Bribecaster: Documenting bribes through community participation

Citation: Manas Mittal, Wei Wu, Steve Rubin, Sam Madden, and Bjorn Hartmann. 2012. Bribecaster: documenting bribes through community participation. In Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work Companion (CSCW '12). ACM, New York, NY, USA, 171-174.

As Published: http://dx.doi.org/10.1145/2141512.2141570

Publisher: Association for Computing Machinery (ACM)

Persistent URL: http://hdl.handle.net/1721.1/72949

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

Bribecaster: Documenting Bribes Through Community Participation

Manas Mittal and Steve Rubin
Computer Science Division
University of California, Berkeley
{mittal, srubin}@cs.berkeley.edu

ABSTRACT

Corruption is endemic in emerging economies, where many transactions of private citizens with government institutions require payment of bribes. The Bribecaster web application enables citizens to report and consume corruption information about dealing with government offices. Bribecaster uses a novel privacy-preserving implicit login schema and one-way hashing for protecting user identities while simultaneously ensuring the accuracy and integrity of reports. This citizen-induced transparency facilitates rational social and individual responses to corruption. Participants in our first-use user study of Bribecaster rated the system highly for its usefulness.

Author Keywords

Privacy, Crowdsourcing, Mobile Applications, Corruption

ACM Classification Keywords

H5.m. Information interfaces and presentation, K.4.1 Public Policy Issues, J.4 Social And Behavioral Sciences

General Terms

Human Factors, Design, Economics

INTRODUCTION

A significant percentage of the world's population lives in developing countries and corruption is a major problem in many such countries. For example, India has made fighting corruption a key component of its development strategy [7]. Corruption is a two-sided problem. People in positions of power demand bribes for performing or expediting work. Individuals and corporations pay these bribes, which are often considered part of normal business practices in the developing world [7, 11]. Quotidian corruption frequently involves paying petty bribes to low and mid-level officials, e.g., in law enforcement, government offices, or to tax and license inspectors [7].

Figure 1: The frontpage of the Bribecaster web app shows a feed of new bribes, our anonymity policy, and a streamlined reporting form.

Many efforts to stem corruption focus on punitive action against corrupt officials. This top-down approach does not currently work in most developing countries [4]. Could a bottom-up approach—where citizens exchange corruption information with each other—be more effective? To motivate our research, we conducted a formative corruption survey of 102 Indian participants; results indicate that individuals who have had to pay bribes are open to reporting corruption information. Our research investigates whether bribe market transparency can be achieved by a confidential bribe-reporting application.

We are developing Bribecaster, an application that enables community members to anonymously report their interactions with government functionaries. Reporting has two principal benefits: first, surfacing information about the bribe market can empower individuals to make rational choices (e.g., deciding to seek out a different office, or deciding how much to pay) [3]. Secondly, transparency can draw public attention to egregious violations. Such scrutiny may ultimately lead to a decrease in corruption levels. We are initially targeting India, because of our team's experience, India's significant English speaking population,

DIS 2012, June 11-15, 2012, Newcastle, UK.

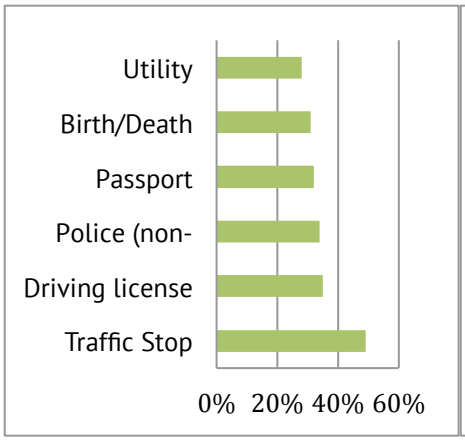

Figure 2: Common transactions that required payment of bribes

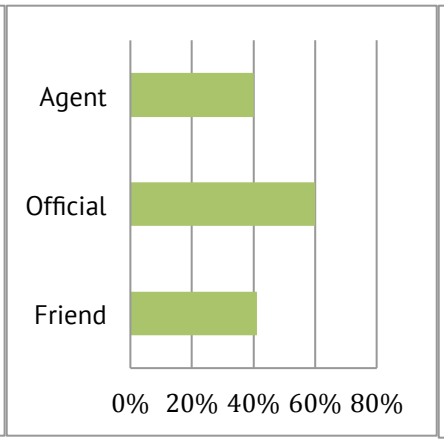

Figure 3: Common sources that provided information about bribe prices

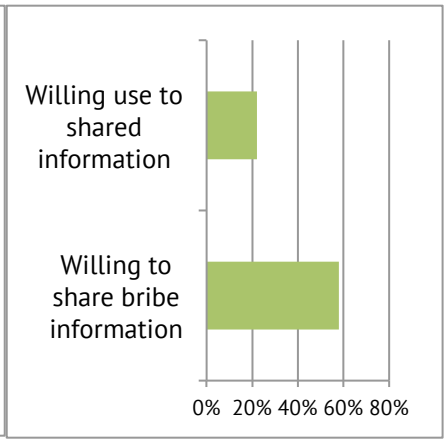

Figure 4: Respondents are wary of shared information.

and its democratic government, which should be receptive to anti-corruption measures [2].

We are specifically interested in collecting information about *harassment bribes*. Harassment bribes are a subset of bribes where the payment is essentially mandatory following a legitimate process (i.e., rules are not bent, but a bribe is demanded from people so that they can get what they are legally entitled to). Such bribes are pervasive in the developing world. For example, getting a tax refund often involves paying a part of the refund as a kickback [6].

In the current legal framework in countries like India, both paying and accepting bribes is illegal. Basu et al. have suggested how this leads to a convergence of interests between the bribe payer and the bribe taker. They advocate political and legal change to make it legal to give bribes (while retaining the illegality of taking bribes). This divergence of interests will lead to a better "mutual check"—officials are less likely to demand bribes when the payer has no incentive not to report it.

While the political and legal change is being suggested by government economists [6], it is unlikely to be realized given the lack of political will to affect such change. Politicians are often a part of the corruption chain where slices of bribes collected by lower officials are passed up the government hierarchy, all the way to the top elected officials. A system such as Bribecaster provides an alternative—by enabling individuals to report their bribe experiences anonymously and in a public forum, thus shielding the reporter from legal action while publically broadcasting their experiences. This introduces a key challenge in the design of the Bribecaster system – protecting the reporters' privacy. This objective supersedes all others.

BACKGROUND

In order to evaluate the utility and feasibility of Bribecaster before building the system, and to inform our design, we deployed a formative survey to gain insight about the prevalence of corruption in India, the willingness to report and share information about it, and the current technology environment. The survey was deployed on Amazon Mechanical Turk, which has a large Indian workforce. We collected 102 responses over 2 days, paying each participant $0.25. Our results have a potential bias in that respondents may be more tech-savvy than the general population. However, we have no reason to believe that participants will show a systematic bias regarding corruption behavior.

The key findings of the survey were:

Most Respondents Pay Bribes

90% (92/102) of respondents indicated that they had given bribes in the past, and over 82% (84/102) identified specific individuals to whom they had paid bribes. Figure 1 shows transactions that often required bribes. These data indicate that corruption is pervasive. Results also indicate that harassment bribes are common – for example, over 30% of users paid a bribe to get a passport.

Respondents Use Outside Information to Price Bribes

We asked participants about how they determined the price to pay for a bribe. 60% (56/92) of those who paid a bribe indicated the official provided a number; 41% indicated that their friends told them the amount, and 40% indicated that a middleman (commonly referred to as an "agent") told them the appropriate amount (Figure 3). Agents are valuable because they know who to bribe and how much to pay. Such information could also be obtained through a crowdsourced database of bribe reports, and provides an important incentive to consume information from a system like Bribecaster.

Respondents Already Share Their Bribe Experiences

We next asked about existing forms of bribe-related information exchange. 52% had told friends or relatives about paying bribes because they felt bad about the transaction; 34% told others to keep them informed. Only 14% did not report paying a bribe because they felt

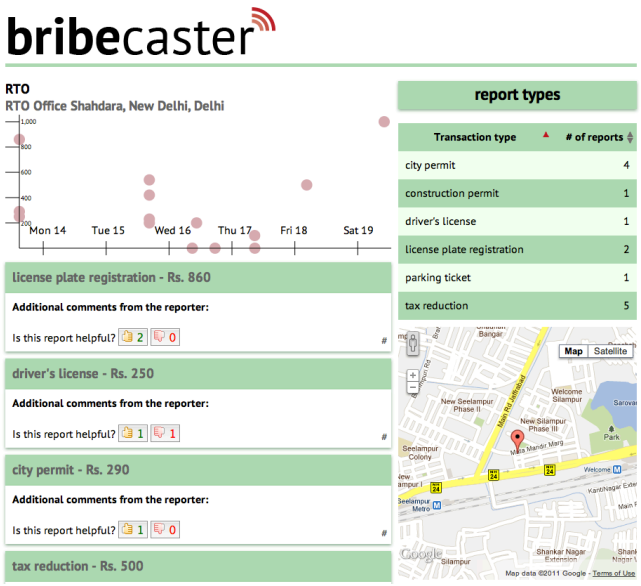

Figure 5: An office-specific page features and interactive graph of pricing trends, the ability to restrict to certain types of transactions, and buttons to vote for the helpfulness of each report.

embarrassed. The high level of informal sharing, and the low level of embarrassment about paying bribes suggest that users may be willing to share information electronically.

Respondents Are Wary of Anonymous Information

When explicitly asked if subjects would anonymously report bribes through a website or mobile application, 58% responded positively. Would respondents use anonymous bribe information? Figure 4 shows that of those who had valid answers, 22% indicated yes, while 78% indicated no. More people are willing to report bribes than are willing to use this information. We speculate that this result may be due to a lack of trust in anonymous reporting. We conclude that the trade-off between trust and anonymity is a key design consideration.

RELATED WORK

The goal of our research is to learn how to build systems that can foster trustworthy knowledge exchange between anonymous users in sensitive situations. The Bribecaster application provides a platform on which to run studies to investigate issues of trust and anonymity. There are both commercial apps and academic research on corruption. IPaidABribe.com [3] is a website for collecting bribe-related citizen information, and Bribespot [1] is an iPhone app for similar purpose. Both these applications are focused on reporting bribes, but don't provide finely grained bribe information that will be useful for when having to pay a bribe, and don't address the tension between anonymity and trust.

Specifically, the IPaidABribe site has similar reporting structure to the Bribecaster interface, but different objectives, mechanisms and philosophy. IPaidABribe provides no explicit anonymity guarantees to bribe reporters. Instead, its privacy policy states that it will willingly provide information to state actors and law enforcement. The site then deals with the apparent illegality of such reports by rendering them toothless – official names and offices are (perhaps manually) redacted from the reports. Thus, the principal operating mechanism appears to be to "not ruffle the feathers" and is bound to remain largely ineffectual in affecting change. In contrast, the Bribecaster system attempts to protect the reporters by anonymizing their reports, and relies on crowdsourcing to filter out false reports.

Prior research [10] has focused on creating trust in decentralized anonymity networks, but our problem of having a centralized server and a web-based user interface is not addressed. Yelp.com uses a filter [5] to display only the most trusted reviews, but their algorithm relies on the existence of trusted users—something that we cannot use in an anonymous system. Linguistic-based methods have been developed to detect fraudulent reviews on web sites [8]. While these techniques are for reviews, we may be able to modify their methods to identify fake bribe reports.

INTERFACE

The Bribecaster web app allows users to search for and report transactions. The frontpage (Figure 1) shows recent bribes in an updating stream and map display, and provides both search and reporting forms.

The reporting form asks for the type of transaction, its location, the involved official name/position, the amount of the bribe if any, and an extended comment about the transaction. The location and transaction type information is completed based on existing entries in our database. Additionally, the location suggestions also include nearby locations. These are obtained by using the "nearby locations" feature Google Places API, where the reporter's location is computing by using an IP to City Name mapping database (PyGeoIP). This autocomplete functionality is provided throughout the Bribecaster service.

Search is the primary navigation mechanism provided by the Bribecaster system. Users can search for bribes based on a specific location, which defaults to their current location (computed via IP geolocation).

Each office has its own page (Figure 5), which shows either all bribes at that location, or that location's transactions filtered by some keywords. The top of the page features an interactive graph built with d3.js that allows the user to see how bribe amounts for transactions have changed over time. Selecting a report type in the right column restricts the graph and the listed transactions to only those that match that type. Each report contains thumbs up / thumbs down buttons that give users indicators of whether others users found the reports useful and reliable. This is a key construct – we are relying on users of the website to provide signals

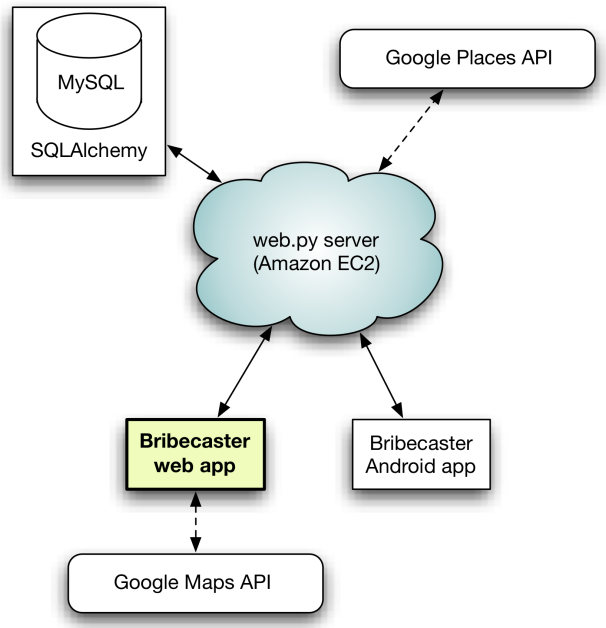

Figure 6: An overview of Bribecaster's architecture

for other users about the veracity and usability of the reports.

Location pages with and without keyword filtering have human-readable permalinks for easy sharing. Individual transactions have dedicated pages as well. Location pages also contain a direct link to report additional transactions at that location.

By restricting our system to just a few easy-to-understand page types, we hope that users will have a cohesive and streamlined user experience.

IMPLEMENTATION

Scalability & Modularity

Bribecaster is built on the python web framework web.py and runs on Amazon EC2. Applications like Bribecaster have to potential to go viral, so the scalability of EC2 is vital to our implementation. The EC2 server handles the back-end and maintains the database. At the moment there are two application front-ends that access this server: an Android app (not discussed here) and a web application, which we use for our studies. The modularity provided by this server makes it straightforward to extend Bribecaster to new web-enabled platforms. Figure 6 shows our system architecture. Additionally, the EC2 servers are based in the United States, and provide a certain level of physical security.

Protection

In order to ensure that users maintain their privacy when using Bribecaster, we have implemented various protections. Users who visit the web application are not tracked using conventional methods like cookies and logins, which are potentially incriminating indicators of participation. While we do not expect repercussions for users who merely browse the site, we hope to protect those who submit sensitive information. Instead, a submitted transaction report is indexed using a one-way hash of the user's IP address, which then serves as an "implicit login" on behalf of the user. Note that using this means that there is no known way for any individual (including the site administrators) to decode a submitter's IP address. This way we can control against spam—e.g. someone posting several fictional reviews to maliciously affect (boost or degrade) the reputation of an office—by disallowing similar reports from identical hashes. Proxies can get around this barrier, but it provides a rudimentary layer of both quality assurance and protection in our system.

We also have an SSL enabled version of the site that protects against man in the middle attacks. These attacks might be orchestrated by ISPs on behalf of the local law enforcement.

Data Model & Search

Bribecaster uses a MySQL database with python SQLAlchemy wrappers to provide an object relational mapping for our data. Our data model has two classes of objects: transactions—which cover both bribes and bribe-free dealings—and offices/locations. Using IP geolocation and the Google Places API, we allow users to search for locations that are near them. When transactions are added, their locations are added to our database if they are not yet present. This method uses Google Places to supplement location search, but does not rely on it completely because we maintain a local database of locations.

The Bribecaster server features a search provider to efficiently search for transactions near locations. Built on top of the SQLAlchemy ORM, our search provider returns bribe transactions at locations within a given radius of GPS coordinates provided by IP geolocation or by city-based geolookup. It also facilitates autocomplete for locations in both searching and reporting, which improves the user experience.

Malicious Behavior: Prevention and Detection

By using the implicit weak login of the one-way hashed IP address, we can restrict posting of new transaction reports for IP addresses that have, in the past k hours, submitted a report at that same location. This prevents a malicious reporter from artificially inflating or deflating an office's reputation on Bribecaster. The value of k is currently 12 hours, but may be modified once Bribecaster is in production after an analysis of its traffic patterns.

EVALUATION

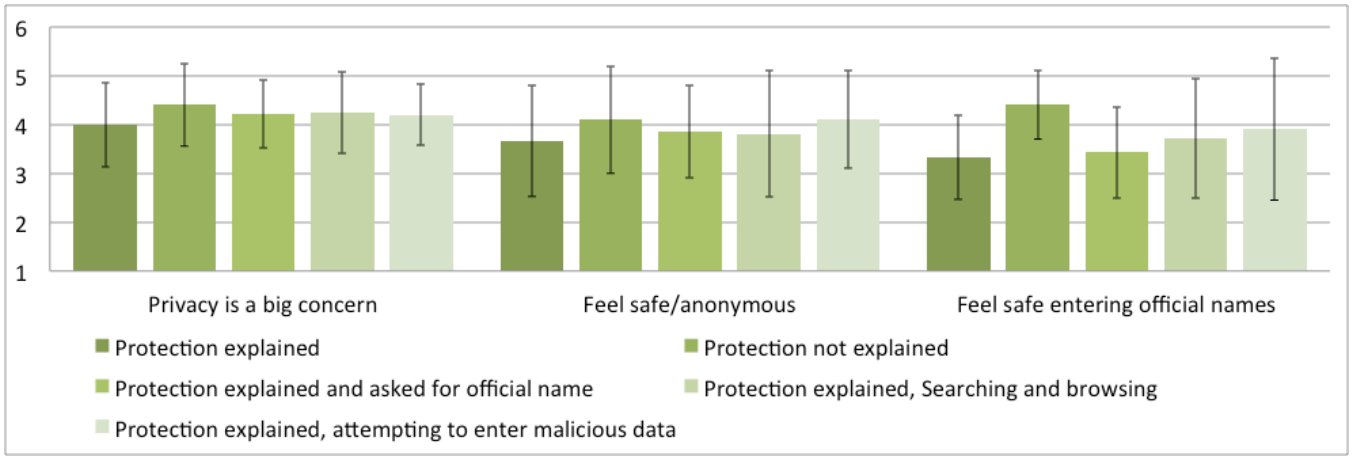

Figure 7: Results of A/B tests on Mechanical Turk that test how users react to different features of the site. These charts show the mean and standard deviations of responses on a Likert scale (1- strongly disagree, 5 – strongly agree)

Our system aims to be both trustworthy and useful. If it fails to be trustworthy, Bribecaster will not be a credible source for bribe-related information. Likewise, if our system fails to provide a useful means of navigating and learning about bribe transactions, Bribecaster's trustworthiness is irrelevant. For our experiments, we used Amazon Mechanical Turk to recruit our target audience—people from India who have likely taken part in the Bribe-based economy—to perform tasks using the Bribecaster web app and then to complete Likert-scale surveys about their experiences. Although a longitudinal study over a few years would be the ideal validation of our work, we have begun our analysis with multiple experiments to refine and evaluate the Bribecaster prototype along key dimensions—trust, anonymity, usefulness and virality.

Tension Between Trust and Anonymity

In a less anonymous system, a user is less likely to lie, but also less likely to report sensitive information. So one key question is, "what is the right level of submitter information to reveal." Other key questions are: will people feel comfortable reporting, will they report the right information, and will others trust it. We could handle anonymity in three granularities:

Approach 1 – Complete Anonymity

Reporter information is kept anonymous, and their IP is recorded and stored as a one-way hash to prevent multiple entries for the same office. This is the approach we employ in our tests and in the current deployment of bribecaster.com.

Approach 2 – Artificial Names

Artificial names are created as a proxy for the bribe reporters. We tie an artificial name to a hashed IP address.

Approach 3 – Partial Anonymity

"Reporter by someone in Delhi" style bylines are displayed to readers. The location of the bribe reporter is determined by their IP address. Will people trust this more? This provides more information to the bribe-reader, but the new information may be redundant. We could use this approach on the back-end to verify that bribe reporters were located near the offices in question.

Will people enter information?

We have done studies to test whether people are willing to enter information into the system.

Our experiments have included:

- A/B test of what information people are comfortable entering (Official Full Name, Title). In an initial study ($n=21$) we found no statistical indication that users asked to enter official names when reporting bribes felt any difference in safety on the web site.
- A/B test of what information about our privacy controls we need to reveal and feature on our website. In an initial study ($n=19$), we found that users that we did not explain the protection features to felt safer and trusted the site more, but the t-test results were not significant ($p=.47$). This seems to fit the classic adage, "ignorance is bliss."
- A/B test of "institutional flow of trust," i.e., Berkeley hosting this as opposed to it being run locally. Does the user care about this? (RELEVANT?)

We rely on Likert scales for many of these results as a proxy for a true longitudinal study.

In the future we will test how upvotes and downvotes affect readers. We test this by informing bribe-reporters and viewers that the average bribe report gets 4 positive reviews (and perhaps sending them an email when they get positive ratings). After this, we can measure traffic to particular bribe pages as a metric of interest. Surveys can then be used to determine how people interpret the rankings.

Virality

The success of a two-sided system such as Bribecaster depends on its adoption by a critical mass of people. We would like a high *virality factor*, i.e., the willingness of users to recommend the system to other users, which is a

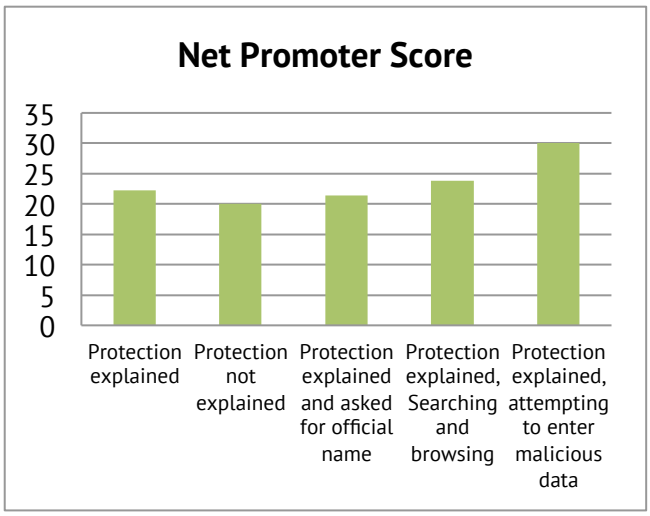

Figure 8: The change in Net Promoter Score over time, as features are added and removed

proxy for future success of Bribecaster. We use the *Net Promoter Score* (NPS) [9] metric to gauge the virality of the system. NPS is also proxy for how useful users find the system to be. The NPS is determined by asking users the following question on the scale, 0: not at all likely, to 10: extremely likely: *How likely are you to recommend Bribecaster to a friend, colleague, or relative?* NPS is defined as the number of 9's and 10's minus the number of 0-6 responses. We use the net promoter score as a success metric as we continue to develop and tweak the system. In Figure 8 we plot the NPS as we add and remove features.

NPS is generally considered a important virality metric. However, in our case, users were recruited from Mechanical Turk and paid for participating in our studies. Users may have been attempting to please us by responding favorably to this question. However, we do notice relative variations among the net promoter scores. For example, participants of the study who were asked to enter multiple reports maliciously, and failed showed a higher net promoter score (30) than in other tests. This indicates that the NPS serves as a useful proxy for even paid users. In general, our NPS scores trended upwards and helped us refine our design.

Usefulness, Engagement and User-Retention

We tracked the general level of engagement of the users in a free form searching and browsing task. On average, a user performed over 6 searches, although we prompted the user to perform 5 searches or fewer. Figure 9 shows a histogram of the number of searches that were performed. Additionally, across all our studies, the average participant (n=122) spent 8.25 minutes on the website. Again, this is quite high. Our bounce rate was about 28.22% and participants visited 9.06 pages on average.

Additionally, we received many ideas and suggestions from study participants indicating a high level of enthusiasm. Several users remarked that the website was very useful. One user suggested reaching out to the local media to popularize the website. Users also gave design suggestions—one user suggested improving the design of the website to make it look professional so that it appears more trustworthy. Other users suggested changes to format and layout.

Malicious Intent Detection & Prevention

If successful, the Bribecaster system is likely to attract malicious users. The key attack we have to guard against is that in which one individual or sets of individuals report incorrect data with an intent to malign another individual or office. Conversely, multiple reports might be filed to embellish offices with positive reports and to drown out offices and officials with legitimate positive reports. Our key strategy is to limit the number of reports that an individual (or a group of related individuals) can file.

One-Person, One-Vote

We currently use the users *IP address as a weak implicit login***.** A given IP can only submit one report for a given office, task, from a given IP addresses. Since we store a one-way hash of the IP address, we can easily determine this. Currently, we explicitly reject such reports, i.e., the user is informed that their report was not registered. In the future, we intend to silently reject it, i.e., to store the report but to not display it or use it in aggregate computations. Similarly, a user can only upvote or downvote an existing report only once.

The IP based scheme has obvious limitations – users can use multiple different computers, call their friends and ask them to submit reports. Additionally, users using a cyber-café might be needlessly restricted from reporting such bribes. We recognize such limitations, but believe that such constraints are likely not dominant factors.

Hypothesis: An IP-restricted reporting mechanism is an effective, good-enough mechanism for malicious bribe reporting.

In a tasks posted on Amazon Mechanical Turk, we asked participants to post multiple bribes for the same task and office. Participants were prompted to use *any means necessary*. We explicitly stated that users would be paid for successfully reporting a given bribe instance, and not penalized for using any potential mechanism. We paid participants $0.25 for attempting the task, and offered a bonus ($0.50) for each multiple report they filed (up to 6 bribes for $3).

Of the 10 users who participated in the study, 1 user was able to insert 3 bribes. The remaining 9 users spent an median of 8.2 minutes (with some users spending as much as 18 minutes) but did not succeed. The one user who succeeded reported using an online proxy (Ultrasurf.com).

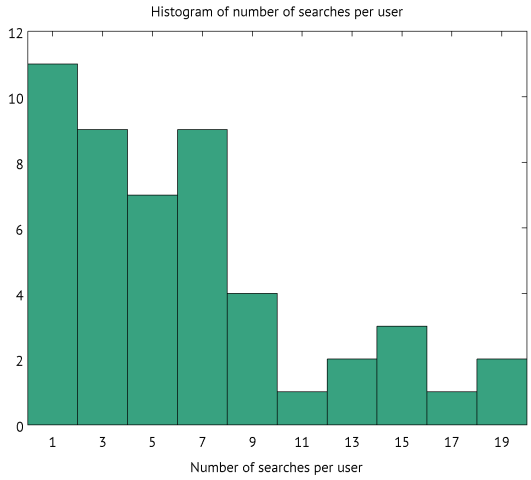

Figure 9: Users in the free form tasks searched more than we asked them to.

This indicates that while the system is easily beatable, a majority of non-technical users are unlikely to be able to easily fool with our present defenses. We also note that these users were sourced from Mechanical Turk, and are likely more technically competent than the typical user. Interestingly, over 54% of users browsed Bribecaster with Chrome, indicating a more technically competent set of people than those using the default browser.

We will continue to monitor and institute additional defenses as the threat model evolves. One approach would involve restricting reports to the same province as where the office is located (based on the reporting IPs), and to blacklist proxy IP addresses.

De-anonymization attacks

Another class of attacks attempts to decipher the reporting person based on the amount and the *modus-operandi* reported. We currently offer no protection against such attacks, but we are planning on preventing them in the future. The core idea would be to initially reveal aggregate statistics for a given office/location, and *selectively reveal* more information as more reports are filed. For example, the official names may be revealed only when more than half a dozen reports are filed, or when more than 3 reports name the same official.

We also believe that given the nature of reports, i.e., commonplace everyday tasks, it is likely to be common enough that data based de-anonymization attacks will be ineffective.

FUTURE WORK

Bribecaster as an information source

We intend to provide the name and contact information of associated with a particular office. We will also provide information of the chain of command, and provide a public mechanism to enable users to contact these officials.

Further, given enough data, we will be able to detect the change in bribe levels as a new official in transferred into our out-of a job. By creating a public 'corruption index' of administrators, we hope to discourage bribe taking.

Additionally, providing this information also assists in "seeding" the site, i.e., making it useful without having to have a lot of initial reviews to begin with.

A Positive System

Bribecaster is currently focused on reporting bribe instances. It has a negative skew in that it collects reports of corruption in day-to-day life. Because of this negativity, questions of security and trust are important to consider. In order to skirt these issues, we could re-imagine the system as a primarily positive site, a sort of "Linked-In" for public officials. Users could leave reports of positive transactions and write brief notes of recommendation for officials. If an official had no recommendations on the site, a user could assume that the official was corrupt or in some way untrustworthy. None of the information in this system would need to be private or protected because there is no negative information shared, and it might put pressure on officials to clean up their acts and collect positive reviews. There are a number of issues that would arise here, as well, such as ensuring that positive reviews of officials are not falsified, but it may be easier to accomplish these goals without putting users at risk.

A Wider Net

One study participant reported bribes he had to pay to the officials at a private educational institution. Others report having to pay bribes in non-governmental contexts. We intend to extend Bribecaster to cover private organizations. We also intend to deploy Bribecaster for other developing regions, particularly Pakistan and parts of Africa.

Quality Control

Because we do not have a large user base at the time of writing, we have yet to run into issues of quality control. However, if the application becomes popular and receives an influx of transaction reports, we will need to filter out "bad" responses. While typical collaborative applications would use a login system to achieve quality standards, the sensitive nature of our information prevents that. Instead, we will identify patterns using the content of reports paired with hashed IP addresses as probabilistic identifiers.

Sister Cities

Widespread use of Bribecaster could give rise to social incentives for administrators and professionals to lower bribes. One idea is to facilitate competition among offices and office administrators is to periodically publish some corruption index metric. Another approach is to create competition with sister cities in neighboring countries or regions.

CONCLUSION

In this paper we presented the Bribecaster service that enables individuals in the developing world to safely and

securely report bribe-related transaction information. We presented new mechanisms that facilitate privacy control while simultaneously guarding against malicious reporting.

ACKNOWLEDGMENTS

We would like to thank Professor Bjoern Hartmann for helpful discussion and advice, and Wei Wu for help in informing the design of the Bribecaster API, and a special thanks to Dinsha Mistree for valuable product suggestions.

REFERENCES

[1] *Bribespot*. bribespot.com.

[2] *Indians See Threat From Pakistan, Extremist Groups*. Pew Research Center, http://www.pewglobal.org/2010/10/20/indians-see-threat-from-pakistan-extremist-groups/, 2010.

[3] *IPaidABribe*. IPaidABribe.com.

[4] *Transparency International Annual Report 2010*. Transparency International, http://www.transparency.org/content/download/61964/992803, 2010.

[5] *Yelp's Review Filter Explained*. Yelp.com, http://officialblog.yelp.com/2010/03/yelp-review-filter-explained.html, 2010.

[6] Basu, K. Why, for a Class of Bribes, the Act of Giving a Bribe should be Treated as Legal. *India Ministry of Finance Report*2011).

[7] Bertrand, M., Djankov, S., Hanna, R. and Mullainathan, S. *Does corruption produce unsafe drivers?* , National Bureau of Economic Research, 2006.

[8] Ott, M., Choi, Y., Cardie, C. and Hancock, J. T. *Finding deceptive opinion spam by any stretch of the imagination*. 2011.

[9] Reicheld, F. F. The one number you need to grow. *Harvard Business Review*, 81, 12 2003), 46-55.

[10] Sassone, V., Hamadou, S. and Yang, M. *Trust in anonymity networks*. Springer, 2011.

[11] Treisman, D. What have we learned about the causes of corruption from ten years of cross-national empirical research? *Annu. Rev. Polit. Sci.*, 102007), 211-244.