Information Extraction and Manipulation Threats in Crowd-Powered Systems

Walter S. Lasecki
Computer Science Department
University of Rochester
wlasecki@cs.rochester.edu

Jaime Teevan
CLUES Group
Microsoft Research
teevan@microsoft.com

Ece Kamar
ASI Group
Microsoft Research
eckamar@microsoft.com

ABSTRACT

Crowd-powered systems have become a popular way to augment the capabilities of automated systems in real-world settings. Many of these systems rely on human workers to process potentially sensitive data or make important decisions. This puts these systems at risk of unintentionally releasing sensitive data or having their outcomes maliciously manipulated. While almost all crowd-powered approaches account for errors made by individual workers, few factor in active attacks on the system. In this paper, we analyze different forms of threats from individuals and groups of workers extracting information from crowd-powered systems or manipulating these systems' outcomes. Via a set of studies performed on Amazon's Mechanical Turk platform and involving 1,140 unique workers, we demonstrate the viability of these threats. We show that the current system is vulnerable to coordinated attacks on a task based on the requests of another task and that a significant portion of Mechanical Turk workers are willing to contribute to an attack. We propose several possible approaches to mitigating these threats, including leveraging workers who are willing to go above and beyond to help, automatically flagging sensitive content, and using workflows that conceal information from each individual, while still allowing the group to complete a task. Our findings enable the crowd to continue to play an important part in automated systems, even as the data they use and the decisions they support become increasingly important.

Author Keywords

Crowdsourcing; privacy; security; extraction; manipulation

INTRODUCTION

Crowd-powered systems have recently become a popular way to surpass the capabilities of automated systems in many real-world domains. For instance, VizWiz [3] and Chorus:View [19] answer visual questions for the blind, Legion:Scribe [16] converts speech to text in real-time, Shortn [2] rephrases text into a more condensed form,

CSCW '14, February 15–19, 2014, Baltimore, Maryland, USA.

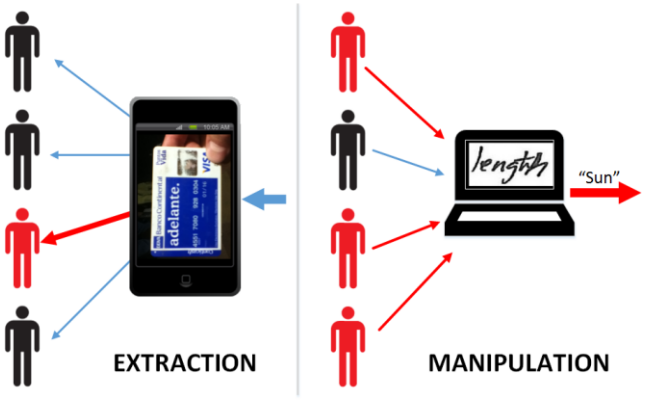

Figure 1. Crowd-powered systems are vulnerable to two types of attacks: unwanted information extraction (left), and malicious manipulative control (right).

Legion:AR [18] recognizes activities it has never seen before from video, Adrenaline [1] picks the "best" frame from a video, and Chorus [20] is an on-demand personal assistant capable of holding two-way conversations with a user. Each of these systems uses human intelligence to robustly solve a problem that artificial intelligence is not yet able to handle. However, doing so requires sharing potentially sensitive information with unknown people. For example, a photograph of a medication taken by a blind person for assistance reading the label (as in VizWiz) may include personally identifying information about the user. Figure 1 illustrates an example of a task where four workers are asked to extract text from an image of a credit card. Even if most workers are trustworthy, all it takes is one bad worker to steal the card number. Little is known about how systems can prevent unintentional extraction of data when using human intelligence as a computational resource [8].

The success of crowd-powered systems also means that the decisions made based on the input of crowd workers are becoming increasingly critical. For example, comScore is company that provides digital analytics to some of the world's largest enterprises, agencies, and publishers. Many significant business decisions are made based on comScore data, some of which are created using Mechanical Turk (www.mturk.com). As another example, Planet Hunters (www.planethunters.org) uses crowd input to determine where a new planet is most likely to be found, and then uses this information to dedicate scarce telescope resources.

Responses collected from workers are used to power systems that do language translation [30], search results ranking [4], and even fine-grained image recognition [5]. With increased reliance on crowdsourcing to make real-world decisions, the potential for external manipulation could become a costly threat. Significant resources could be devoted to attacking crowd systems much in the way they are currently devoted to influencing search engine rankings. The search engine optimization market is estimated at $20 to $30 billion dollars in the United States alone [28], and poses a real challenge for search engines. As crowd systems become ubiquitous, they will likewise become targets for new types of malicious manipulative attacks, which perhaps even use the crowd itself. Figure 1 illustrates how a group of malicious crowd workers might convince a hand writing recognition crowd system to incorrectly interpret an image.

This paper lays the groundwork for addressing the threat of information extraction and manipulation in crowd systems by investigating the potential vulnerabilities of current crowdsourcing approaches. It contributes:

- An overview of the space of potential threats to existing crowd-powered systems and types of attacks,
- Tests that illustrate the viability of using Mechanical Turk to recruit workers for malicious attacks, and
- Ways to use the crowd to self-regulate against attacks for high-risk tasks, using various techniques.

We begin with a discussion of existing crowdsourcing practices. While previous efforts have explored how to combat worker errors and workers who want to be paid for doing as little work as possible, we highlight potential threats to crowd-powered systems, such as the extraction of valuable information from a task or the manipulation of a task's outcome. We study the feasibility of individual and group attacks, and analyze how group attacks can be organized by a group of malicious workers or by the hiring of workers by a malicious entity. We present the results of a study performed on Mechanical Turk with 1,140 unique workers that demonstrates the vulnerabilities of the current platform to malicious tasks that actively attack another task by directing workers. We analyze the behaviors of workers in contributing to these threats, and find that while such attacks can be successful; some workers are unwilling to participate. This suggests there is an opportunity for crowd systems to self-regulate to protect themselves, and we conclude with a discussion of how future crowd-powered systems might be designed to prevent information extraction and manipulation.

PRIOR WORK ON CROWDSOURCING

Crowdsourcing is a form of human computation that relies on a diverse group of nearly anonymous workers with unknown skills and reliability to contribute to a larger task by completing small pieces of the task called *micro tasks*. A *crowdsourcing platform* is the system that recruits crowd workers and connects these workers with micro tasks for them to perform. Within a crowdsourcing platform, a task *requester* is the individual or organization who creates a public call (in this case in the form of a description of the task) and hires crowd workers.

Systems that use crowdsourcing are at risk of attack because requesters know very little about the workers they hire and have limited means for quality control. Here we give an overview of what is known about crowd workers and how quality control is currently handled. We then highlight some of the vulnerabilities of crowd systems and discuss existing approaches to thwart malicious workers.

Understanding Crowd Workers

Crowd workers are remotely recruited by a crowd platform to micro tasks issued by a requester. Typically the crowd platform and requester are different entities, and the relationship between the requester and worker is mediated by the platform. As such, the relationship is very limited, with little information provided for context. The crowd employed by a system might consist of a few individuals or of a large population, and the requester might not even know what country the workers are each located in.

Members of a crowd have many different incentives for contributing, such as monetary payments in paid crowdsourcing or a desire to contribute to scientific research in citizen science projects. In this paper, we discuss how different motivations workers have may influence attacks in crowdsourcing. In our experiments, we focus on paid crowds recruited from Amazon's Mechanical Turk marketplace.

The most common approach (used by Mechanical Turk) for requesters to recruit workers on a crowd platform is for the requester to post a task, and then let workers choose which task they want to complete from a list of options. This is beneficial to requesters who can begin to use knowledge they retain between sessions [21] and beneficial to workers who can choose tasks they enjoy [27].

Quality Control in Crowdsourcing Systems

Because the relationship between a requester and worker is limited and workers are required to have little or no background knowledge, quality control is very important to the success of a crowd-powered system. As a result, maintaining quality is an active area of research [10,11].

Since individual workers' inputs are often error-prone, it is common for task requesters to implement agreement-based filtering (e.g., voting or averaging) or iterative workflows. Using contributions from a group of workers helps ensure the quality of the final output. It has been shown in previous work that collective responses result in better performance than any single worker could achieve [2,9].

Research in crowdsourcing has focused primarily on improving crowdsourcing quality based on responses collected from a group of error-prone workers. For tasks with one unknown correct answer, voting is one of the most

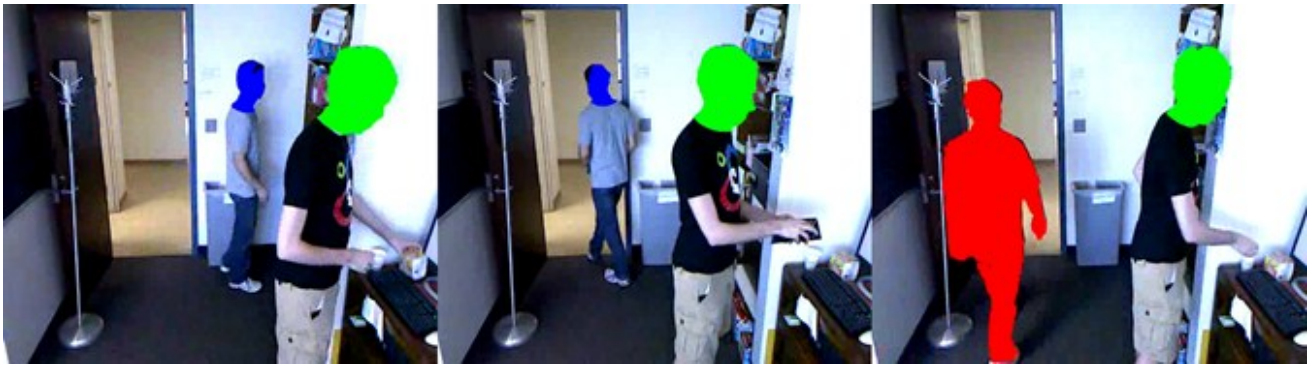

Figure 2. Prior systems such as Legion:AR have attempted to hide personal data from the crowd. Actors in the scene have been automatically identified and veiled in separate colors to preserve privacy and help workers identify which actor in the scene they should be focusing on. These veils can cover only the face (left two panels), or the entire silhouette of the user (right panel).

common techniques for aggregating multiple worker responses to predict the correct answer (e.g., Galaxy Zoo, Legion [16]). For estimation tasks, averaging responses is often an effective option (e.g., Legion:AR for image segmentation, Evanini and Zechner [6] for speech prosody). Other models of voting (e.g., ranked voting or tournament voting) have been used in idiom translation when majority voting alone is not successful [25]. Another common approach is to use iterative workflows for creating and revising content over a series of steps from multiple workers (e.g., TurKit [14] and Soylent [2]). Different than voting and averaging methods, which aggregate multiple workers' responses to the same micro task, iterative workflows involve individual or a small number of workers providing input at each step, which as we will see, may make these tasks more vulnerable to attacks from individuals or from groups of workers.

Existing Defenses Against Malicious Attacks

While quality control in crowdsourcing is an active area of research, little is known about coordinated malicious attacks on crowd systems. Efforts to avoid malicious workers have focused mostly on workers who want to strictly optimize their payment relative to the time or effort spent to complete a task [22]. These workers want to get paid the most for the least amount of effort, and they may take shortcuts (e.g., write scripts or give simplistic answers) to do so. However, in many important cases, workers are not just motivated by task payments, but also curiosity, beliefs, or interests. As crowd systems are used for increasingly valuable tasks, the payment involved may be minimal compared to the value of attacking the task. We identify two primary threats maliciously motivated workers pose to crowd systems beyond trying to optimize payments: 1) they can extract the private or sensitive information from tasks posted to a crowd platform, and 2) they can maliciously manipulate the outcomes of particular tasks.

Several existing approaches have been tried to prevent data leakage. The division of a task into micro tasks is typically used to allow multiple workers to contribute, facilitating agreement as well as simplification of the task. However, it can also be used to prevent any one crowd worker from seeing too much information, such as an entire medical record [25]. Legion:AR [18] used automatic methods to generate masks for specific pieces of information, such as a user's face (shown in Figure 2). Similarly, EmailValet [12] allows users to specify what data they share with the crowd.

Several existing approaches have also been explored to thwart task manipulation. Many of the crowd-based quality control measures designed to reduce noise and errors impede individual attempts to manipulate a task because they rely on agreement across different workers. However, coordinated efforts could create an artificial appearance of agreement. Gold standard tasks [13], where the requester knows the answer to a subset of questions, can be used to evaluate worker quality even when the majority is wrong, but must be generated a priori, which can be difficult or impossible in real-world settings.

Participants in a coordinated attack on a crowd-powered system can have varying motivations. For example, external influencers can recruit workers from existing markets by posting a task that asks workers to complete a different task. They may or may not get paid for the task they are asked to attack, depending on the request, but the net payment for completing the attack must be at least as high as that paid for the original task to incentivize opportunistic workers. Groups of individuals may also coordinate attack when it is in line with their beliefs or interests. For example, a group of users from 4Chan, Anonymous, or other similar community could attack a crowd-powered system as they have previously done to websites and service providers with denial of service attacks.

Web service attacks are frequently carried out using automated bots. While this represents a manipulation threat to systems relying on agreement, bots are relatively easy to detect in most crowd systems because the tasks require human understanding. When they do not, it is possible to add a ReCAPTCHA (www.google.com/recaptcha) task to check for worker authenticity.

Allowing workers to self-select tasks based on preference enables malicious workers or groups to target specific tasks and all give the same response. One way to avoid this is to directly route workers to particular tasks. Routing is supported by market places such as MobileWorks (www.mobileworks.com), and can be implemented on top of platforms that otherwise use self-selection [16, 17, 29]. Successful implementation of routing requires learning about individual workers' interests and capabilities for different tasks. To take advantage of this approach on an individual basis the requester must have enough active tasks to make the chance of returning to the same one very low.

These measures, however, only begin to address the threats posed to crowd systems as the data used by crowd systems and their outcomes become increasingly valuable. We now look more closely at the underlying vulnerability of crowd systems to information extraction and answer manipulation.

INFORMATION EXTRACTION

We call the threat of leaking private or sensitive information in tasks to others by posting tasks to crowd platforms the threat of *information extraction*. For instance, using a crowd captioning service such as Legion:Scribe [16] might result in letting workers hear a phone number, personal detail, or company secret. We discuss three types of potential types of threats related to workers extracting information from tasks: the thread of exposure, exploitation, and reconstruction.

Exposure

In some cases, a crowdsourcing task may contain information private or sensitive enough that were any worker to be *exposed* to the content, it would be harmful. For instance, a person using the crowd to help label and categorize their personal photos might consider an inappropriate or revealing picture to fall into this category. Exposure threats cannot be prevented by filtering workers since exposing the content even to honest workers is considered damaging. However, because the damage increases with each new worker exposed to the content, it can be limited by minimizing the number of workers that the information is shown to in situations where the content cannot be filtered prior to exposure to the crowd.

Exploitation

Often the concern with exposure of information is that the information could be exploited by the worker either for their own benefit or to harm the owner of the information. We call such threats as *exploitation* threats. For instance, a blind VizWiz user who turns to the crowd for image labeling might accidentally take a picture that includes their home address or credit card number. If a malicious worker were shown this image, that worker may steal this personal information. The difference between exposure and exploitation is that while exposure risks are always considered harmful, regardless of the worker, not all workers will actively exploit information.

Given no other knowledge of the workers, each worker hired for a task equally increases the risk of exploiting the information presented in the task. In such cases the risk of information explication grows linearly with the number of workers, as is the case with exposure. There may be opportunities for limiting exploitation threats if maliciousness of workers can be predicted.

Reconstruction

For some tasks, no individual piece of information incurs a large risk for the requester. For example, individual words in a private document usually do not hold enough context or meaning independent of the rest of the document to allow the worker to glean any valuable information. However, the more information that is revealed, the larger the potential risk that individually revealed pieces of information will come together to expose potentially harmful information about the requester. For instance, while information about a user's ZIP code, gender or date of birth alone reveals very little about them, knowing all three can be used to uniquely identify 87% of people in the United States [30]. We call such attacks as *reconstruction* attacks, since information extracted from individual tasks need to come together to cause damage for the requester.

Reconstruction attacks differ from exploitation and exposure attacks in that the harm to the requester (or benefit to the worker) grows non-linearly. Typically, the potential harm will grow either super linearly, or as a step function, meaning information either builds on prior knowledge to be more revealing (e.g., words in a document), or a certain subset of information must be recovered before anything important can be known (e.g., multiple partial images of a credit card number). This case differs from the previous cases because the harm from a set of revealed information is greater than the sum of the risks of each piece, meaning the risk grows *non-linearly*.

Prior examples of this type of attack in other domains have also shown that combining recovered records with other information can also result in an even larger privacy leak. For example, the information extracted from the Netflix challenge dataset, AOL search query logs, and Massachusetts medical records all appeared to preserve user anonymity on their own, but when joined with external databases, yielded personally identifying information.

Because the risk grows super-linearly due to the increase in context, the threat for requesters is significantly higher for group attacks. Reconstruction tasks with large numbers of individual pieces almost always require coordinated groups of workers (or bots) to successfully attack, since the number of tasks that must be viewed to cause harm can be high. By using a large enough group of workers, malicious entities can recover sizable portions of the information posted by a given task, even when the rate of workers outside of their group taking tasks is high.

Figure 3. Ambiguous text that could read in many ways, including as *gun*, *sun*, *lun, or fur*. The actual text is *fun*.

Figure 4. Easily read handwriting example which was used in our experiments. The text is *length*.

ANSWER MANIPULATION

Another type of threat to users of crowd-powered systems is that the answers they receive are intentionally incorrect due to workers manipulating the answers they provide. In this section, we discuss three areas of concern regarding workers manipulating tasks: classic manipulation, disruption, and corruption.

Classic Manipulation

Classic manipulation threats are ones where the worker or a group of workers changes the outcome of a task to reflect a particular outcome that is different from what the requester is looking for. We borrow the term *manipulation* from the breadth of research in election theory that strives to prevent this type of situation, in which one party attempts to attain undue influence [24]. Because most crowd systems use voting or other decision aggregation approaches to ensure quality, classic manipulation requires workers to comprise a large enough portion of the final contributing set to ensure the answer they want wins. Since this is not possible to do alone, workers must collaborate with others to successfully accomplish their goal.

Disruption

Another type of attack aims to simply disrupt the crowd-powered service. This means that the goal is to make the final result either any incorrect answer, or not allow the system to reach sufficient consensus to provide an answer at all. For tasks that use aggregation as a quality measure, disruption also requires a group of malicious workers, but it may not require them to coordinate on a fixed manipulation strategy. The difference between classic manipulation and disruption is the existence of an intended controlled outcome versus any outcome that stops the system from giving a valid response.

Corruption

Some crowdsourcing tasks ask sets of workers to contribute input, and then ask a different group to determine if that contribution was helpful or progressed the task. This process is repeated over and over with different workers until a final answer is reached (e.g., Wikipedia, TurKit [14]). Since these systems do not require that the entire crowd that contributed to a given piece be present throughout the lifetime of the decision process, the malicious input of a relatively very small group of workers can potentially destroy the progress made on the entire task.

Corruption can be manipulative or disruptive, and occurs in iterative workflows in which a small group can undo the progress of a much larger set if they can manipulate just one contribution and the corresponding verification step. To prevent corruption Wikipedia uses change logs that allow people to easily revert to prior versions. However, to the best of our knowledge most iterative workflows that run using Mechanical Turk do not protect themselves in this manner, meaning corruption of a task at any given point can prevent future good workers from repairing the damage.

RAISING AN ARMY

While attacks on a crowd-powered system conducted by an individual require no coordination of effort or motivation, many of the most harmful potential attacks on crowd-powered systems involve coordinated effort by multiple workers. Group attacks can be carried out by a number of malicious workers organizing around a common objective. Such attacks can also be organized by a malicious entity directing a group of not necessarily malicious workers to execute an attack. This type of group attacks organized by a malicious entity may pose threats for the worker community since workers may be directed to carry out tasks with ethical or legal problems.

To understand how vulnerable crowd systems are to extraction and manipulation attacks, we conducted several experiments where we actively attacked a task of our own creation. We used crowds of people, recruited via the same crowdsourcing platform (in our case Mechanical Turk), to carry out the attacks. In doing so, our goal was to identify whether such attacks were feasible, and to build a picture of the different roles crowd workers take when asked to attack another task. We identify the following worker types as interesting to our analysis:

- **Passive workers** are not looking to find or capture any content. These workers might view unintended information (e.g., nude images), creating an exposure risk, but they will perform the task honestly and not misuse any information they are exposed to.
- **Opportunistic workers** do not actively seek sensitive information or chances to harm a task, but will seize such opportunities when presented. They may exploit the information they encounter, or perform a task incorrectly if paid to do so.
- **Malicious workers** actively seek information that they can exploit or try to manipulate tasks for personal gain. They may, for example, look for image recognition tasks that provide access to sensitive information.

Condition	Return Rate
Baseline	73.8%
Innocent	62.1%
Malicious	32.8%

Table 1. Extraction experiment return rates. Significantly fewer ($p < .0001$) workers were willing to copy task information to us when we asked a question that contained information that looked potentially sensitive.

Condition	Return Rate
Baseline	73.8%
Innocent	75.0%
Malicious	27.9%

Table 2. Manipulation experiment return rates. Significantly fewer ($p < .0001$) workers were willing to follow the manipulation instructions given in the attack task when the instructions clearly directed them to answer incorrectly.

- **Beneficent workers** are willing to go beyond completing the task correctly to further help the requester, instead of following economic incentives alone. This has been observed in terms of feedback on task design (including in this work), and reporting workers they see doing a task incorrectly [20].

As we vary the appearance of the attack we coordinate in our experiments in terms of maliciousness, we are able to identify passive workers (who will not complete tasks they see as potentially harmful), opportunistic or malicious workers (who are willing to complete potentially harmful tasks), and beneficent workers (who go above and beyond and contact us regarding the attack task).

EXPERIMENTS

Our goal was to test how viable it is to recruit a group of crowd workers to attack a different crowdsourcing task for us. We posted the two tasks under different requester names. In the first task (referred to as the *attack* task), workers were directed to follow a link to another task (the *target* task) that was visible on the crowdsourcing platform. This setup is illustrated in Figure 5. The target task always requested workers type the text they saw in an image. For example, users were asked to type the text shown in Figures 3 (perhaps "fun", "lun", "sun", or "gun") and 4 ("length") or transcribe the text given on a credit card.

Since a worker's hesitation to complete the attack task could possibly draw heavily on a moral component, we designed the attack task to either appear innocent or harmful (malicious). For the malicious condition of the extraction study, workers were asked to extract real-looking credit card information from the target task and return it to us. In the manipulation study, workers were asked to provide a clearly incorrect answer to the target task.

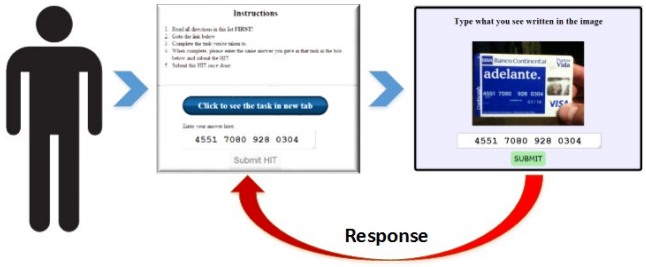

Figure 5. Layout of our information extraction tests. Initially the worker accepts our task, but then is redirected to another task where they are asked to return the answer they gave.

All workers were unique, and both tasks paid $0.05. In total, 1,140 workers completed our task, with at least 140 workers participating in each condition. In the baseline condition, the attack task directed workers to the target task and asked them to report back after completion by clicking a "done" button without extracting information or manipulating the target task. 73.8% of workers completed this baseline condition.

Extraction Study

Our first experiment tested whether or not crowd workers were willing to retrieve information and return it to us. We used the following conditions in the extraction experiments:

- **Baseline:** We used the same baseline described above.
- **Innocent:** Workers were shown a computer-generated picture of a credit card and asked to transcribe the text content. While this card had as much information on it as a real credit card (name, number, expiration date, etc.), the information was clearly fake (e.g. the card number was "01234[…]" and the name was "Card Holder") and the picture was clearly not a real image.
- **Malicious:** To test the effect of perceived intentionality on the crowd, we ran the same extraction task with a photograph of a real-looking (but still fake) credit card. We expect this to appear to attentive workers as more of a threat than the artificial image.

We measured the rate of attack task response relative to the target task. In other words, we looked at how many workers who completed the task were willing to return the answer to us and get paid for the attack task. Our results are shown in Table 1. In all cases, a sizable portion of crowd workers were willing to extract information from other tasks. When the information being extracted appears innocuous, almost as many people (no significant difference) returned to provide the extracted information as they do when they are not asked to extract anything (62.1% v. 73.8%, $p > .1$). Even when asked to extract credit card information (the

malicious case), workers were willing to do so 32.8% of the time, but this is a significant drop in return rate ($p < .0001$). We also saw a significant reduction in return rate (62.1% v. 78%) when completing the credit card task compared to the same baseline using the handwriting task ($p < .01$).

Manipulation Study

We also conducted a set of experiments to find out if workers were willing to actively manipulate a task based on our instructions. We again divided the conditions into two levels of potential harm and tested the following conditions:

- **Baseline:** We used the same baseline described above.
- **Innocent:** To see if the crowd would follow directions that overrode the instructions in a different task, we gave them a specific response to provide the target task in the instructions of the attack task. We used a handwriting example that was unclear what the true answer was (Figure 3), and provided a plausible solution (for this ambiguous task, workers were instructed to type "sun" when the correct answer was "fun"). We also ran another condition that asked workers to label the same image without instructions in order to get a baseline so we could detect a manipulative effect in the results.
- **Malicious:** To see if workers are also willing to knowingly provide false information (disobey the instructions of the target task at the request of the attack task), we created another task manipulation condition in which workers asked to respond the inner task in a way that is clearly incorrect. In this condition, the target task asked workers to transcribe an image (Figure 4) and the attack task instructed them to transcribe this image as "sun" (the answer is "length").

Our results, shown in Table 2, again demonstrate that people are willing to perform tasks that act on other tasks. As many people (no significant difference) were willing to complete the target task when instructed to manipulate it in an innocent fashion as were willing to complete the task when given no additional instructions (75.0% v. 73.8%, $p > .4$. Interestingly, when the attack task instructed to give a clearly incorrect response to submit, a significant portion of the crowd refused to comply with the request to complete a task obviously incorrectly. We saw a significant ($p < .0001$) decrease the response rate for the malicious condition. We also want to know what those who did complete the task submitted, and what the final effect on the system was. For the malicious manipulation task, 28% who saw the word "length" (Figure 4), wrote "sun" as instructed. The rest correctly labeled the image despite initial instructions. Our results are shown in Table 3. This suggests that workers might be subject to intentional external biasing when they do not perceive the answer as obviously wrong.

Summary

Our results suggest that it is possible to mobilize the crowd to attack other tasks. When an attack task appears innocent to workers, they are as willing to complete the task as if they were not asked to extract or manipulate the target task. Even when the attack task appears likely to do harm (e.g., extracts credit card information or enter incorrect information), a significant portion of the crowd is willing to complete it, suggesting there are significant vulnerabilities for crowd systems to coordinated attacks.

Response	Baseline ("fun")	Innocent ("fun")	Malicious ("length")
"Sun"	12%	**75%**	28%
"Gun"	**36%**	16%	-
"Fun"	26%	7%	-
"Lun"	14%	1%	-
"Jun"	9%	1%	-
Other	3%	-	-
"Length"	-	-	**72%**

Table 3. Results from the manipulation experiment. While "gun" is the most commonly guessed word for this example when no instructions were given, the most popular answer in the innocent manipulation conditions is "sun", as instructed.

It is notable, however, that significantly fewer workers were willing to complete the attack task when it appeared malicious as when it appeared innocent, both in the extraction case and in the manipulation case. This suggests that some crowd workers can recognize the attack and object to it. Nonetheless, it appears possible to manipulate even these passive workers to act in ways that they would not intend if it the task appears innocuous, as many more workers gave different responses to the target task than they might naturally when given a plausible response.

DISCUSSION: SECURING AGAINST ATTACKS

Our findings demonstrate the vulnerability of existing crowdsourcing practices to extraction and manipulation attacks. Although there have been efforts for quality control in the presence of error-prone workers, little has been done on protecting these systems from the kind of threats studied in this paper. We believe that this same focus can and should be given to preventing systems from information extraction and from worker manipulation as well.

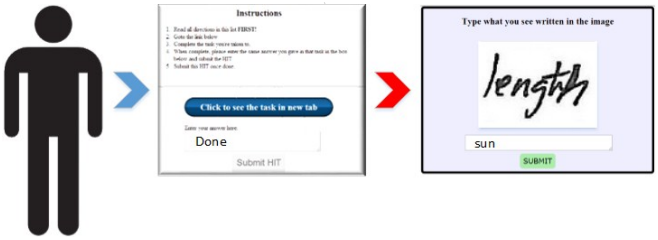

Figure 6. Layout of our information extraction tests. Initially the worker accepts our task, but then is redirected to another task where they are asked to provide a specified answer.

While our study showed a significant effect of changing the target task's content on workers' willingness to complete the attack task, our results do not show the exact reason why. In future work we would like to be able to isolate workers' reasons for participating and not participating in malicious extraction and manipulation tasks. Similarly, exploring trade-offs in worker motivation, such as the price paid for completing the target versus attack tasks, or the purpose of the task being completed, might impact workers' willingness to partake in potentially harmful actions. For example, if a worker was well paid to help a blind user in a system such as VizWiz, it might be the case that they are less likely to return user information to an attack task.

Our study was also carried out exclusively on Mechanical Turk, but provides a method of testing other crowdsourcing platforms that attain significant usage in the future.

Automation for Privacy Preservation

Previous work presented task-specific approaches for limiting information extraction from crowdsourcing tasks. As shown in Figure 3, face detection algorithms can be used to automatically detect and cover faces to protect privacy. Similar approaches can be used for text if the structure of private or sensitive information is known (e.g., SSN, credit card numbers and addresses). Another approach is dividing tasks into smaller pieces to limit information extraction. For example, the work by Little and Sun [15] proposed dividing images into pieces for protecting privacy. Similar approaches have been applied to OCR tasks such that no one worker gets the whole word, but multiple workers can each transcribe a set of letter that can then be recombined into the correct word for the requester. Such approaches help to prevent against extraction attacks from individuals. However, they are vulnerable against coordinated group attacks and may diminish performance of tasks that require contextual information about the entire task to be able to produce a solution. More attention is needed to generalize automated approaches for real-world tasks and offer solutions applicable to tasks that require context about the entire task.

Leveraging Reliable Workers

The results of our experiments showed that not all workers behave the same in extraction and manipulation attacks, especially when the nature of the attack is malicious. There are opportunities for designing workflows that can utilize reliable workers for early detection of information extraction and manipulation attacks. For instance, workers can be instructed to alert requesters about the existence of private or sensitive information. Iterative workflows can be designed to gradually release tasks – first to reliable workers and then to a larger worker pool based on feedback from the initial set of workers.

Reliable workers can also be of use to protect against manipulation attacks. As shown by our analyses, existing quality control approaches are effective to protect against individual manipulation attacks. Since workers in current

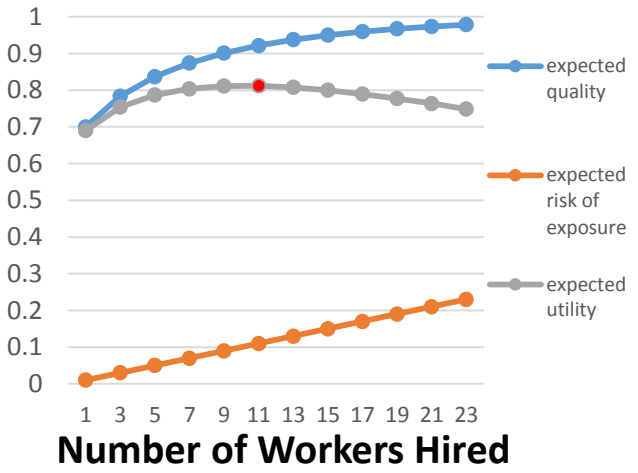

Figure 7. Tradeoff between expected quality of final decision, and the expected risk of exposing information to the crowd. The red dot at X=11 represents the number of workers that maximizes the expected utility for this example user.

crowdsourcing systems are mostly anonymous, large-scale manipulation attacks may need to reach workers with open calls as demonstrated in our experiments. Reliable workers that are aware of such attacks can alert requesters about manipulation attacks and aggregation mechanisms used for quality control can be adjusted accordingly. Workers can also be incentivized to report such attacks, helping to leverage opportunistic workers for the benefit of the task.

Decision-Theoretic Analysis for Privacy Preservation

A common approach for quality control in crowdsourcing systems is hiring multiple workers for a task to eliminate the effect of errors from individual workers on the final result. As discussed above, hiring more workers increases the risk of information extraction. For instance, the risk of exposure may grow linearly in the number of workers hired for a task. This observation highlights a trade-off between higher quality output and higher risk of information extraction. Figure 7 shows an example of this trade-off. Given the probability of a task containing private or sensitive information, the risk of exposure (the expected number of workers exposed to the content) grows linearly with the number of workers hired for the task in this example. Based on majority voting applied for deciding the correct answer of the task, the expected quality (the probability of correctly identifying the answer) also grows with the number of workers hired. For a system that equally weights the utility for correctly identifying the answer and the cost for risking exposure, the expected overall utility of the system is also displayed in the figure. In this particular setting, hiring 11 workers (marked red in Figure 7) for a task maximizes the system utility. This case demonstrates a quality-risk trade-off, and highlights opportunities for designing dynamic decision-theoretic policies that reason about the risk of manipulation and extraction and use trusted workers in its workflow to minimize these threads.

CONCLUSION

In this paper, we studied the vulnerability of existing crowdsourcing practices to information extraction and manipulation threats from individual workers and groups. We demonstrated with experiments on the Mechanical Turk platform that a simple task design is sufficient to perform both an information distraction and manipulation attacks and workers have fewer tendencies to participate when tasks appear to be more malicious. We then outlined future directions for making crowdsourcing systems more resilient to these attacks.

As crowdsourcing becomes an integral component of many systems, threats such as the ones studied in this paper pose a significant danger. This paper is a first step towards understanding the viability of these threats as well as the behaviors of workers in their presence. We hope that gaining more understanding of these threats will influence further efforts in the future for more secure and resilient crowd-powered systems.

REFERENCES

1. Bernstein, M.S., Brandt, J.R., Miller, R.C. and Karger, D.R. Crowds in two seconds: Enabling realtime crowd-powered interfaces. In *Proceedings of UIST 2011.*
2. Bernstein, M.S., Little, G., Miller, R.C., Hartmann, B., Ackerman, M.S., Karger, D.R., Crowell, D. and Panovich, K. Soylent: A word processor with a crowd inside. In *Proceedings of UIST 2010*.
3. Bigham, J.P., Jayant, C., Ji, H., Little, G., Miller, A., Miller, R.C., Miller, R., Tatarowicz, A., White, B., White, S. and Yeh, T. VizWiz: nearly real-time answers to visual questions. In *Proceedings of UIST 2010*.
4. Chen, X., Bennett, P.N., Collins-Thompson, K. and Horvitz, E. Pairwise ranking aggregation in a crowdsourced setting. In *Proceedings of WSDM 2013*.
5. Deng, J., Krause, J., and Fei-Fei, L. Fine-Grained Crowdsourcing for Fine-Grained Recognition. In *Proceesings of CVPR 2013*.
6. Evanini, K. and Zechner, K. Using crowdsourcing to provide prosodic annotations for non-native speech. In *Proceedings of Interspeech 2011*.
7. Featured requester: Discover how comScore benefits from mechanical Turk. *The Mechanical Turk Blog*, March 2013. http://bit.ly/Ye64Sb
8. Harris, Christopher G. Dirty Deeds Done Dirt Cheap: A Darker Side to Crowdsourcing. In *SocialCom* 2011.
9. Kamar, E., Hacker, S., Lintott, C. and Horvitz, E. Combining human and machine learning intelligence in large-scale crowdsourcing: Principles, methods, and studies. *MSR-TR-2012-58*, 2012.
10. Kittur, A. and Kraut, R.E. Harnessing the wisdom of crowds in Wikipedia: Quality through coordination. In *Proceedings of CSCW 2008.*
11. Kittur, A., Nickerson, J.V., Bernstein, M., Gerber, E., Shaw, A., Zimmerman, J., Lease, M., and Horton, J. The future of crowd work. In *Proceedings of CSCW 2013.*
12. Kokkalis, N., Köhn, T., Pfeiffer, C., Chornyi, D., Bernstein, M.S. and Klemmer, S.R. EmailValet: Managing email overload through private, accountable crowdsourcing. In *Proceedings of CSCW 2013.*
13. Le, J., Edmonds, A., Hester,V., and Biewald, L. Ensuring quality in crowdsourced search relevance evaluation. In Proceedings of SIGIR 2010 Workshop on Crowdsourcing for Search Evaluation.
14. Little, G., Chilton, L.B., Goldman, M. and Miller, R.C. TurKit: Human computation algorithms on mechanical turk. In *Proceedings of UIST 2010.*
15. Little, G. and Sun, Y-A. Human OCR. *CHI 2011 Workshop on Crowdsourcing and Human Computation.*
16. Lasecki, W.S., Miller, C., Sadilek, A., Abumoussa, A., Borrello, D., Kushalnagar, R. and Bigham, J.P. Real-time captioning by groups of non-experts. In *Proceedings of UIST 2012*.
17. Lasecki, W.S., Murray, K.I., White, S., Miller, R.C. and Bigham, J.P. Real-time crowd control of existing interfaces. In *Proceedings of UIST 2011.*
18. Lasecki, W.S., Song, Y.C., Kautz, H. and Bigham, J.P. Real-time crowd labeling for deployable activity recognition. In *Proceedings CSCW 2013*.
19. Lasecki, W.S., Thiha, P., Zhong, Y., Brady, E. and Bigham, J.P. Answering Visual Questions with Conversational Crowd Assistants. In *Proceedings of ASSETS 2013*.
20. Lasecki, W.S., Wesley, R., Nichols, J., Kulkari, A., Allen, J.F. and Bigham, J.P. Chorus: A Crowd-Powered Personal Assistant. In *Proceedings of UIST 2013*.
21. Lasecki, W.S., White, S.C., Murray, K.I. and Bigham, J.P. Crowd Memory: Learning in the collective. In *Proceedings of Collective Intelligence 2012*.
22. Mason, W. and Watts, D.J. Financial incentives and the performance of crowds. In *Proceedings of HComp 2009.*
23. Massively multiplayer pong. http://collisiondetection.net, 2006.
24. Menton, C. and Singh, P. Manipulation can be hard in tractable voting systems even for constant-sized coalitions. *CoRR* abs/1108.4439.
25. Sun, Y-A. Roy, S. and Little, G. Beyond independent agreement: A tournament selection approach for quality assurance of human computation tasks. In *HComp 2011.*
26. Sweeney, L. Uniqueness of simple demographics in the U.S. population. In LIDAP-WP4, CMU. 2000.
27. Quinn, A.J. and Bederson, B.B. Human computation: A survey and taxonomy of a growing field. In *Proceedings of CHI 2011*.
28. Van Boskrik, S. US interactive marketing forecast, 2011 to 2016. *Forrester*. August 2011.
29. Von Ahn, L. Games with a purpose. *Comp.* 39(6), 2006.
30. Zaidan, O.F. and Callison-Burch, C. Crowdsourcing translation: professional quality from non-professionals. In *Proceedings of ACL-HLT* 2011.