Dissent: Accountable Group Anonymity

Henry Corrigan-Gibbs (http://www.henrycg.com/)
Bryan Ford (http://bford.info/)

Published in 17th ACM Conference on Computer and Communications Security (CCS 2010) (http://www.sigsac.org/ccs/CCS2010/)

Introduction

Dissent (Dining cryptographers Shuffled Send Network) is a protocol for accountable anonymous messaging between members of an ordered and well-defined group. Dissent provides stronger anonymity guarantees than do **Mix networks (http://www.torproject.org/)** and is resistant to the anonymous DoS attacks that can cripple **DC nets (http://www.ece.cmu.edu/~adrian/731-sp04/readings/dcnets.html)**.

Dissent generalizes an anonymous data collection protocol by Justin Brickell and **Vitaly Shmatikov (http://userweb.cs.utexas.edu/~shmat/)**. Dissent allows for decentralized deployment, adds the ability for honest nodes to identify misbehaving participants, and maintains efficiency even when participants transmit messages of radically different lengths. These improvements should make Dissent a practical and deployable solution for anonymous messaging within small decentralized groups.

To explore the practicality of Dissent, we coded up a **prototype (dissent.zip)** in Python and ran it on 40+ **Emulab (http://www.emulab.net/)** nodes. The current prototype is *not* a secure implementation of the protocol and should be used only for evaluating Dissent's performance and other behavior. Nevertheless, our preliminary experiments with the protocol are encouraging. **Read the full paper (dissent.pdf) or slide deck (dissent-slides.pdf) to learn more about Dissent.**

Formal Abstract

Users often wish to participate in online groups anonymously, but misbehaving users may abuse this anonymity to disrupt the group. Messaging protocols such as Mix-nets and DC-nets leave online groups vulnerable to denial-of-service and Sybil attacks, while accountable voting protocols are unusable or inefficient for general anonymous messaging.

We present the first general messaging protocol that offers provable anonymity with accountability for moderate-size groups, and efficiently handles unbalanced loads where few members have much data to transmit in a given round. The *N* group members first cooperatively shuffle an *N* x *N* matrix of pseudorandom seeds, then use these seeds in *N* "pre-planned" DC-nets protocol runs. Each DC-nets run transmits the variable-length bulk data comprising one member's message, using the minimum number of bits required for anonymity under our attack model. The protocol preserves message integrity and one-to-one correspondence between members and messages, makes denial-of-service attacks by members traceable to the culprit, and efficiently handles large and unbalanced message loads. A working prototype demonstrates the protocol's practicality for anonymous messaging in groups of 40+ member nodes.

Downloads

- Paper (**.pdf (dissent.pdf)**), *Errata:* 1 (**.pdf (erratum.pdf)**) 2 (**.pdf (erratum2.pdf)**)
- Code (**.zip (dissent.zip)**, **.tar.gz (dissent.tar.gz)**)
- Slides (**.pdf (dissent-slides.pdf)**, **.pptx (dissent-slides.pptx)**)

Related Links

- **Emulab (http://www.emulab.net)** - The network testbed we used for our implementation.
- **DC Nets (http://www.ece.cmu.edu/~adrian/731-sp04/readings/dcnets.html)** - David Chaum's original description of DC nets.
- **Herbivore (http://www.cs.cornell.edu/People/egs/herbivore/index.html)** - An application of small-scale DC nets to large-scale anonymous communication.
- **PeerReview (http://peerreview.mpi-sws.org/)** - A practical way to deal with silent nodes.
- **Tor (http://www.torproject.org/)** - An implementation of Mix networks over the Internet.
- **WikiLeaks (http://www.wikileaks.org/)** - One potential application for Dissent.

Acknowledgements

We wish to thank **Vitaly Shmatikov (http://www.cs.utexas.edu/~shmat/)**, **Mike Fischer (http://www.cs.yale.edu/~fischer/)**, **Bimal Viswanath (http://www.mpi-sws.org/~bviswana/)**, **Animesh Nandi (http://www.mpi-sws.org/~animesh/)**, Justin Brickell, **Jacob Strauss (http://pdos.csail.mit.edu/jastr/)**, Pedro Fonseca, and the anonymous reviewers for valuable feedback and discussion. This work was

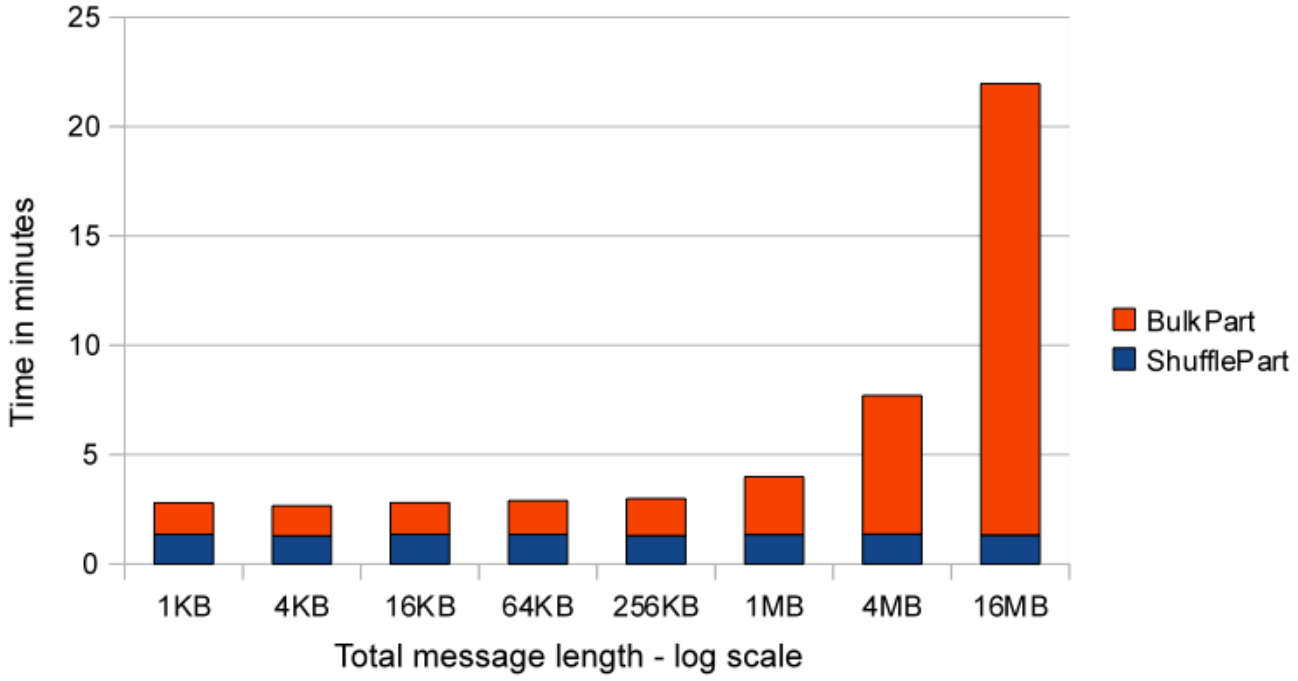
supported by the National Science Foundation (http://www.nsf.gov/) under grant CNS-0916413 (http://www.nsf.gov/awardsearch/showAward.do?AwardNumber=0916413).

Summary of Performance Results

Refer to our paper (dissent.pdf) for details on how we implemented the protocol and conducted these tests.

Phase Time over Message Length

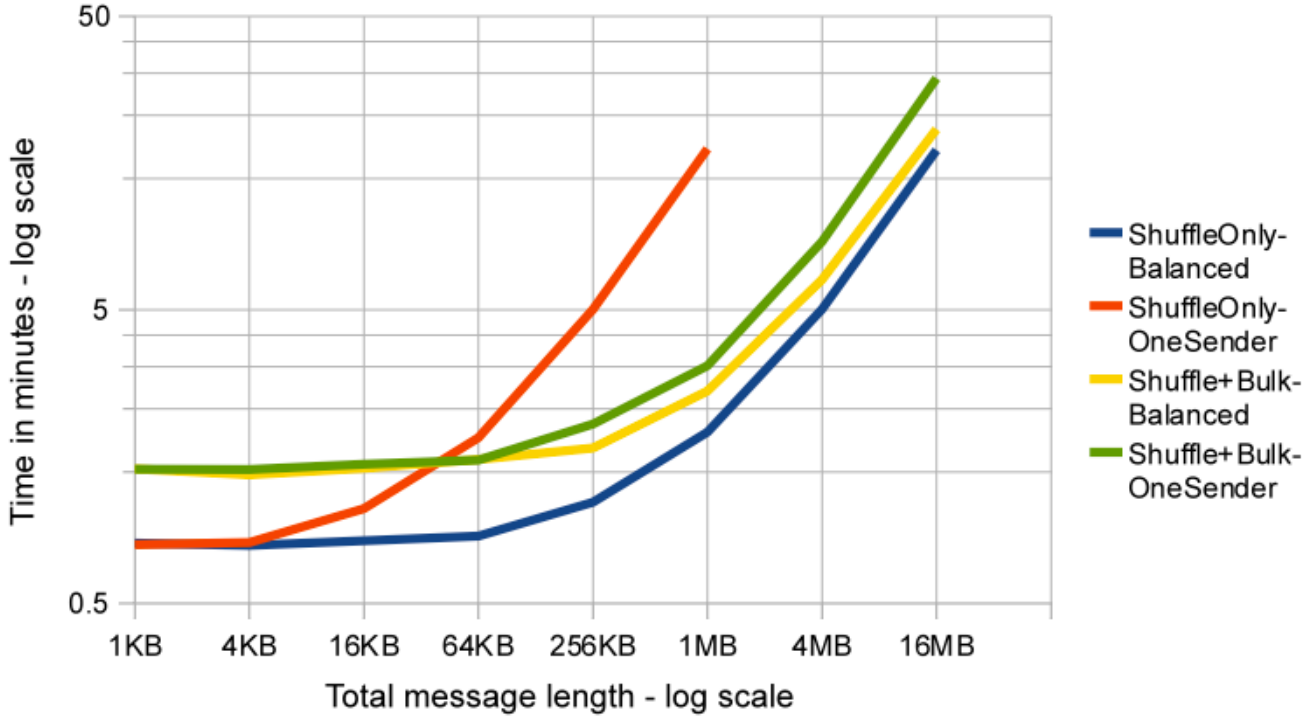
The shuffle phase of the protocol takes a constant amount of time with respect to message size, as expected. The bulk phase of the protocol takes approximately a linear amount of time with respect to message length (note the log scale on the x axis).



Total Time over Message Length

Dissent (Shuffle+Bulk) outperforms the Brickell (ShuffleOnly) protocol when participants send messages of differing lengths. When all participants send messages of equal length, the Brickell protocol outperforms Dissent because of the overhead incurred by Dissent's shuffle set-up phase.

Dissent run slightly slower when only one participant transmits a message because the computational load falls almost entirely on a single node (the sending node).



Total Time over Group Size

The running time grows near-quadratically in the number of participating nodes. We expect a quadratic rate of growth because the size of the message descriptor table contains one row and one column for each node (i.e., it has size $O(N^2)$).