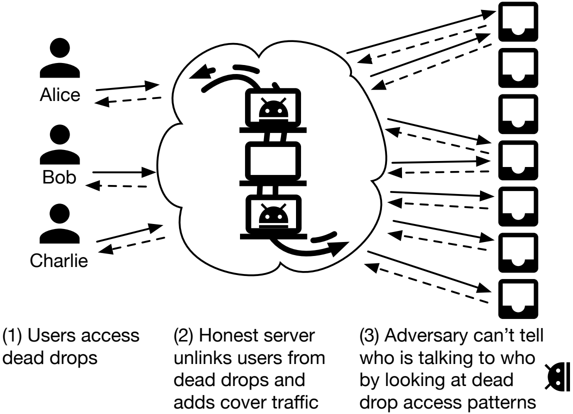
Conversion Protocol (Strawman)

- Synchronous rounds coordinated by first server
- Ephemeral conversation dead drops with 128-bit ID



Conversion Protocol

- Dead drop selected based on shared secret derived from public keys of communication partners and round number



Evaluation

Prototype

- Implemented in Go with $\sim$2700 SLOC
- No CDN- or Torrent-based distribution for dialing protocol

Evaluation

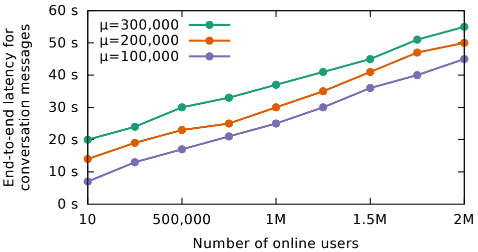
Prototype

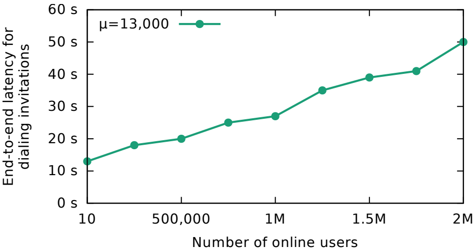
- Implemented in Go with ~2700 SLOC
- No CDN- or Torrent-based distribution for dialing protocol

Setup

- Amazon EC2 virtual servers: 36 Xeon E5-2666 v3, 60 GiB RAM, 10 Gbps network
  - 3 Vuvuzela server
  - 5 servers to simulate clients
  - Dedicated (untrusted) entry server
- Deterministic amount of nose, i.e. number of fake messages
- 80/256 bytes per dialing/conversation message
- 5 % of users dial another user each dialing round
- 100 clients fetch their dialing dead drop + extrapolation of required bandwidth

Results





- 1.2M fake message
- With 1M users and $\mu = 300$k: 37 s end-to-end latency, 68k messages/s, 166 MB/s per server, 10s of seconds per round
- Limiting factor: 340k Curve25519 DH operations per second and server

- 12 kB/s per user
- 12 GB/sec in aggregate

Conclusion

- Private messaging system scalable to millions of users
- Protects against traffic analysis of powerful attacker
- Minimizes observable variables and hides them with noise
- Quantifiable security properties
- High bandwidth demands (especially on servers)