Using Dust Clouds to Enhance Anonymous Communication

Richard Mortier[1], Anil Madhavapeddy[2], Theodore Hong[2], Derek Murray[2], and Malte Schwarzkopf[2]

[1] Horizon Digital Economy Research
Sir Colin Campbell Building, Triumph Road
Nottingham NG7 2TU, UK
richard.mortier@nottingham.ac.uk
[2] University of Cambridge
15 JJ Thomson Avenue
Cambridge CB3 0FD, UK
firstname.lastname@cl.cam.ac.uk

1 Introduction

Cloud computing platforms, such as Amazon EC2 [1], enable customers to lease several *virtual machines* (VMs) on a per-hour basis. The customer can now obtain a dynamic and diverse collection of machines spread across the world. In this paper we consider how this aspect of cloud computing can facilitate anonymous communications over untrusted networks such as the Internet, and discuss some of the challenges that arise as a result.

Most anonymous networks act as a *mix network*, creating hard-to-trace communications by using chains of proxy servers. For example, Mixmaster[3] is an anonymous remailer based on the mix-net protocol [2], MorphMix [10] is a peer-to-peer circuit-based mix network, and the popular Tor [4] network is onion routed.

Tor faces a major challenge: the pool of nodes available to route other people's traffic is limited. Individuals often desire anonymity for their communications, but they may be unwilling to route other (potentially illegal) traffic through their personal machines, and expose themselves to legal action in doing so. In addition, user-provided nodes are often short-lived and unreliable because they run on home machines or laptops, which can only route traffic when they are powered on and connected to the Internet.

In this position paper, we introduce the concept of a *dust cloud*: a dynamic set of short-lived VMs that run on cloud computing platforms and act as Tor nodes. Users can join a dust cloud when they require anonymous communication, and incur charges only while participating. We outline the core architecture and benefits of dust clouds (§2), before considering some of the challenges for successful adoption of dust clouds and suggesting avenues for development and research (§3).

[3] http://mixmaster.sourceforge.net/

2 Dust Clouds

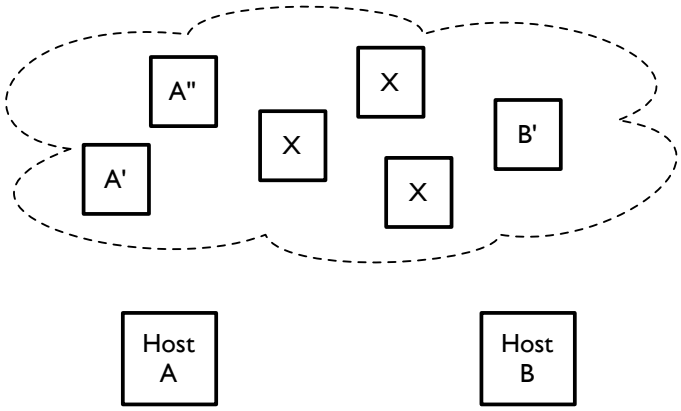

Fig. 1. Dynamic VM-based Tor nodes in the cloud. Host A spawns two VMs (A' and A'') and Host B spawns a single VM (B'). All VMs are full Tor routers.

To use a dust cloud, we assume that users have access to public cloud computing facilities, e.g., Amazon EC2 or Rackspace. When a user wants to communicate anonymously, they spawn one or more *dust motes*, cloud VMs that run Tor and are configured to survive for fixed periods of time (possibly all different). These Tor nodes receive the user's traffic as possible ingress points, as well as routing for other nodes as normal. When the user finishes communicating via a particular dust mote, they disconnect from it, leaving it to continue routing other traffic until its allotted time expires and it is destroyed.[4]

How is this better than the existing model of the user's host directly acting as a router?

Privilege Separation. It is no longer the user's host computer that routes other traffic in the Tor network, but their dust motes instead. In case of, for example, complaints about abuse originating from the user's Tor node [8], the user can terminate the dust mote and demonstrate from the code image (also hosted on the cloud) that the offending traffic came from a participant in the mix network. This is considerably more difficult to do when Tor is installed directly on a personal computer, which typically have multiple applications that can generate traffic. Privilege separation also makes it more convenient for users to run exit routers, an important benefit as the Tor network currently suffers from a lack of exit bandwidth.

[4] A real-world analogue uses ionization trails of meteors during atmospheric entry to establish short-lived communications paths [7].

Storage Separation. A dust mote can run independent processing and has some storage, enabling it to buffer traffic to improve mixing. Doing this efficiently requires protocol proxies inside the dust mote that understand the application traffic to some extent. For example, an SMTP proxy could receive e-mail, and hold it for a random amount of time before passing it along to the next Tor node. Crucially, this delay does not require the user to be online and connected—the dust mote continues running for its allotted time without the original user being connected.

Processor Separation. Running proxies alongside nodes enables other interesting use models. Streaming large video files is currently impractical since the regular traffic patterns and limited node bandwidth make anonymising the paths difficult. Viewing a long video or, more generally, performing any long-lived transfer over Tor makes it easier for an adversary to perform a timing correlation attack. If the user requests the content in advance, the dust mote can cache the content from the origin server. The user then re-connects to Tor and retrieves the content for offline viewing anonymously, but without having to be online while it is assembled. Similar methods work for PDF files (HTTP range requests) and peer-to-peer networks, since the dust mote's local storage can serve as a buffer. For the truly paranoid, the dust mote could even connect to Freenet [3] to store its data.

One extremely useful combination of streaming and clouds might be to provide anonymous audio, and even video, conferences. Self-organising transcoding systems have been proposed to let real-time audio scale with network capacity in multicast trees [6]. These can be adapted for scaling with *anonymity* in mind, by introducing jitter when routing audio to the participants so that everyone receives a slightly different signal. The CPU capacity for this would be provided by dust motes owned by those participating in the call.

Usage Accounting. Each dust mote that the user creates is paid for, and helps to grow the network as well as anonymise that user's traffic. Thus, a heavy user will naturally pay more by spawning more dust motes, and since these dust motes *must* route other user's traffic in order to preserve anonymity, it also helps to grow the network as a whole.

Ease of Use. The provision of pre-configured VM templates makes it easy to create and destroy Tor nodes in the cloud. Ease of use is important for mix networks in order to grow the set of users, and hence the size of the anonymity set and the relay bandwidth available. Using pre-defined templates which can clearly be shown to act *only* as Tor nodes and *not* to manipulate the data also helps to achieve plausible deniability.[5]

[5] There are cases of arrests due to running Tor exit nodes, e.g., `http://bit.ly/5WAflQ`.

3 Challenge/Response

Cloud computing is a promising platform for a mix network such as Tor, but it also presents several interesting challenges. In this section, we discuss some of these challenges, and suggest potential mitigations wherever possible.

3.1 Longevity

Challenge. Dust motes are relatively short-lived and exhibit a high churn rate. This can disrupt Tor usage: a dust mote might be terminated while still in use as a relay.

Response. As a cloud VM's lifetime is paid for and fixed ahead of time, dust motes can advertise their scheduled termination time, allowing users to manage the effects of termination. Tor circuits are pre-emptively constructed and rotated frequently, so clients can stop using a dust mote that is about to shut down without disruption. Indeed, in many ways the predictable life expectancy of dust motes makes the cloud a more suitable platform for Tor than personal computers, which may be turned off at any time without warning.

3.2 Diversity

Challenge. There are two associated diversity challenges for the dust cloud, provider and geographic. At present there are relatively few significant cloud computing providers such as Amazon or Rackspace, and these major providers have restricted geographical diversity—and hence applied jurisdiction—of their datacenters. For example, Amazon EC2 has datacenters in only three regions: California, Virginia, and Ireland. Both effects make it relatively easy to monitor or block access to the cloud for a set of users.

Response. In response we note that there is a similarly limited number of major ISPs, and as cloud computing takes off we can expect the number of cloud computing providers to increase from its current level. Indeed, previous studies of location diversity [5] have indicated that the best places for nodes may be at points that are connected to a large number of other ASes. Cloud providers are likely to be positioned at such locations in order to provide good network connectivity to their customers.

3.3 Billing

Challenge. As cloud instances are paid for by users, billing records may provide a way to link individuals with their dust motes.

Response. We note that the same is also true for residential ISP connections and standard use of Tor. This effect could be mitigated in at least two ways. First, by building an exchange where users can swap access to dust motes. For example, user A pays for one dust mote but allows user B to use it as a cloud node, while user B pays for a different dust mote but allows user A to use it. Second, by providing coarse-grained pre-payment options so that users' detailed usage need not be tracked by the cloud provider. If the user then multiplexes "legitimate" use of cloud nodes with their use as dust motes, it may help them to attain plausible deniability.

3.4 Traffic Mixing

Challenge. As the number of dust motes might easily scale in proportion to the number of users, it could be difficult to ensure a sufficiently rich mix of traffic to provide effective hiding.

Response. Again, we propose two ways in which this can be addressed. First, by establishing the market proposed above for dust motes, in order to restrict the dust cloud's rate of scaling and ensure that, while providing acceptable service, it does not scale up so fast that the level of mixing is too low. Second, by using the fact that dust motes have compute and storage available to generate spurious mix-in traffic. Doing this effectively is an open problem, but the extra resources available to dust motes might make it more tractable in this environment.

3.5 Attack by the Cloud Provider

Challenge. Having both physical access to the cloud VMs and control of the virtualization stack, the cloud provider can potentially inspect a dust mote's memory and storage, compromising private keys and modifying code [9].

Response. There are two layers of trust here: (*i*) the provider controls the host machine and can inspect memory, network and disk; and (*ii*) other users on the same cloud infrastructure may snoop on traffic via timing attacks [11]. At some level protecting against the provider is impossible to prevent: with physical access to both computing and network infrastructure, the provider can do anything it so chooses without the user even being aware.

In order to mitigate the second risk, we require some cooperation from the provider. By using trusted computing techniques [12] or disaggregating the virtualization stack [9], the provider can limit the opportunity for compromise. In practice, the user can make it more difficult for a particular provider to compromise the system (*i*) by ensuring diversity in placement among their set of dust motes; (*ii*) by frequently and randomly hopping dust motes between cloud nodes; and (*iii*) by taking care with items such as private keys, e.g., avoiding storing them on the disk to which they are applied. We believe one of the biggest

dangers to anonymity in Tor remains the shortage of exit nodes and the possibility of malicious snooping on them [13]: a dust cloud would greatly increase their number and help mitigate this problem.

We would like to thank Christian Kreibich, Alastair Beresford and Jon Crowcroft for useful discussions on this paper.

References

1. Amazon EC2. `http://aws.amazon.com/ec2/`.
2. D. L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–88, Feb. 1981.
3. I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong. Freenet: A distributed anonymous information storage and retrieval system. *Lecture Notes in Computer Science*, 2009:46–66, 2001.
4. R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium*. USENIX Association, 2004.
5. N. Feamster and R. Dingledine. Location diversity in anonymity networks. In *Proceedings of the 2004 ACM Workshop on Privacy in the Electronic Society*, pages 66–76. ACM, 2004.
6. I. Kouvelas, V. Hardman, and J. Crowcroft. Network adaptive continuous-media applications through self organized transcoding. In *Proceedings of the 8th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, pages 117–128, 1998.
7. K. Mahmud, K. Mukumoto, and A. Fukuda. Development of MBC System Using Software Modem. *IEICE Transactions on Communications (Special Issue on Software Defined Radio and its Technologies)*, pages 1269–1281, 2000.
8. D. Mccoy, K. Bauer, D. Grunwald, T. Kohno, and D. Sicker. Shining Light in Dark Places: Understanding the Tor Network. In *Proceedings of the 8th International Symposium on Privacy Enhancing Technologies*, pages 63–76. Springer-Verlag, 2008.
9. D. G. Murray, G. Milos, and S. Hand. Improving Xen security through disaggregation. In *Proceedings of the 4th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments*, pages 151–160. ACM, 2008.
10. M. Rennhard and B. Plattner. Practical anonymity for the masses with MorphMix. In *Proceedings of the Financial Cryptography Conference (FC 2004)*, pages 233–250, Key West, USA, Feb. 2004.
11. T. Ristenpart, E. Tromer, H. Shacham, and S. Savage. Hey, you, get off of my cloud: Exploring information leakage in third-party compute clouds. In *Proceedings of the 16th ACM Conference on Computer and Communications Security*, pages 199–212. ACM, 2009.
12. N. Santos, K. P. Gummadi, and R. Rodrigues. Towards Trusted Cloud Computing. In *Proceedings of the 1st USENIX Workshop on Hot Topics in Cloud Computing*, Berkeley, CA, USA, 2009. USENIX Association.
13. L. Sassaman. The Faithless Endpoint: How Tor puts certain users at greater risk. Technical Report 2007-003, ESAT-COSIC, 2007.