

Error and attack tolerance of collective problem solving: The DARPA Shredder Challenge

Nicolas Stefanovitch^{1†}, Aamena Alshamsi^{1†}, Manuel Cebrian^{3,4} and Iyad Rahwan^{1,2*}

*Correspondence:
irahwan@acm.org

¹Masdar Institute of Science and
Technology, Abu Dhabi, 54224, UAE

²School of Informatics, University of
Edinburgh, Edinburgh, UK

[†]Equal contributors

Full list of author information is
available at the end of the article

Abstract

The Internet has unleashed the capacity for planetary-scale collective problem solving (also known as crowdsourcing), with ever increasing successful examples. A key hypothesis behind crowdsourcing is that, at a critical mass of participation, it has the capacity not only to agglomerate and coordinate individual contributions from thousands of individuals, but also to filter out erroneous contributions, and even malicious behavior. Mixed evidence on this front has arisen from limited observational data and controlled laboratory experiments with problems of moderate difficulty. We analyze behavioral data from our participation in the DARPA Shredder Challenge, an NP-hard combinatorial puzzle beyond computational reach, which involved 3,500 participants from five continents over three consecutive weeks. We study thousands of erroneous contributions and a number of large-scale attacks, and quantify the extent to which the crowd was able to detect, react, and recover from them. Whereas the crowd is able to self-organize to recover from errors, we observe that participants are (i) unable to contain malicious behavior (attacks) and (ii) the attacks displayed persistence over the subsequent participants, manifested in decreased participation and reduced problem solving efficiency. Our results raise caution in the application of crowdsourced problem solving for sensitive tasks involving Financial Markets and National Security.

Keywords: crowdsourcing; collective intelligence; error and attack tolerance

1 Introduction

Crowdsourcing [1] allows us to harness human intelligence and skills in order to solve problems beyond the reach of current algorithms and computers [2–15]. Solving a problem using crowdsourcing typically proceeds through a dedicated Web site or social media outlet, and inviting almost any willing individual over the Internet to join in and contribute. The system collects answers from these individuals (volunteers or paid workers) from the crowd and combines their answers into a complete solution. The crowd of incentivized users can either parallelize and speedup the completion of a batch of tasks [16], or solve problems beyond the reach of any individual even if given sufficient time to do so [17].

A successful crowdsourcing platform requires a sufficiently large pool of users who actively engage their time and effort. In order to build such a user base, Web sites need to incentivize people to participate, for example using small payments. Among the platforms

following this principle, *Amazon Mechanical Turk* is the best known. Crowdsourcing projects relying on financial rewards have to satisfy time, budget and quality constraints [18–22].

In many cases, people are also willing to participate freely in a crowdsourcing project when they see it as fun, or useful to a greater good. Examples of crowdsourcing projects based on voluntary work are *Wikipedia*, the collaboratively edited encyclopedia, *FoldIt*, a gamification of protein folding for biomolecular research [23] and *Tomnod*, a company specialized in rapid natural disaster response by providing crowd-based satellite image analysis [24, 25].

Crowdsourcing systems adopt different incentives and algorithmic strategies to both enhance the quality of the collective endeavor and to protect it from malicious behavior [14, 19–22, 26–43]. However, there is no incentive scheme that is foolproof. Users might commit errors either due to lack of sufficient motivation, due to honest mistakes, or as part of trial-and-error exploration of the task.

A typical approach to dealing with such errors is to replicate a single task, then aggregate the answers of several users, for example using fusion algorithms [44]. The simplest form is to average numerical answers, but sophisticated machine learning techniques can also be used to cluster answers and detect outliers [36]. Some platforms, such as *Wikipedia*, tolerate errors assuming that erroneous contributions will be corrected by other genuine users after some time [45].

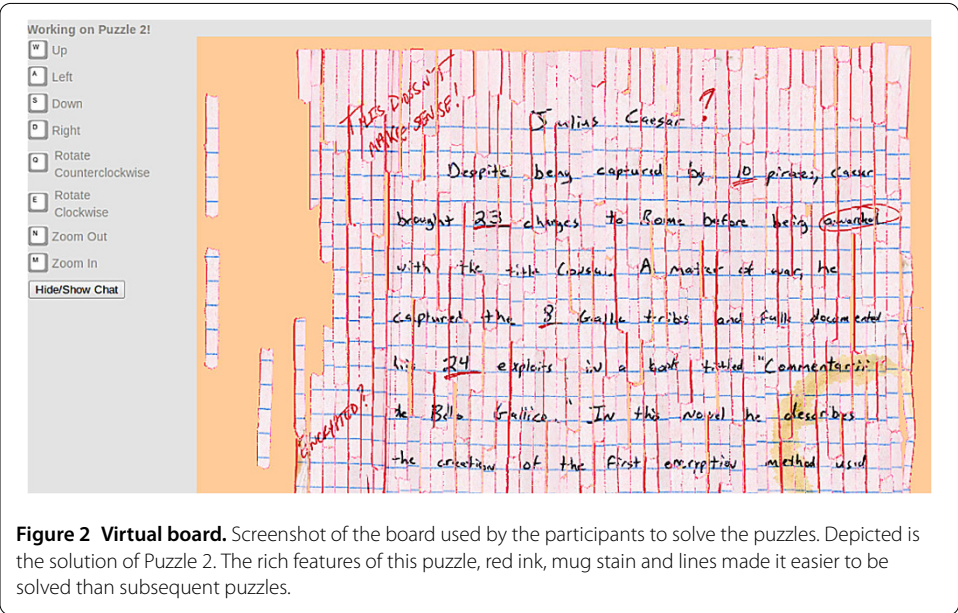
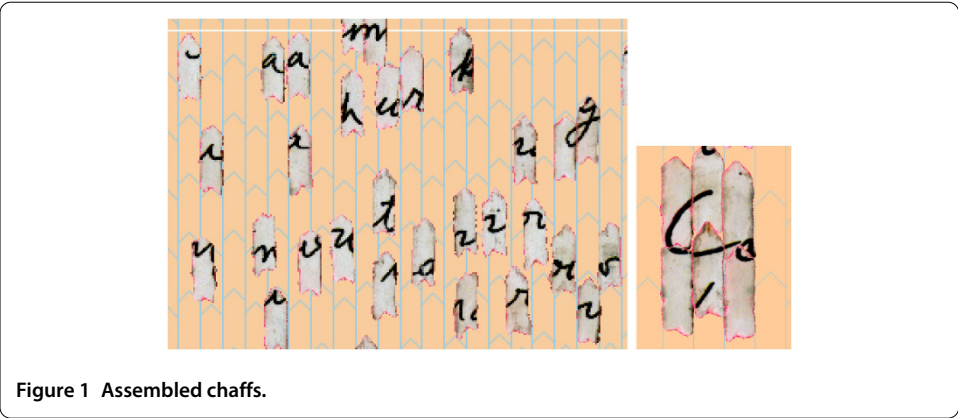
Deviation, however, seems possible and quite pervasive. In the context of paid crowdsourcing, users often exert the minimum effort possible in order to secure the pay. This form of deviation is well understood. Platforms like *Mechanical Turk* employ several safe guards to counter it, such as a reputation system that can be used to filter workers who do not meet quality standards [46].

While errors and deviations from incentivized users received much attention, an important and largely unaddressed problem is dealing with users willing to invest time and resources to actively attack a crowdsourcing system [47–51].

Crowdsourcing systems are particularly vulnerable to attacks due to their accessibility, allowing attackers to easily feed the platforms with false information [49, 52–55]. The increasing appeal of crowdsourcing solutions from governments and companies exposes them to the risk of suffering deliberate attacks from competing or adverse agents. Crowdsourcing can also be used to promote distributed denial of service attacks performed by politically [56] or financially [57, 58] motivated attackers.

In this paper, we study the capacity of a crowdsourcing system to cope with errors, lone-wolf attacks, and distributed attacks. We use data collected during a month-long crowdsourcing effort led by the University of California, San Diego (UCSD) to solve the DARPA Shredder Challenge [59]. In this challenge, contestants were asked to solve five puzzles of increasing difficulty. Puzzles were the result of documents being processed through different commercial shredders. A prize of \$50,000 was at stake for the first team able to answer a set of questions that required re-assembling the shredded documents.

Solving a jigsaw puzzle is known to be an NP-complete problem [60], and requires highly specific cognitive capabilities and observational skills (e.g. recognizing matching patterns in images, and using context to infer likely word completions) as well as tedious repetitive work. As such it lends itself ideally to crowdsourcing.



The UCSD team was not the only team to use crowdsourcing [61]. It is, however, the only one that succeeded at using an open platform, while all other teams kept their solution and methods closed. The UCSD team provided users with a virtual puzzle board, depicted in Figures 1 and 2, over which anybody could directly modify the solution state. In doing so, the team was able to leverage the collective capacity of the crowd [62]. Users were incentivized to contribute effort and to recruit additional users by dividing the prize using a split contract scheme in case of winning [35].

The UCSD team entered late (two weeks into the challenge) but was able to solve the first three puzzles in 4 days. The team managed to reach the same level of achievement as the top contenders, reaching as far as the second place on the scoreboard put in place by the challenge organizer to keep track of progress.

Members of a competing team subsequently attacked the UCSD platform. They exploited the platform's openness by making deliberate detrimental moves. The system underwent a series of such attacks of different scale, span and nature over three days. Despite the short span of the attacks, the platform was unable to recover, progress stalled and no other puzzle could be successfully assembled.

One day after the attacks, the attackers sent an anonymous email to the UCSD team summarily describing their intentions, actions and how they changed their attack strategy to defeat the protective measures put in place by the team and exhibited by the crowd. Noteworthy is their claim to have recruited a crowd of attackers through *4chan*, an anonymous image hosting and discussion board with a large and reactive following [63] that is regularly involved in massive distributed pranks and militant attacks [64].

To our knowledge, this is the first detailed dataset documenting an attack on a deployed crowdsourcing system. The dataset provided a rare opportunity to explore the behavior of crowdsourcing under two different regimes (i) *normal*: when only genuine users interact with the system; and (ii) *attack*: when one or more attackers are active. While it is impossible to obtain perfect understanding of the underlying processes behind the crowd's behavior, we made substantial progress in this direction, and drew several key observations that can inform future crowdsourcing deployment.

To identify suspicious behavior, we developed a set of domain-independent features and domain-specific quality measures to evaluate the behaviors of individual users and the crowd. In the normal regime of puzzle assembly, we find that only a small proportion of users (at most 10%) drove most of the progress. We also found that recovery from errors follows a long-tailed distribution with an exponential cut-off. This implies that the crowd can cope with small perturbation with high efficiency.

Under the attack regime, we find that just few attackers are able to inflict considerable damage even when numerous genuine users react in real-time, giving the upper hand to attackers. However, we could not find conclusive evidence that the platform sustained a distributed attack from a large number of recruited attackers. This has an important consequence that only motivated attackers can be harmful.

Moreover, we find that attacks on crowdsourcing systems are successful not because they destroy progress made by the crowd, as this can be easily restored from backups. Rather, it is attackers undermine the confidence of the crowd, eventually leading to the destruction of the user base. In particular, we find an order of magnitude decrease in activity subsequent to the attacks on the platform.

Crowdsourcing systems are complex socio-technical systems [5], whose behavior is determined by technical design, individual human cognition, and social interaction among workers. The DARPA Shredder Challenge data provides a rare glimpse into some of those dynamics, particularly in relation to error and attack tolerance. We believe that the insights gained from this study are important in the development of future crowdsourcing platforms where state security or money are at stake. Our study highlights the importance of taking into account the security of crowdsourcing systems at the core of its design, and motivates the need to adapt reputation systems for these settings [65].

2 Results

2.1 Crowdsourcing under normal regime

The crowdsourcing approach to puzzle-solving proved effective as it was able to solve the first three puzzles within almost one day each. It was able to do so despite the necessary transient errors of users exploring different combination of pieces. This section describes the mechanisms behind this.

Crowd behavior is the combination of the individual actions of users in a common space, a virtual board in our case. In order to study it, we use several ad-hoc measures related to

puzzle	pieces	edges	moves	span (h)	man hour (h)	users	users >0
1	221	257	6200	11.87	58.91	94	41
2	374	835	19224	10.22	159.91	558	228
3	1137	650	79658	49.07	523.32	1011	224
4b	2306	189	49834	38.24	243.11	367	121
4a	2306	219	79846	335.68	514.03	544	59

Figure 3 Puzzle characteristics and crowd performance statistics. *Pieces* is the number of pieces in a puzzle, *edges* is the number of correctly assembled edges - which correspond for the first three puzzles the minimum number of edges required to solve the puzzles, *moves* is the total number of moves, *span* is the difference in time between the last and first move on a puzzle, *man hour* column is the cumulative total number of hours spent on the system collectively by the users, *users* is the total number of users participating in a puzzle, *users > 0* is the number of users which assembled at least one correct edge. The last two rows describe two separate periods of Puzzle 4: before (4b) and after (4a) the attacks.

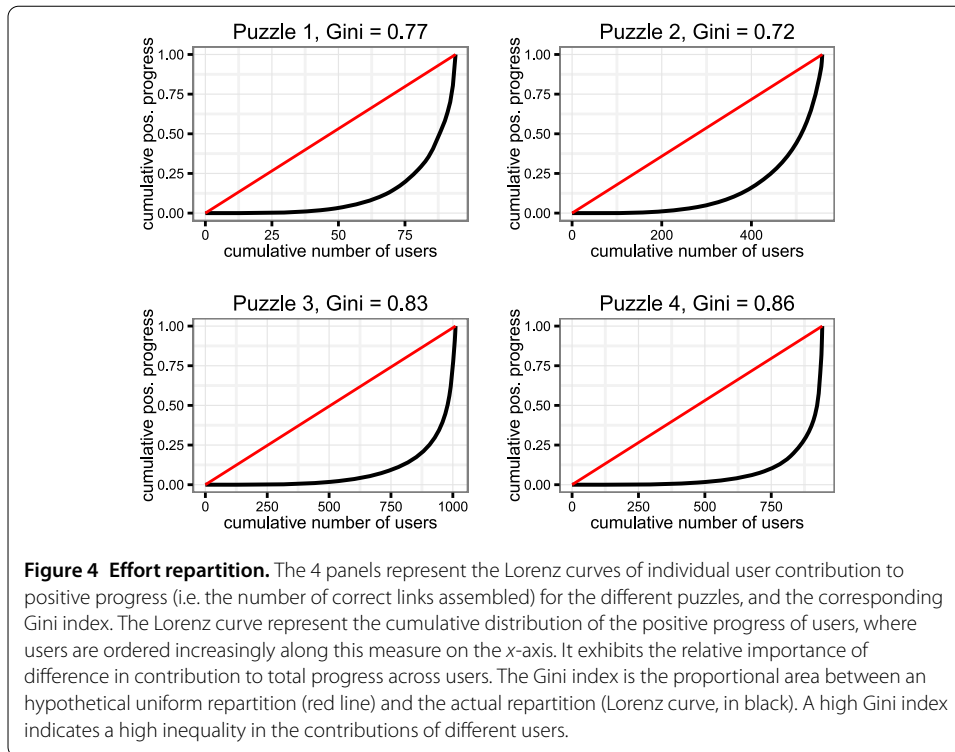
the evolution of cumulative achievement. Of particular importance are the *progress* and *exploration* measures, which are related to the linking and unlinking of correct (respectively, incorrect) links between pieces. The cumulative or instantaneous value of these measures will be used throughout the paper (refer to Section 4 for details on their computation).

Characteristics of the puzzles and statistics of the crowd performance are given in Figure 3. While Puzzle 4 was never solved as a consequence of the attacks, we report the statistics relative to the two longest stable periods: before (4b) and after the attacks (4a). We focus in this section only on the first three puzzles. The impact of attacks is the topic of Section 2.2.

2.1.1 Crowd efficiency

While the complexity of solving a puzzle increases with the number of pieces to be assembled, this measure fails to capture actual difficulty. First, puzzles possess an increasing number of characters written in a decreasing font size. Second, puzzles differ in the number and richness of extraneous features (e.g. different colors, stains, line markers) that add significantly to the difficulty. Third, the puzzles need to be solved only to the extent that they can provide the answer to the set of questions laid by the organizers, possibly requiring only a subset of the pieces to be reconstructed. For all these reasons it is not possible to precisely compare the difficulty of the puzzles. Note that Puzzle 3 is actually a sparse black and white image, with mostly featureless pieces, making it equivalent to a puzzle of a smaller size.

However, comparing the effort needed to assemble a given number of correct links provides an indirect way to compare the efficiency of a crowd. We can observe that puzzles 1 and 2 took about the same time to solve (around 12 hours). Puzzle 2 required to assemble 3 times as many edges and took 3 times more man hours, by about 6 times as many users. This is a straightforward illustration of the power of the crowd. Puzzle 3, with 2.5 times more edges than Puzzle 1, required 10 times more users developing 10 times more man hours, within 5 times the time span needed to solve Puzzle 1. This shows a non-linear increase in the complexity of solving puzzles of increasing size due to the combinatorial nature of the problem, as well as the intricate relationship between different features.

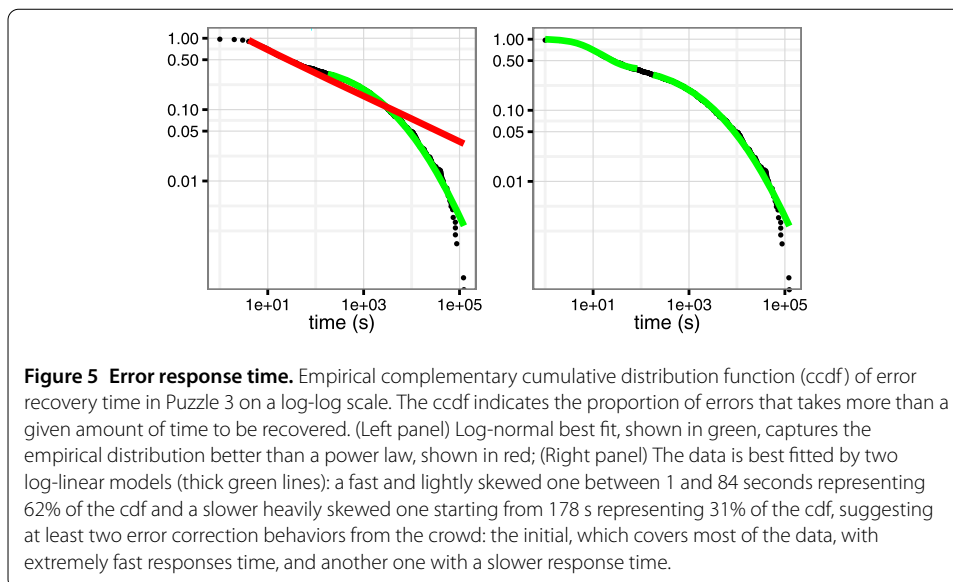


2.1.2 Work repartition

We study the inequality among users' contributions by studying the Lorenz curves [66] and respective Gini index [67] of the different puzzles, reported in Figure 4. The curves indicate large inequality of contributions across users. Consistently, half of the progress has been made by at most 10% of the users, while in puzzle 3 and 4 this ratio is close to 5% of users. Moreover, as the platform gains momentum, the Gini index increases, except for Puzzle 1, which implies that the proportion of highly active user tends to decrease as the pool of participants gets larger. This can also be seen in Figure 3, under the column of users with a positive count of correct links created: the ratio of those active users to the total number of users is decreasing, except for Puzzle 4b, which is the part of Puzzle 4 after the attack.

We can conclude that most of the progress has been achieved by a small proportion of users. While the contribution of others cannot be downplayed, it nevertheless indicates that the primary driving force behind crowdsourcing are motivated overachieving individuals. The pool of such users is scarce or hard to reach, as their proportion decreases with respect to the total number of participating users.

These results are consistent with recent results from another DARPA challenge, the Network Challenge (a.k.a. Red Balloon Challenge), which show that finding the right people to complete a task through crowdsourcing is more important than the actual number of people involved [15]. This seems to indicate a trend in crowdsourcing were it is not the crowd per se that solves the problem, but a subset of individuals with special task-specific attributes.



2.1.3 Error recovery

An error is defined here as the unlinking of a correct edge between two pieces. Correct edges are defined as belonging to hand-picked clusters of correctly assembled pieces on the final state of the board. Correct edges between these pieces can be created and destroyed several times during the solving procedure. We study the error recovery capacity of the crowd by measuring how long it takes for a destroyed correct edges to be re-created.

We focus on the recovery capacity exhibited by the crowd on Puzzle 3, which is the largest solved puzzle that did not experience an attack. Figure 5 shows the empirical complementary cumulative distribution function (ccdf) of error recovery times in Puzzle 3. The ccdf shows the proportion of errors that have been solved after a given amount of time. Most errors are recovered quickly with 79% of the errors corrected in under 2 minutes, 86% in under ten minutes, and 94% in less than an hour.

The distribution is heavy tailed which is an important factor behind the efficiency of the crowd to cope with errors. We fit a power law, log-normal and exponential models to the data following the procedure of [68], and compare them using the Vuong test [69]. The Vuong test is a likelihood ratio test able to discriminate which of two models better fits the data and is able to tell whether it can confidently do so. While close fits do not provide a proof that the data follow the best model, they nevertheless pose a basis for studying the generating process. The log-normal model is the most supported one, with a high test statistic and p -value close to zero, indicating that it is a better fit than the power law. The fitted model has a location (i.e. mean in a logarithmic scale) of 6.89 and a shape (i.e. standard deviation in logarithmic scale) of 1.92. This distribution has a mode of 24 s but is heavily skewed with a median of 27 min and a mean of 1.75 hours.

However the region fitted by this model starts at 178 s and as such represents only 31% of errors. So we performed an additional study of the previously non-fitted region of the data. The log-normal model is again supported by the Vuong test as the best fit in the range of 1 to 84 seconds, with a location of 2.4 and a shape 0.98. The upper value of the range was found by sweeping through the unfitted section of the data and selecting the value minimizing the Kolmogorov-Smirnov statistic. This model accounts for 62% of errors, and

Figure 6 Returning users. The entries above the diagonal gives the return rate of users of puzzle i (row) in puzzle j (column). The lower triangle gives the proportion of users of puzzle j (column) returning from puzzle i (row). Return rate of users in Puzzle 4 from before the attacks to after the attacks is 0.98 and 1.0 in the opposite direction (no new user have been recruited after attacks).

	1	2	3	4
1	1.	0.24	0.40	0.46
2	0.04	1.	0.25	0.25
3	0.04	0.14	1.	0.26
4	0.05	0.15	0.28	1.

is lightly skewed, with a mode of 4.1 s, a median of 11 s and a mean of 18 s. Unaccounted in the data are the errors which have never been solved, which represent 0.013% of the error and are therefore negligible. Thus, the data suggest that the crowd is able to respond to errors in at least two different ways with different characteristic times.

2.1.4 User engagement

As seen previously, the bulk of progress relies on the number and dedication of a few talented users. It is therefore important to attract and retain such users. We report the proportions of shared users across pairs of puzzles in Figure 6. Given the sequential solving of the puzzles, we can quantify the attrition rate, which is about 75% for all puzzles except Puzzle 1. The reason behind the peculiarity of Puzzle 1 may lie in the fact that only a few users participated, and that these users were likely more tightly connected to the social network of the UCSD team.

Figure 3 also shows that Puzzle 3 has about twice as many users as Puzzle 2, but has actually the same number of performing users. This indicates that despite the increase in crowd size, the number of people willing to participate is small and increases more slowly than the number of people visiting the site. The rarity of talented users, combined with a high attrition rate, means it is necessary to keep new users coming as previous users leave.

2.2 Crowdsourcing under attack

Our analysis reveals that the attacks sustained by the UCSD varied in scale, duration and nature. The attackers adapted to the defensive behavior exhibited by genuine users and the UCSD team. We were able to identify five different attack intervals. Each interval corresponds to one or more attackers attacking either once or several times, giving short breaks to genuine users before attacking again. To compare crowd behavior under the normal and attack regimes, we define a set of time series. Moreover, in order to help identify attackers, videos replaying the moves of all users have been produced (see Additional files 1 and 2). We refer the reader to Section 4 for more details.

Figure 7 summarizes the identified attacks and their diversity characteristics. The duration of the attacks ranged from 3 minutes to 1.5 hours, and the scale varied between less than 200 moves to more than 5,000. The average moves-per-second (MPS) of a period is the ratio of number of moves by the crowd during this period and the length of this period (we similarly define the user MPS as the number of moves s/he performed divided by participation span). The MPS of the attacks ranges from 0.1 to more than 10. High MPS is a combination of at least three factors: fast motivated attackers, parallel attackers and imperfect times-tamp collection.

We can distinguish at least two attack mechanisms: scattering the pieces of the correctly assembled cluster, and piling pieces on top of each over. Two attacks, a2 and a5, use an unknown mode of operation, which may correspond to the exploitation of a software bug, as claimed by the attackers.

attack	span	attack moves	non attack moves	attack MPS	attack type
a1.1	491	88	167	0.179	scattering
a1.2	2825	349	392	0.123	piling
a2	14	100	19	7.142	unknown
a3	133	190	51	1.428	piling
a4.1	1000	5618	594	5.618	piling
a4.2	133	1775	60	13.345	piling
a5	376	658	110	1.750	unknown

Figure 7 Attacks statistics and type. Attacks a1 and a4 proceeded in two phases and as such are considered separately in this table, labeled .1 and .2. Reported are the *span*, the duration of each phase of an attack (in seconds), the total number of *attack moves* and *non attack moves*, the average moves per second (*MPS*) of attacks, the *type* of the attack, as it can be observed on the videos. Attacks of type *unknown* are perpetrated by reported attackers whose mode of operation is unfathomable. Note that total span of attack a4 is of 72 min, consisting of 19 min of actual attacks and 54 min of recovery between the two phases.

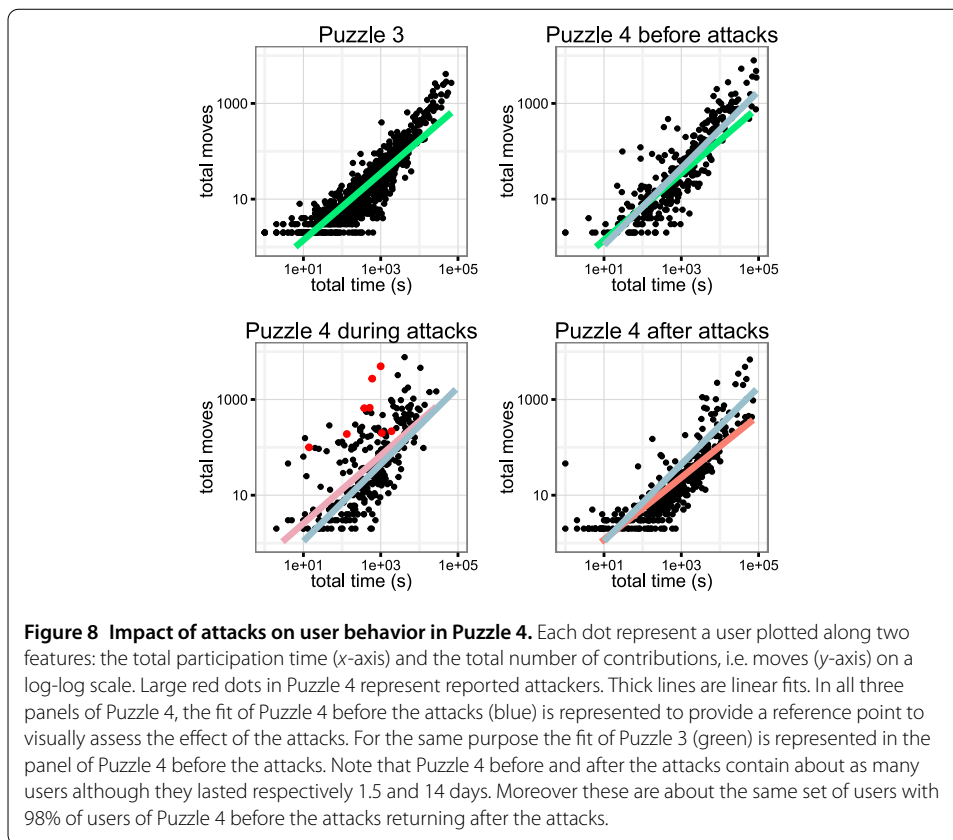
2.2.1 Unrolling of attacks

The first attack happened about two days after the start of Puzzle 4 and proceeded by scattering the pieces of the largest assembled clusters (attack a1.1). However, the attacker quickly changed his strategy from scattering to piling (attack a1.2), as the other users detected and repaired the damage. By piling up the pieces, the attacker made the work of genuine users much more complex, as they had first to unstack the pieces and then search for correct matches. While dedicated users were able to counter the attack, albeit slowly (refer to Section 2.2.5), the system was put offline for 12 hours and the state of the board reverted to a previous safe state (first rollback). The team also reacted by banning the IP addresses and logins used by the attacker, which was only a temporary solution.

The next attack (attack a2) was very small in scale and its nature is dubious, as it did not have any impact on progress and does not show any piece displacement on the video. The two subsequent attacks (attacks a3, a4) were clear, involving only piling moves. Thanks to the small scale of a3, it did not have a lasting impact (we discuss this attack later in more detail). Attack a4 was the largest, and it disrupted the system for about two hours, which was then put offline for 8 more hours before being reverted again to a safe state (rollback 2).

Attackers adopted different strategies because the crowd was able to observe the unusual behavior and react accordingly by unpiling, unscattering and possibly reconnecting edges. This is reported by the attackers in the email they sent to the UCSD team. While we observed additional attacker activity after a4, examination of the video does not conclusively support the attackers' claims. This last attack, a5, was undetected at the time and no rollback was performed.

In their email, the attackers claimed to be part of a competing team and to have recruited a crowd of attackers via *Achan* and to have instructed them to disconnect assembled clusters and stack pieces on top of each other. Whether this recruitment was successful is hard to evaluate. Nevertheless, to the best of our knowledge, only two physical attackers perpetrated the attacks under 8 different log-in names. In the remainder of the paper we will not make this distinction and talk about each attacker log-in as a separate attacker.



2.2.2 Immediate impact of attacks on user behavior

The total duration of the attacks, from the first to the last move made by an attacker, spans only about two days. We assess the immediate impact of attacks by looking at the features of users during and after the attacks and comparing them to baseline behavior. This baseline is given by Puzzle 3 and Puzzle 4 before the attack.

In Figure 8 users are plotted along two behavioral features: the total time spent on the system and the total number of moves. Linear regression lines are plotted for each panel, along the baseline in Puzzle 4. Reported attackers are indicated by red dots.

Across all panels except Puzzle 4 during attacks, a relationship between total time spent and total number of moves is apparent. This relationship however varies across panels as the linear regression coefficient shows: 0.69 in Puzzle 3, a faster 0.82 in Puzzle 4 before the attacks and a slower 0.66 after the attacks, all with comparable intercepts.

The attacks have an immediate impact of slowing progress and changing the behavior of genuine users. The relation between time spent on the system and number of moves does not hold any more: a large proportion of users (about half), makes a large number of moves within a short period of time. The slope of the linear fit not only decreases to 0.70 but also the intercept is higher. The behavior of six out of the eight reported attackers departs from the baseline, but so does an even higher number of genuine users, compelled to match the speed of attackers to respond to the attacks. The two attackers which do not depart significantly from the baseline of behavior correspond to the 2 logins used during attack a1, which was short and small paced.

While users departing from the baseline behavior can be suspected of being attackers, this is unlikely. Figure 15 plots users of puzzles 3 and 4 along the user participation span

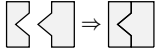
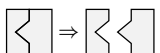
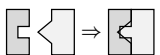
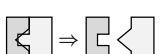
category	operation	visualisation	valuation
progress	create correct link		+1
progress	destroy correct link		-1
exploration	create incorrect link		-1
exploration	destroy incorrect link		+1

Figure 9 Move evaluation and time series. Moves are numerically evaluated with respect to their impact on the overall puzzle completion. Moves that involves two pieces that should be connected fall under the *progress* category, moves that involves other pieces fall under the *exploration* category. Progress moves have a direct impact over the completion, while exploration move have an indirect impact. A move is valued +1 if it creates a correct edges or destroy an incorrect edges, and -1 if it destroys a correct edge or create an incorrect edges. Given the time at which a move is performed, we obtain time series for each categories. We define an additional *completion* time series as the sum of the cumulative progress and exploration time series. More details can be found in Section 4.

and the number of exploration moved performed by the user. This combination of two kinds of features is able to highlight the pilling attackers, which clearly stand apart, with a high number of exploratory moves within a short time span.

2.2.3 Long term impact of attacks on crowd behavior

Attacks had a long lasting impact over the performance of the system, reducing its activity. It is noteworthy that after the attacks, no new users have been recruited, while 98% of the users before the attacks contributed at least once after the attacks. The same set of users are therefore being compared, ruling out difference that could arise from different user sets.

The long term impact of attacks can already be seen in Figure 8, when comparing the panel of Puzzle 4 before and after the attacks. The overall reactivity of users measured by the linear regression lines decrease from its highest before attacks at 0.82 to its lowest after attacks at 0.66, thereby requiring more time to achieve a given number of moves.

To further assess the long term impact of the attacks, we first define three discrete time series: progress, exploration and completion. These time series are based on a numerical evaluation of each move, which is given in Figure 9. Progress corresponds to the actual solving of the puzzle, exploration captures moves that do not directly lead to the creation of correct links. Completion is defined as the sum of the cumulative progress and exploration.

An illustration of the attacks unrolling is given in Figure 10. The figure represents the cumulative progress and exploration time series, highlighting the attacks and their direct impact. While none of these time series is sufficient by itself to capture different kinds of attacks, their combination is more informative. For this reasons we use the completion time series as the basis of our study of crowd behavior.

We now define three measures based on the set of local optima of the completion time series: the difference in time (Δ_{time}) and completion ($\Delta_{\text{completion}}$) between two successive optima, and the time to the same level after a local maxima ($\Delta_{\text{same level}}$) (for details see Fig-

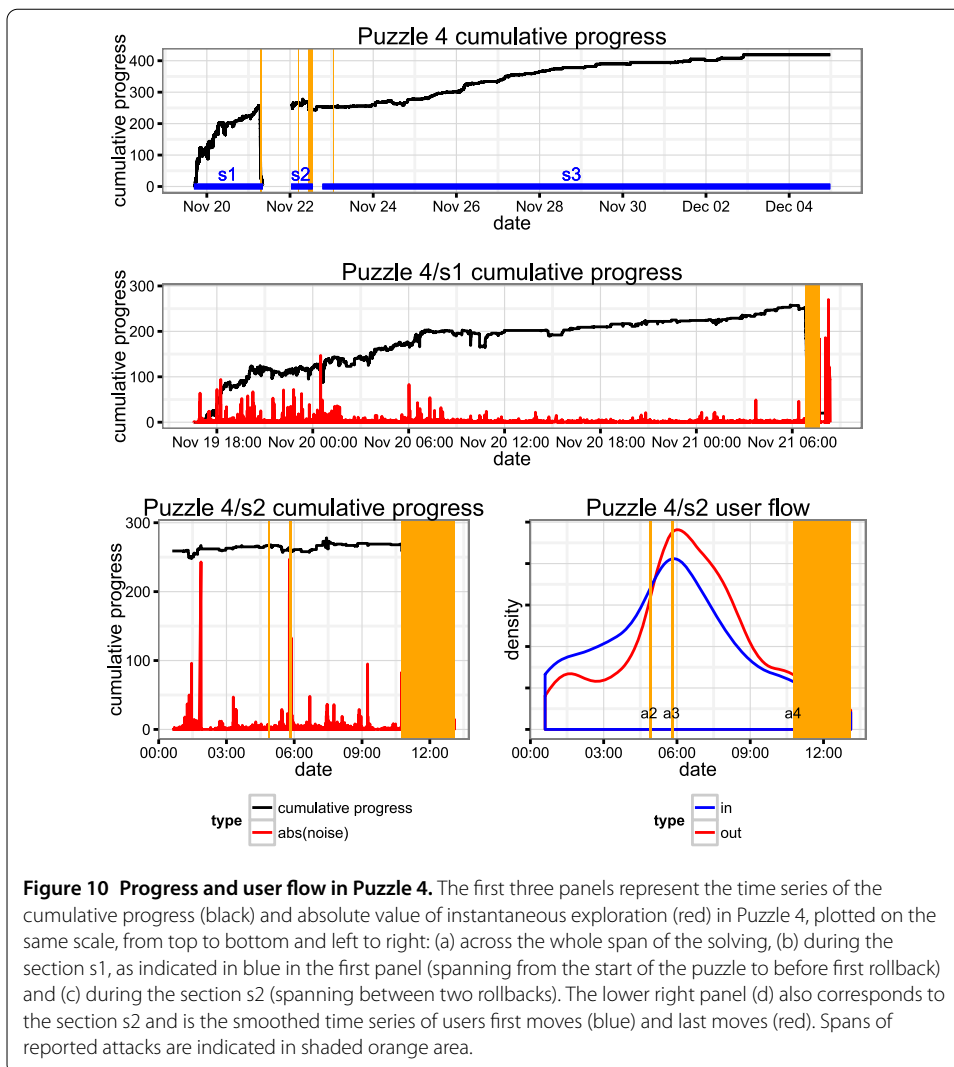
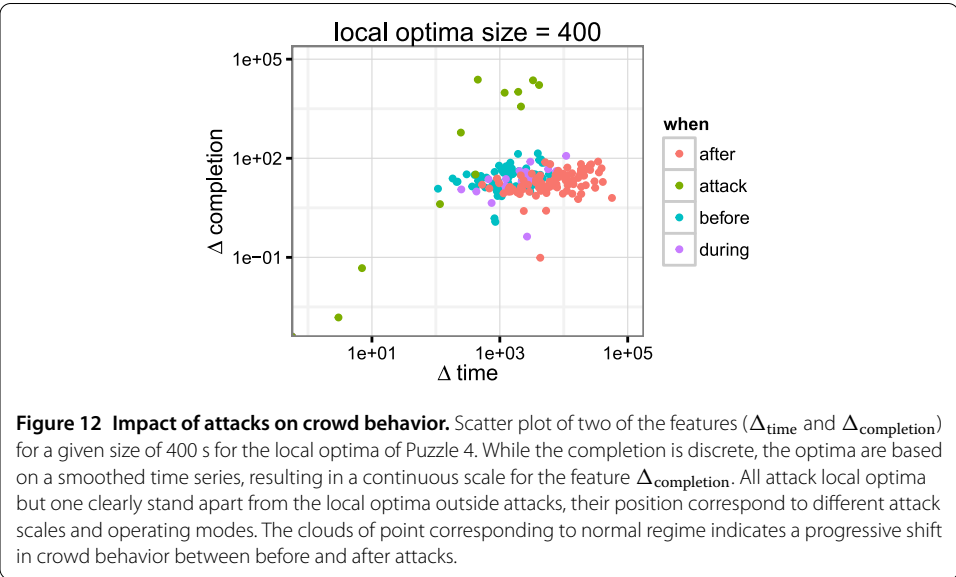
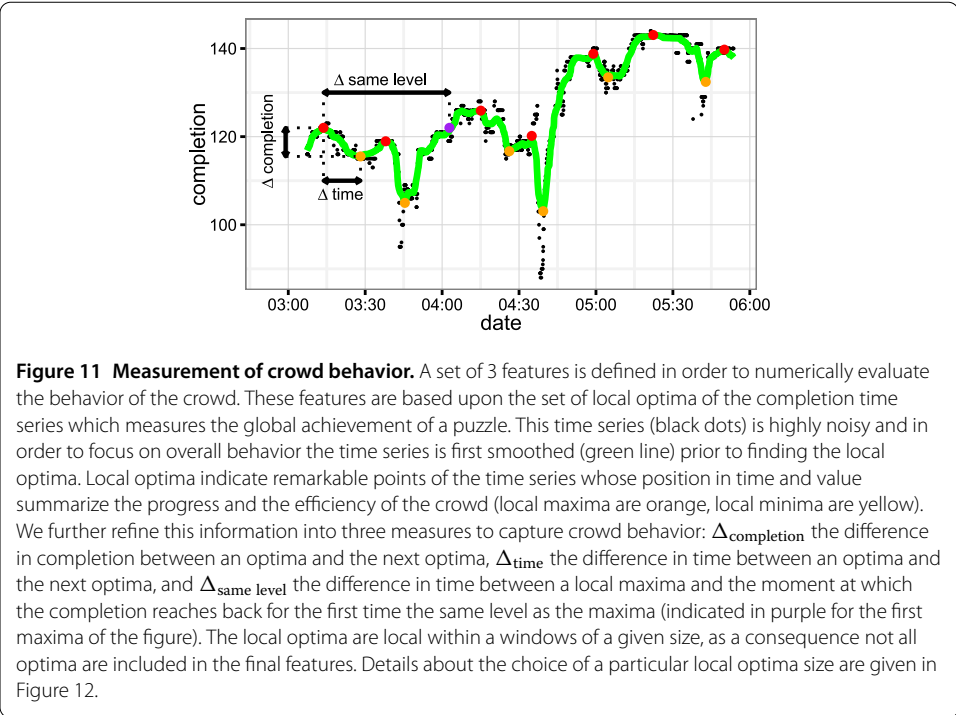


Figure 10 Progress and user flow in Puzzle 4. The first three panels represent the time series of the cumulative progress (black) and absolute value of instantaneous exploration (red) in Puzzle 4, plotted on the same scale, from top to bottom and left to right: (a) across the whole span of the solving, (b) during the section s1, as indicated in blue in the first panel (spanning from the start of the puzzle to before first rollback) and (c) during the section s2 (spanning between two rollbacks). The lower right panel (d) also corresponds to the section s2 and is the smoothed time series of users' first moves (blue) and last moves (red). Spans of reported attacks are indicated in shaded orange area.

ure 11). The two last measures capture the reactivity of the crowd, while the first captures its efficiency.

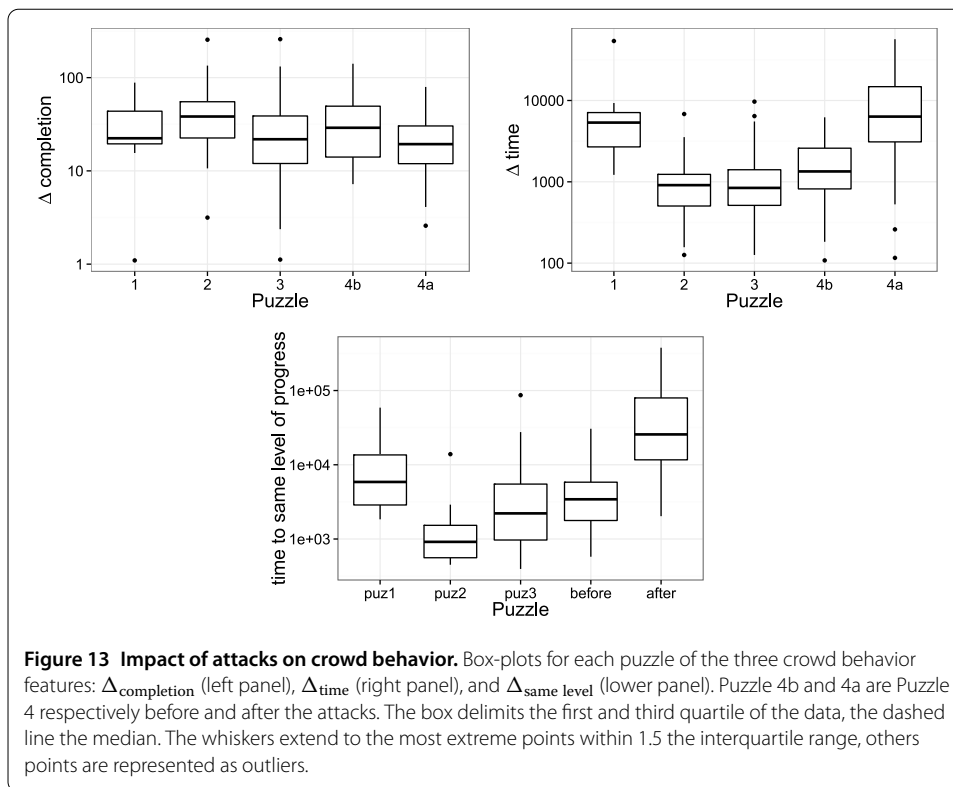
In order to rule out the impact of a particular choice of local optima size on the analysis, we computed the ratio before attack over after attack for each feature and for 13 values of local optima size ranging from 10 s to 1000 s. The ratios are remarkably stable, indicating that crowd behavior is invariant to the time resolution. Here we report their mean and standard error: $\Delta_{\text{completion}} = 1.438 \pm 0.105$, $\Delta_{\text{time}} = 0.192 \pm 0.003$, $\Delta_{\text{same level}} = 0.098 \pm 0.002$. We can already see that on average, after the attack, the crowd is about ten times slower to recover from a drop in completion, and that it is about 1.5 times less efficient. We further refine analysis of the crowd behavior by studying it in details for a given size of the local optima size of 400 s. See Section 4 for more details on this particular choice.

The local optima of Puzzle 4 are plotted in Figure 12 along two dimensions: Δ_{time} and $\Delta_{\text{completion}}$. Optima are partitioned along a qualitative variable indicating when the optima happened: before the attacks (blue), after the attacks (red), between the attacks but excluding them (purple), and during the attacks (green). This plot provides two key insights.



First, crowd behavior during attacks stands apart from the normal behavior. Second, there is a marked slowdown in crowd response time after the attacks.

This slowdown is better visualized in Figure 13, which shows box-plots of the three features of Puzzle 4 before and after the attacks, excluding the attacks themselves, compared to all the other Puzzles. The measure $\Delta_{\text{completion}}$, while being a lower bound on the actual variation of completion due to smoothing, is nevertheless comparable across all puzzles, implying that completion always varies by the same order of magnitude. There is however a significant variation across puzzles when considering the two other measures, related to the reactivity of the crowd.



When considering the box-plot of Δ_{time} , Puzzle 1 and Puzzle 4 after attacks clearly stand apart with a median difference in time magnitude slower than the baseline of the other puzzles. Specifically, Puzzle 3 has a mean of 18 min, which makes Puzzle 4 before attacks 1.83 times slower and Puzzle 4 after attacks 9.73 times slower. A likely explanation for the low reactivity in Puzzle 1 is because the crowdsourcing effort had just started and the platform did not have enough users (c.f. Figure 3), while Puzzle 4 after the attacks lost momentum due to the attack itself.

When considering the box-plot of $\Delta_{\text{same level}}$, Puzzle 1 and Puzzle 4 after attack stand apart even more clearly: Puzzle 3 and 4 before attacks have a mean time of about 1.5 hour, which is 11.53 times faster than Puzzle 4 after attacks. This order of magnitude slow-down indicates the drastic impact of attacks on crowd performance.

2.2.4 Distributed attacks

Despite the attackers' claim that they recruited a crowd of attackers on *4chan*, it is difficult to corroborate this claim from the data. Log analysis shows that all of the anonymous logins and IP addresses related to massive attacks are related to two real name emails. Moreover, at no time has there been more than two reported attackers acting simultaneously. We are thus led to conjecture that there were only two individuals behind all the known attacks.

We have already seen, in Figure 8, that during the attacks the behavior of attackers and genuine users gets entangled. Nevertheless, Figure 15 reassures us that we did not miss any important attacker as these clearly stand apart from the rest of the users.

However, this plot does not allow us to rule out the existence of several low scale attackers which would have a behavior similar to low performing genuine users. As matter

of fact, the bulk of participants contribute only a few moves before leaving the system. Attackers with such a behavior are impossible to detect as their profile is too similar to normal users. Indeed, the only two such reported attackers were detected thanks to the acquaintance network inferred from logs.

It is therefore not possible with these methods to detect a crowd of unrelated distributed attacker, which is our concern in this section. This is a matter of concern as the combined effect of a large crowd of several low scale attackers could do as much damage as one large scale attacker and would also be significantly harder to detect.

However, upon further inspection of the data, it is possible to find clues, pointing towards a distributed attack. The span between the first rollback and the start of the large attack a4 is the time during which the alleged crowdsourced attack is most likely to have happened. Incidentally, it is also during this span that the behavior of users deviates most from the norm. This change in behavior is characterized by a high number of pileups, high user MPS and a low total number of moves.

Additionally, the smoothed time series of first and last moves within this span, reported in Figure 10(d), shows a sharp influx and outflux of users peaking during attack a3. This indicates a massive wave of users joining and leaving immediately, which is what would be expected after an announcement on *4chan*. During this time, significant noise is added to the system while no significant progress is achieved. However, such is also the case during the beginning of Puzzle 4, which experienced a massive influx and outflux of users with similar consequences.

We are therefore not able to elucidate whether a *4chan* crowdsourced attack actually took place. We can argue that if it did, its impact was negligible; and if it did not, this would mean that the original attackers failed to recruit even a small crowd of attackers. In both cases, we can conclude that for a crowdsourced attack to work, it is necessary and sufficient for the participants to be motivated as it is illustrated by the very large damage that only few users were able to infringe. Effective attackers may have belonged to a rival team and were clearly motivated by the \$50,000 prize at stake. Such was not the case for attackers recruited through the Internet. They had no incentive to attack the UCSD platform as it was not involved in any controversial (e.g. political) activity, and they had no financial gain from doing so. In contrast, the UCSD team's strategy was altruistic, and promised to redistribute the potential prize wholly among the participants.

2.2.5 Attack recovery

Out of the five attacks the system sustained, two shattered all progress, and two had almost no effect. The two phases of attack a4 were separated by 54 minutes without attacks during which users, despite trying, were unable to recover from the first phase. Thanks to the rollbacks, users did not have to start from scratch, but this prevents us from adequately measuring how users recover from attacks.

We were nevertheless able to study the recovery of a smaller undetected attack, attack a3, which did not call for a rollback. This attack was not significant enough to be detected by the UCSD team but important enough to be detected by the crowd. We will therefore be able to describe crowd recovery only on this single attack. However, because of its small in scale, the conclusions we can draw from it are limited. The precise recovery capacity remains an open question that requires targeted experiments.

Attack a3 lasted about 133 seconds, and involved a unique attacker making 190 moves to disconnect and pile up pieces. This attack destroyed 88 correct edges. Recovery, defined

as the time to reach the same level of completion as before the attack, lasted 199 second and involved 628 moves from 16 unique users that recovered 86 of the 88 disconnected correct edges.

Interestingly, recreating what has been destroyed took genuine users about 1.5 times the duration it took to perform the destruction. However, it took them about 3 times more effort by 16 times more users. The attack MPS was 1.5 for the attackers against 3.15 for the defenders, which is significantly above the average MPS of 0.5 around this attack. This indicates that users identified an attack and deployed frantic effort to counter it.

Most of the users present during recovery were genuine and actively reacted to the attack as indicated by Gini index of their move count which is 0.35. This indicates a clear imbalance of power between attackers and defenders. Moreover, counting the number of moves as a proxy for the time individual users spent on recovery, we observe 0.23 man hours of work to defeat the damages inflicted in 0.03 man hours of works, i.e. an 8-fold increase in required effort.

Although the small scale of this particular attack made it easier to resist, it nonetheless illustrates the power imbalance that attackers have over genuine user. It can be expected that the time required to recover from an attack is super-linear in the number of pieces involved, while the destructive effort is linear.

2.2.6 Destructive effort

Rather than measuring the effort it takes to recover from an attack, we can also consider the efforts an attack needs to destroy what had been previously done. We also have a unique sample of this phenomenon: the effort it required the first attack to destroy most of the progress that was made since the beginning of puzzle 4. During this attack, all the assembled clusters were first scattered and then pieces were randomly piled on top of each others. After this attack the system was rolled back to a safe state.

Reaching the state of the puzzle before attack 1 required 39,299 moves by 342 users over 38 hours. Destroying all progress required 416 moves by one user and was done in 8 minutes of scattering and 47 minutes of piling. The user MPS of the attacker was 0.15, which is considerably slower than normal behavior, and much slower than the MPS of subsequent massive attacks. This can be explained due to the low reactivity of genuine users who contributed only a total of 473 moves by 18 users during the span of this attack and an average period MPS 0.13.

One attacker destroyed progress in about 1/100th of the moves and 1/60th of the time it took a crowd of several hundred users to create. The defenders were hapless. The imbalance of power is even starker in this sample.

3 Discussion

The question of how people organize to solve complex problems has always fascinated scientists. Of particular interest is the way incentives that shape individual behavior impacts the performance of the group [70–72], and how people are often able to overcome their self-interest to create sustainable communities [73]. However, it is often very difficult to observe a system of collective problem solving by observing every individual action. The data we collected from the DARPA Shredder Challenge provided us with unique opportunity to do just that.

As collective problem solving gets increasingly open to the public, it becomes more crucial to understand its resilience to errors and attacks [74]. Crowdsourcing is a powerful

technique for problem solving, yet highly vulnerable to disruption. The crowdsourcing approach proved itself highly effective, as in a matter of 4 days, the first three puzzles were solved. However, in the two remaining weeks the fourth puzzle, which was twice as large as the third puzzle, got completed only up to approximately 10%.

Crowd reaction to errors was efficient and seamless, with most errors recovered within minutes. However, the crowd was helpless against motivated attackers that could inflict damage far quicker than the crowd's recovery.

Long term impact of the attacks has been a sharp decrease in participation and activity. Previously active users were driven away and new users possibly discouraged. If no deadline is set for a crowdsourcing project, slowing down would not be an issue. However, such was not the case for the DARPA Shredder Challenge and each additional delay decreases the likelihood of winning, thereby reducing the motivation of users even further. Thus, the real impact of the attack was not to destroy the assembled pieces but to destroy the user base of the platform, and to disrupt the recruitment dynamics. We believe this is the main reason behind the inability of the UCSD team to win the challenge, which finished in the sixth position despite persistent large-scale attacks.

Openness is both the strength and the weakness of crowdsourcing. The power of crowdsourcing lies in allowing a large number of individuals to contribute freely by giving them unfettered access to the problem. However, because the power and the freedom given to the users are important, crowdsourcing stops working as soon as users start abusing this power. Such abuse can happen to any crowdsourcing project. However, large widely-publicized projects, which naturally attract more attention, or projects involving political or financial stakes, may easily find highly motivated opponents.

A number of safeguards have to be placed in crowdsourcing systems. Monitoring is the first component, as prompt measures have to be taken in case of attacks. Monitoring and behavior assessment are not trivial issues, and require the design of problem-specific measures for detecting attacks modes.

However, monitoring by itself cannot protect the system from damage. Actions such as banning users or IP addresses can help, but are inefficient against motivated, skilled, relentless, and adaptive attackers. Effective safeguards require constant (possibly automated) monitoring to detect unusual behavior and, more importantly, to lower the potential threat of attackers by temporarily closing parts of the system. Such measures could control or restrict one or several mechanisms, such as user recruitment, reach and actions.

It has to be noted that *Wikipedia*, arguably the largest open crowdsourcing platform in the world, is vulnerable to attacks. Such attacks, in the context of collaborative editing, can take the simple form of page defacing, or the more subtle biased rewriting. Page defacing can be easily detected by returning users and are easy to revert thanks to the edit history. To prevent small-scale attacks, a set of tools monitoring both the behavior and content of modification have been deployed to alert the crowd to suspicious updates. While users are not financially motivated in *Wikipedia*, they are often politically motivated to establish their point of view. *Wikipedia's* code of honor and neutral point of view policy does not prevent regular politically motivated attacks. In such a case the only effective measure is to lock down sensible page and only let reputable users contribute to them.

Another popular crowdsourcing platform, *Mechanical Turk*, differs in that users are paid to participate. Quality control is left to the task proposer, and a reputation mechanism is

used proposing higher paid tasks only to users that have proved their efficiency on lower paid ones.

No security solution can be fully automated and all ultimately rely on the supervision of the behavior of the system by the crowd or the organizers [34]. While this may seem to defeat the purpose of crowdsourcing, we believe that integrating security in crowdsourcing system design and investing resources in monitoring is fundamental. Any attacked crowdsourcing project risks not only converging to the wrong solutions but, more importantly, losing its core driving force: its user base.

Crowdsourcing can be viewed as a *public good game* in which a group of strangers get together to promote their collective well-being [75]. While these games create opportunities for individuals to free ride, thus leading to the *tragedy of the commons* [76], competitive crowdsourcing has the added dimension of inter-group competition [77–79]. Although our data is insufficient to study inter-team dynamics, capturing such data is possible in future experiments. Nevertheless, our analysis of the limited anonymous attacks provide a glimpse of what such analysis might look like. Simulation-based models can potentially be helpful in exploring these types of complex dynamics [80].

The recently proposed *crowdsourcing contest dilemma* game provides a starting point for studying inter-group conflict in crowdsourcing [48]. Moreover, game-theoretic analysis of incentives provides opportunities for designing mechanisms that encourage individuals to fix errors, as we have demonstrated elsewhere [34]. These incentives are non-trivial. For example, one can explicitly reward people who contribute to ‘error recovery,’ but this may create incentives to collude with attackers who create an artificial need for such recovery.

4 Methods and dataset

All data collection was conducted using our custom-built crowdsourcing platform. A dedicated web site includes further details about the system, recruitment etc. [81].

4.1 Data and features

4.1.1 Puzzle

The Shredder Challenge provided 5 puzzles of increasing complexity. The puzzles were the result of processing papers through document shredders. As a result, the pieces were tiny but had a regular size. All of these documents were handwritten text of increasing density and quantity, except for Puzzle 2 which was an image. It must be noted that the contestant were not required to solve all the puzzle to win the prize but had to answer specific questions. In order to do so, the answers had to found in part or totality of a document. Some artificial features like coffee mug marks, capital letters, different writing colors were introduced in the document to make solving the puzzles and answering the specific questions easier. As a consequence, the connection between puzzle time, solving time and effort is loose. Figure 2 is a screenshot of the virtual board provided by the UCSD team to the public. The figure also depicts the solution of Puzzle 2.

4.1.2 Dataset

The UCSD Shredder Challenge dataset, that will we refer to from now on, is simply the dataset with a log of all users’ actions on the web interface to solve the puzzles. The logging of these information proved itself invaluable during the challenge to rollback to safe

state before attacks and afterward study the behavior of the crowd and the reaction of the crowd in presence of disruption (this study). Among the data collected, two tables are of particular importance, the login table and the moves table. The login table is constituted of: email, IP address and time stamps. It was used to detect connected components of users and help in classifying users into two groups: genuine users and attackers. This table contains 7,725 entries. The move table consists of the following tuples: update_id, piece_id, piece position (3 coordinates: X , Y and rotation), email, IP address, time stamp. This table contains 290,479 entries made by 2,106 distinct users. From this raw data, we derived several measures and features that have been subsequently used to perform all the analysis presented in this paper.

We define a set of simple features that are associated with each move. As a result, we construct a time series for the resolution process and a set of simple features that are associated with each user. We also define complex measures for the crowd behavior through processing and aggregating the feature information in the time series. User features are also used to distinguish between attackers and genuine users, while the complex measures are able to distinguish most of the attacks from normal regime and allow us to quantify the impact of the attackers over the crowdsourcing process.

4.1.3 Features

We define several measures based on the dataset and from these we construct simple features that describe the progress of individual moves and the behavior of individual users. Also, we construct complex features that describe the aggregated behavior of the crowd.

These different features are used to understand the normal behavior of the system and accordingly to detect unusual behaviors that follow attacks and to assess their impact.

Progress. In order to capture how each move contributes to the solution of the puzzle, we define several progress measures associated with the move. Each of these measures is computed by comparing the neighboring pieces of the moved piece before and after the move to the actual neighboring pieces in the final state of the board when the puzzle is completed. More precisely, we count the progress related only to correct clusters of pieces that are hand identified in the final state of the board. We say that two pieces are placed correctly when they are linked in a final cluster, and incorrect otherwise. During the resolution, each piece can be linked and unlinked several times. We define the features CC (correct create) and CD (correct destroy) and associate them with the value of +1 or -1 when two correct pieces are respectively linked or unlinked. We similarly define IC (incorrect create) and ID (incorrect destroy) to refer to incorrect pieces and associate them respectively with a value of -1 or +1. See Figure 9 for illustration. The contribution made by correct edges is called progress and the contribution made by incorrect edges is called exploration or noise in the case of attacks. When a piece is piled over another one, that is to say two piece share the same location on the board. We discard the progress of both pieces if they are piled and count the move as a pile up.

Time series. In order to study the dynamic of puzzle solving process, we create several time series. Each time series correspond to the impact of moves as described in the previous paragraph and illustrated in Figure 9. Among these, we used three particular time series: The *progress* time series captures how many correct links between pieces are created and destroyed that corresponds to the actual solving of the puzzle. We are interested in the cumulative progress time series that reports the overall progression of the puzzle.

The *exploration* time series captures how incorrect links are created and destroyed. Exploration is necessary to test the matching of different pieces. It is reported in the figures using its absolute value to show the quantity of effort that is put into exploration. The actual value of exploration is null on average. The *completion* time series is defined as the sum of the cumulative progress and exploration. It is used to measure the crowd performance at solving puzzles. Time series of the number of first and last participation has been built in order to study the flux of users.

User behavior. In order to study the users' behaviors, we define a set of features which are computed based on the progress related measures and the timing of their moves. Users' contribution is irregular. They can participate only once but can also participate several times across the week, the day or the hour. As such, we distinguish continuous period of participation by setting a threshold of 15 min. If two moves are made during the threshold, they are counted as being part of the same participation. Otherwise, they belong to two different participations. Once we define the participations of a user, we define also: the sum of moves, sum of duration, mean of moves and mean of duration and mean of moves per second (average user MPS over each participation). We also include the span of users' participation and the time of their first and last move.

Crowd behavior. Crowd behavior is the combined effect of the action of different users. To study it, we define a set of complex features that are all based on the local optima of the completion time series. More precisely, we are interested in measuring the crowd reactivity and its work quality. For this purpose, we define three features: the time between two successive local optima (Δ_{time}), their difference in completion ($\Delta_{\text{completion}}$) and the time between a local maxima and the moment when the completion reaches again the same level as the local maxima for the first time ($\Delta_{\text{same level}}$). An illustration of their computation are given in Figure 11. We define the MPS as the number of moves performed by the crowd during a given period divided by its length.

Because the completion time series is highly noisy, a smoothing is performed in order to remove the most important instantaneous fluctuation with a kaiser window of size 50 and parameter beta set to 1. The corresponding smoothing time span is a function of the MPS which varies across several order of magnitudes. The parameters have been manually selected in order to remain close to the original time series while reducing the number of local optima due to instantaneous fluctuations, thereby focusing on the global behavior of the crowd. A side effect of the smoothing is that the $\Delta_{\text{completion}}$ is a lower bound of the actual variation of completion.

The impact of the choice of different local optima size over the values of the features is depicted in Figure 14. Small optima sizes can exhibit changes in the time series at an arbitrary small scale, while large optima size are better able to characterize the global behavior of the crowd. Larger optima size makes also the attacks stand out more clearly.

While the actual value of the features for each size is different, the ratio of the values, as reported in Section 2.2.3, is almost constant. This allow us to choose any particular value of local optima size for a detailed study of crowd behavior. We select a local optima size of 400 s to focus more on the global behavior of the crowd and because it is the smallest value for which attacks become clearly separated from normal behavior.

4.1.4 Data cleaning

During the resolution of puzzle 4, two rollbacks happened to take the system back to a previous safe state. The averaging of the cumulative progress related time series is required to

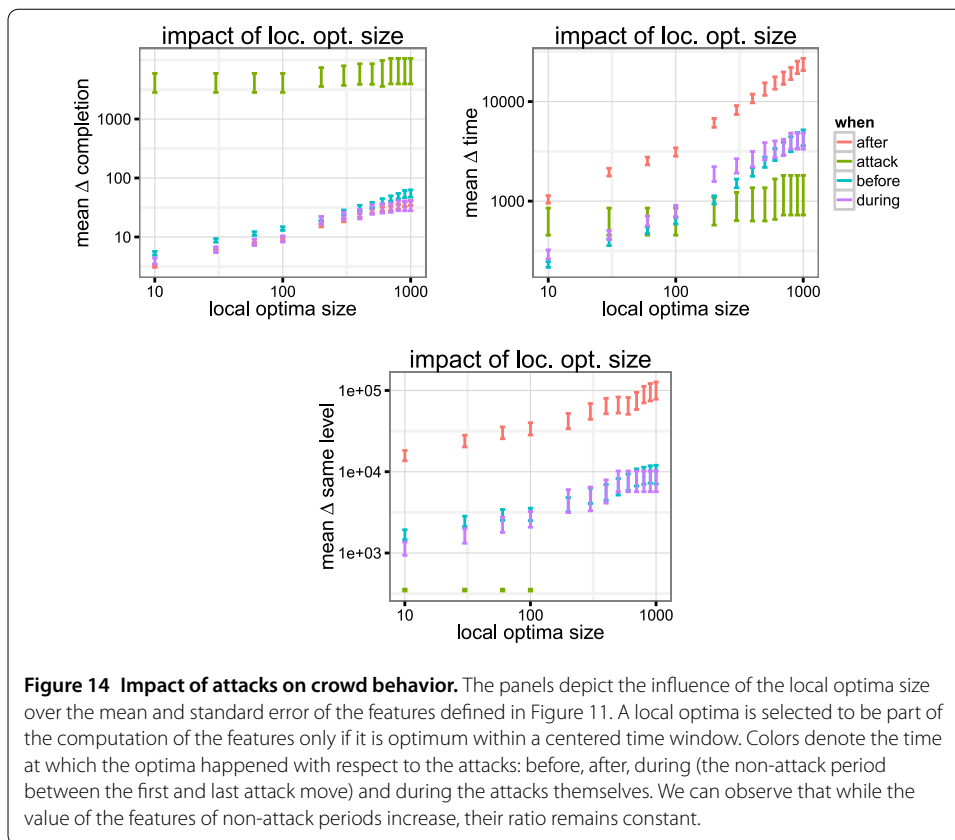


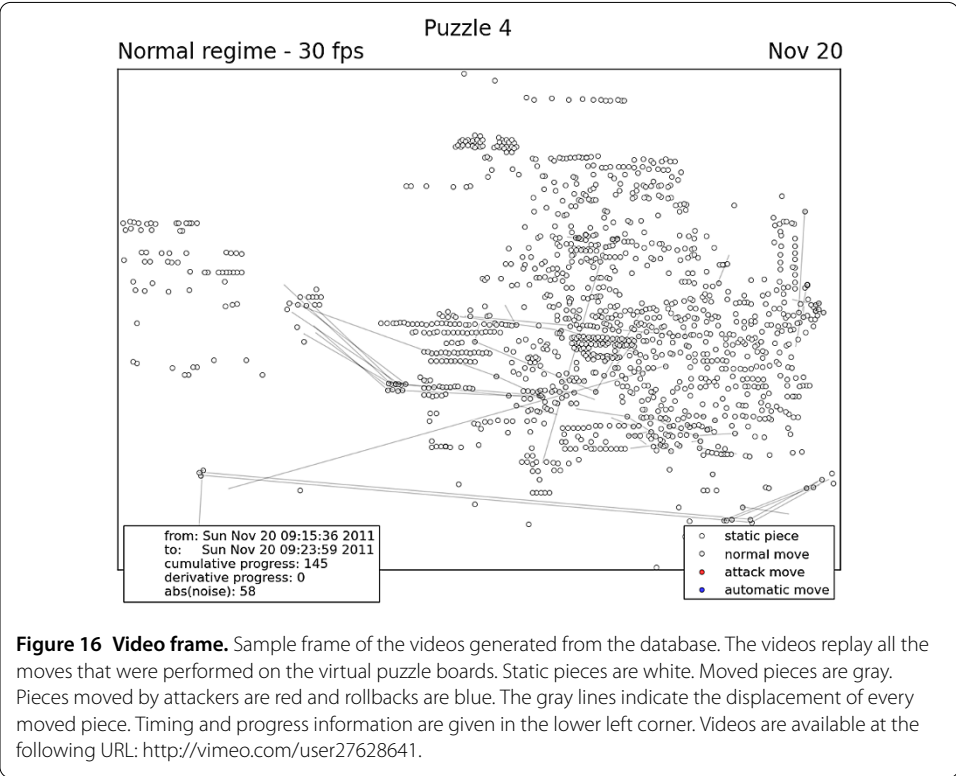
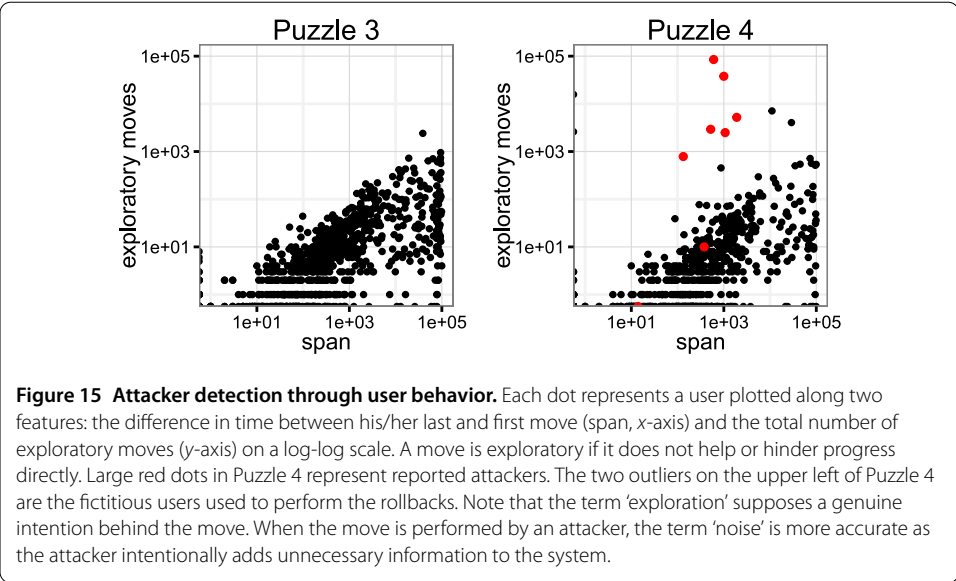
Figure 14 Impact of attacks on crowd behavior. The panels depict the influence of the local optima size over the mean and standard error of the features defined in Figure 11. A local optima is selected to be part of the computation of the features only if it is optimum within a centered time window. Colors denote the time at which the optima happened with respect to the attacks: before, after, during (the non-attack period between the first and last attack move) and during the attacks themselves. We can observe that while the value of the features of non-attack periods increase, their ratio remains constant.

take into account these time boundaries by considering them separate time series because of the sharp discontinuities. The computation of the complex features necessitated to take into account these same limits. This is the reason why for the time to same level feature, there are no points corresponding to attacks for local optima size larger than 200 s: the system never had the time to fully recover from important loss before rollback. This limits our ability to quantify how efficiently the crowd reacted to attacks. However among the several attacks withstood by the system, one was important enough to have an impact over the progress but not enough important to require a rollback as the crowd was able to seamlessly fully recover from it.

4.2 Attacker detection

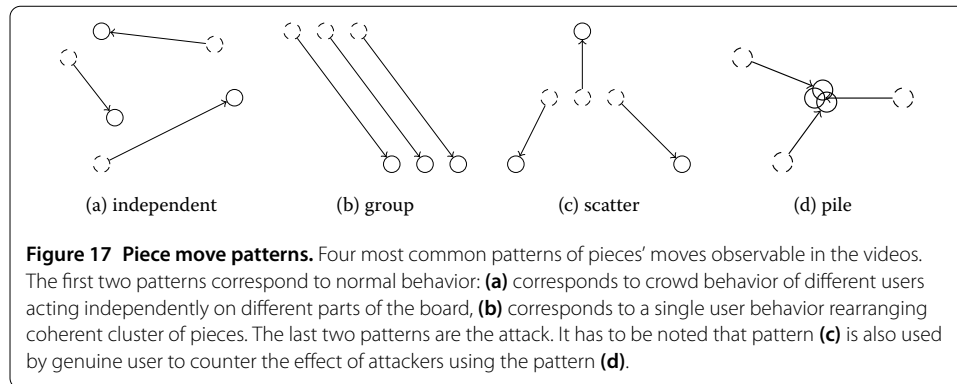
In order to detect attacks and attackers, we proceed in a layered way using increasingly sophisticated approaches. The first step is to use the database to perform a replay of each move of Puzzle 4 and manually identify suspicious sets of moves. In order to do so, we generate a video where each moved piece left a trail on the board between its initial position to its final position. Each frame of the video is comprised of hundred consecutive moves, thereby giving us a stepped global view of the crowdsourcing process. Figure 16 gives an instance of such a video frame.

In the following description, we will distinguish between genuine users, average users, reported attackers, and suspicious users. Genuine users are reported as such, if they fully satisfy at least one of the criterion of being genuine. Reported attackers shall fully satisfy at least one of the attack criterion to be an attacker. Suspicious users are users who take part only during the attacks and present similar traits as attackers but also as genuine users.



Average users are small or medium scale contributors who participate outside the attacks, while they can also do detrimental move. There is no reasonable reason to attribute them as being part of a wider coordinated attack. Most of the subsequent work has been done in order to classify suspicious users as either genuine users, attackers or average users.

Crowd behavior. Visual identification of attacks is what the crowd did at the time of the attacks. It requires high level knowledge about what suspicious move may look like. It is an ad-hoc approach to the problem and hard to automatize. We are able to identify on



the videos four patterns of crowd behavior as depicted in Figure 17: (1) independent users moving pieces uniformly across the board, (2) a user focusing over a cluster: arranging the precise location of its pieces or moving all of them to a new position, (3) scattering stable clusters of pieces to random positions (4) piling up pieces. The first two patterns of behavior were used solely by genuine users while we doubt about the last two patterns. The last two patterns are mostly used by reported attackers. Nevertheless they have been used by some genuine users as well.

Using this simple method, we are able to identify two of the four groups of attacks. The attacks of smaller scales are evading us. Given these large scale attacks, it is easy to identify the attackers' emails in the database by looking which email did massive participation at the same moment. However, attackers always acted while genuine users were presented in the system. The genuine users who spent a long time during solving the puzzle or participated a lot, were easy to tell apart from attackers. They can be identified by looking at their participation records, plotting user names along side the videos and identifying the ones corresponding to moves that could not have been done by attackers: unstacking and rearranging pieces. However, this method remains imprecise, yielding many suspected users, principally among the ones with medium or small scale participation.

Social network. Through the previous method, we are able to identify emails of 5 attackers. We then built from the login database, that provided IP-email relation, an acquaintance graph of users. Two users are accounted if they logged in at some point from the same IP or if there exists a path in the acquaintance graph between them. By using this method, we are able to identify several connected components of non trivial size (>4 users). Users with UCSD email addresses are connected to a dummy "UCSD" node in the graph. A user is considered genuine if the user belonged to the UCSD cluster or to a cluster that is reported as genuine.

The UCSD component contained the largest number of users. The second largest component of 18 users is the one to which all of the five identified attackers belong to. Among the newly identified users of this component, only 2 users made moves. Their naming pattern and disposable email address system that they used allow us to identify two more active attackers.

It has to be noted that a logical identity does not necessarily correspond to a physical identity. It can be suspected that behind these 18 email addresses, there were actually only 2 or 3 people because some emails were blacklisted in response to attacks during the challenge. At no time there was more than two reported attackers who were active simultaneously.

User features: progress related. In order to make sure that suspected attackers were actual attackers, we have to look into the user features. Progress-related measures give a good indication of attackers, even though they are not perfect. No single user feature that is considered alone can group attackers together. It is remarkably also the case for progress. However, this feature as well as exploration and the number of pile ups can give a loose indicator.

By grouping these features together, we get much better classification rules. However, there is no combination of rules that is able to classify perfectly the reported attackers. Considering the sum of the cumulative progress and the exploration, it is possible to get a much better decision rule as 6 out of the 8 attackers that performed moves clearly stand apart in one group. One of these attackers made a negligible number of contributions, but the other was the attacker removing pieces and totally evaded this mode of detection as its impact over the progress was null.

Having a decision rule that is learned on the reported attacker would allow us to filter the database and automatically classify users as attackers or genuine users. However, it is impossible to perfectly classify the few reported attackers, since attacks employed different behaviors in reaction to the defensive strategies exhibited by the crowd. Therefore attacks had different impact on progress measures. Scattering destroyed progress but generated no noise; pilling had a low impact over progress while generating considerable noise; and piece removal was transparent on any progress related measure.

User features: behavior related. For each kind of attack, we have to find the suitable set of features that is able to isolate attackers. However, no such choice makes attackers perfectly separable from the rest of the users and each one of them would detect the set of suspicious users. When looking at progress related measures, 3 users looked strongly suspicious. However, looking at behavior related features was possible to classify them as genuine user as we explain as follows.

By behavioral features, we refer to non-problem specific measures which are more generic features like the span of participation, number of contribution and average contribution length. In Figure 8, each user represents a point and is plotted according to the total number of moves he/she performed and the total number of seconds he/she participated. The color indicates the type of the users: black dots denote genuine users and red dots reported attackers. All suspicious users, when looking at their progress feature, had a behavior totally in par with the one of genuine users. It has to be noted that the two attackers in the middle of the genuine users correspond to the two emails sent during the first attack which was very slow paced. All subsequent reported attacks had a much faster pace, making them stand clearly apart.

As can be observed, we can disambiguate some users. However, when we look at the progress of some other unsuspected users, they become very suspicious as their behavior is close to the one of attackers. However, telling apart attackers and genuine users is made more difficult by the behavior of some genuine users to respond quickly to the fast random moves of the attacks. During the attacks, genuine users respond by making similar quick random moves and thereby acquired traits, high number of moves in short time span and high number of pile ups, that were originally characteristics of attackers. Figure 15 which mixes behavioral and non-behavioral features reassures us that we did not miss any important attacker. However, they can be used to rule out the existence of other low scale attackers. Looking at the video and connected clusters of pieces made it possible to re-

move the ambiguity for half of the suspicious users but not all of them. They might be part of a crowdsourced attack or not.

Additional material

Additional file 1: Video of Puzzle 2 (successful completion). Video of every move made by the crowd in solving puzzle 2. Each shred is shown as a circle (content is not shown to simplify the visualization).
Additional file 2: Video of Puzzle 4 (crowd response to sabotage). Video of every move made by the crowd in solving puzzle 4. Recognized attacks are indicated and slowed down.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

MC collected and preprocessed the data. MC and IR designed the research. NS and AA performed the numerical calculations. MC and IR analyzed the results. IR was the lead writer of the paper. NS, AA, MC and IR wrote the paper.

Author details

¹Masdar Institute of Science and Technology, Abu Dhabi, 54224, UAE. ²School of Informatics, University of Edinburgh, Edinburgh, UK. ³National Information and Communications Technology Australia, University of Melbourne, Melbourne, Victoria 3010, Australia. ⁴Department of Computer Science and Engineering, University of California at San Diego, La Jolla, CA 92093, California, USA.

Acknowledgements

We would like to thank Karyn Benson, Andrea Vattani and Daniel Ricketts for collecting and preprocessing the behavioral dataset. MC is supported by NICTA, which is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

Received: 13 March 2014 Accepted: 26 August 2014 Published online: 30 September 2014

References

1. Howe, J (2006) The rise of crowdsourcing. *Wired Magazine* 14(6)
2. Hand E (2010) Citizen science: people power. *Nature* 466(7307):685-687
3. von Ahn L, Maurer B, McMillen C, Abraham D, Blum M (2008) reCAPTCHA: human-based character recognition via web security measures. *Science* 321(5895):1465-1468
4. von Ahn L (2006) Games with a purpose. *Computer* 39(6):92-94
5. Salesses P, Schechtner K, Hidalgo CA (2013) The collaborative image of the city: mapping the inequality of urban perception. *PLoS ONE* 8(7):e68400
6. Horowitz D, Kamvar SD (2010) The anatomy of a large-scale social search engine. In: *Proceedings of the 19th international conference on world wide web*. ACM, New York, pp 431-440
7. Huberman BA, Romero DM, Wu F (2009) Crowdsourcing, attention and productivity. *J Inf Sci* 35(6):758-765
8. Cooper S, Khatib F, Treuille A, Barbero J, Lee J, Beenen M, Leaver-Fay A, Baker D, Popović Z, Players F (2010) Predicting protein structures with a multiplayer online game. *Nature* 466(7307):756-760
9. Mason W, Suri S (2012) Conducting behavioral research on Amazon's Mechanical Turk. *Behav Res Methods* 44(1):1-23
10. Hellerstein JM, Tennenhouse DL (2011) Searching for Jim Gray: a technical overview. *Commun ACM* 54(7):77-87
11. Zhang H, Horvitz E, Chen Y, Parkes DC (2012) Task routing for prediction tasks. In: *Proceedings of the 11th international conference on autonomous agents and multiagent systems - volume 2*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, pp 889-896
12. Alstott J, Madnick SE, Velu C (2013) Predictors of social mobilization speed. *CoRR abs/1303.3805*
13. Barrington L, Turnbull D, Lanckriet G (2012) Game-powered machine learning. *Proc Natl Acad Sci USA* 109(17):6411-6416
14. Rahwan I, Dsouza S, Rutherford A, Naroditskiy V, McInerney J, Venanzi M, Jennings NR, Cebrian M (2013) Global manhunt pushes the limits of social mobilization. *Computer* 46(4):68-75
15. Pickard G, Pan W, Rahwan I, Cebrian M, Crane R, Madan A, Pentland A (2011) Time-critical social mobilization. *Science* 334(6055):509-512
16. Bernstein MS, Little G, Miller RC, Hartmann B, Ackerman MS, Karger DR, Crowell D, Panovich K (2010) Soylent: a word processor with a crowd inside. In: *Proceedings of the 23rd annual ACM symposium on user interface software and technology*, pp 313-322. ACM
17. Surowiecki J (2004) The wisdom of crowds: why the many are smarter than the few and how collective wisdom shapes business, economies, societies and nations.
18. Mason W, Watts DJ (2010) Financial incentives and the performance of crowds. *ACM SIGKDD Explor News* 11(2):100-108
19. Tran-Thanh L, Venanzi M, Rogers A, Jennings NR (2013) Efficient budget allocation with accuracy guarantees for crowdsourcing classification tasks. In: *Proceedings of the 2013 international conference on autonomous agents and multi-agent systems*, pp 901-908. International Foundation for Autonomous Agents and Multiagent Systems
20. Karger DR, Oh S, Shah D (2011) Budget-optimal task allocation for reliable crowdsourcing systems. *CoRR abs/1110.3564*

21. Bernstein MS, Karger DR, Miller RC, Brandt J (2012) Analytic methods for optimizing realtime crowdsourcing. CoRR abs/1204.2995
22. Ipeirotis P (2011) Crowdsourcing using mechanical turk: quality management and scalability. In: Proceedings of the 8th international workshop on information integration on the web: in conjunction with WWW 2011. IJWeb '11. ACM, New York
23. Khatib F, Cooper S, Tyka MD, Xu K, Makedon I, Popović Z, Baker D, Players F (2011) Algorithm discovery by protein folding game players. *Proc Natl Acad Sci USA* 108(47):18949-18953
24. Meier P (2013) Human computation for disaster response. In: Handbook of human computation. Springer, Berlin, pp 95-104
25. Lorenz J, Rauhut H, Schweitzer F, Helbing D (2011) How social influence can undermine the wisdom of crowd effect. *Proc Natl Acad Sci USA* 108(22):9020-9025
26. Kleinberg J, Raghavan P (2005) Query incentive networks. In: Proceedings of the 46th annual IEEE symposium on foundations of computer science. IEEE Computer Society, Washington, pp 132-141
27. Kittur A, Nickerson JV, Bernstein M, Gerber E, Shaw A, Zimmerman J, Lease M, Horton J (2013) The future of crowd work. In: Proceedings of the 2013 conference on computer supported cooperative work. ACM, New York, pp 1301-1318
28. Zhang H, Horvitz E, Miller RC, Parkes DC (2011) Crowdsourcing general computation. In: Proceedings of the CHI 2011 workshop on crowdsourcing and human computation
29. Mao A, Parkes DC, Procaccia AD, Zhang H (2011) Human computation and multiagent systems: an algorithmic perspective. In: Proceedings of the twenty-fifth AAAI conference on artificial intelligence
30. Kamar E, Horvitz E (2012) Incentives for truthful reporting in crowdsourcing. In: Proceedings of the 11th international conference on autonomous agents and multiagent systems - volume 3. International Foundation for Autonomous Agents and Multiagent Systems, Richland, pp 1329-1330
31. Ipeirotis PG, Paritosh PK (2011) Managing crowdsourced human computation: a tutorial. In: Proceedings of the 20th international conference companion on world wide web, pp 287-288
32. Pfeiffer T, Gao XA, Mao A, Chen Y, Rand DG (2012) Adaptive polling for information aggregation. In: Association for the advancement of artificial intelligence
33. Mason W, Watts DJ (2012) Collaborative learning in networks. *Proc Natl Acad Sci USA* 109(3):764-769
34. Naroditskiy V, Rahwan I, Cebrian M, Jennings NR (2012) Verification in referral-based crowdsourcing. *PLoS ONE* 7(10):e45924
35. Cebrian M, Coviello L, Vattani A, Voulgaris P (2012) Finding red balloons with split contracts: robustness to individuals' selfishness. In: Proceedings of the 44th symposium on theory of computing, pp 775-788. ACM
36. Venanzi M, Rogers A, Jennings NR (2013) Trust-based fusion of untrustworthy information in crowdsourcing applications. In: Proceedings of the 2013 international conference on autonomous agents and multi-agent systems, pp 829-836. International Foundation for Autonomous Agents and Multiagent Systems
37. Babaiöff M, Dobzinski S, Oren S, Zohar A (2012) On bitcoin and red balloons. In: Proceedings of the 13th ACM conference on electronic commerce, pp 56-73.
38. Woolley AW, Chabris CF, Pentland A, Hashmi N, Malone TW (2010) Evidence for a collective intelligence factor in the performance of human groups. *Science* 330(6004):686-688
39. Pentland A (2012) The new science of building great teams. *Harv Bus Rev* 90(4):60-69
40. Drucker F, Fleischer L (2012) Simpler sybil-proof mechanisms for multi-level marketing. In: ACM conference on electronic commerce. ACM, New York, pp 441-458
41. Anderson A, Huttenlocher D, Kleinberg J, Leskovec J (2013) Steering user behavior with badges. In: Proceedings of the 22nd international conference on world wide web. WWW '13. pp 95-106. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland
42. Nath S, Dayama P, Garg D, Narahari Y, Zou JY (2012) Threats and trade-offs in resource critical crowdsourcing tasks over networks. In: AAAI
43. Chitnis R, Hajiaghayi M, Katz J, Mukherjee K (2013) A game-theoretic model motivated by the DARPA network challenge. In: Proceedings of the twenty-fifth annual ACM symposium on parallelism in algorithms and architectures. ACM, New York, pp 115-118
44. Vempaty A, Varshney LR, Varshney PK (2013) Reliable crowdsourcing for multi-class labeling using coding theory. CoRR abs/1309.3330
45. Viegas FB, Wattenberg M, Kriss J, Van Ham F (2007) Talk before you type: coordination in Wikipedia. In: System sciences, 2007. HICSS 2007. 40th annual Hawaii international conference on, pp 78. IEEE
46. Quinn AJ, Bederson BB (2011) Human computation: a survey and taxonomy of a growing field. In: Proceedings of the SIGCHI conference on human factors in computing systems, pp 1403-1412. ACM
47. Naroditskiy V, Jennings NR, Van Hentenryck P, Cebrian M (2014) Crowdsourcing contest dilemma. *J R Soc Interface* 11:0532
48. Oishi K, Cebrian M, Abeliuk A, Masuda N (2014) Iterated crowdsourcing dilemma game. *Sci Rep* 4:4100
49. Tang JC, Cebrian M, Giacobe NA, Kim H-W, Kim T, Wickert DB (2011) Reflecting on the DARPA red balloon challenge. *Commun ACM* 54(4):78-85
50. Rutherford A, Cebrian M, Rahwan I, Dsouza S, McInerney J, Naroditskiy V, Venanzi M, Jennings NR, deLara JR, Wahlstedt E et al (2013) Targeted social mobilisation in a global manhunt. *PLoS ONE* 8:e74628
51. Cebrian M, Torres MR, Huerta R, Fowler JH (2013) Violent extremist group ecologies under stress. *Sci Rep* 3:1544
52. Zhang R, Zhang J, Zhang Y, Zhang C (2013) Secure crowdsourcing-based cooperative spectrum sensing. In: INFOCOM, 2013 proceedings IEEE, pp 2526-2534. IEEE
53. Ghosh A, Kale S, McAfee P (2011) Who moderates the moderators?: crowdsourcing abuse detection in user-generated content. In: Proceedings of the 12th ACM conference on electronic commerce, pp 167-176. ACM
54. Watts D, Cebrian M, Elliot M (2013) Dynamics of social media Public response to alerts and warnings using social media: report of a workshop on current knowledge and research gaps. The National Academies Press, Washington
55. Palmer C (December 2011) UC San Diego team's effort in DARPA's Shredder Challenge derailed by sabotage. Technical report, California Institute for Telecommunications and Information Technology Press Release, East Lansing, Michigan. <http://calit2.net/newsroom/article.php?id=1938>

56. Lesk M (2007) The new front line: Estonia under cyberassault. *IEEE Secur Priv* 5(4):76-79
57. Zuckerman E, Roberts H, McGrady R, York J, Palfrey J (2010) Distributed denial of service attacks against independent media and human rights sites. The Berkman Center
58. Pras A, Sperotto A, Moura G, Drago I, Barbosa R, Sadre R, Schmidt R, Hofstede R (2010) Attacks by "Anonymous" WikiLeaks proponents not anonymous
59. DARPA Shredder Challenge. <http://archive.darpa.mil/shredderchallenge>. Accessed 26 Dec 2013
60. Demaine ED, Demaine ML (2007) Jigsaw puzzles, edge matching, and polyomino packing: connections and complexity. *Graphs Comb* 23(1):195-208
61. Zhang H, Lai JK, Bächer M (2012) Hallucination: a mixed-initiative approach for efficient document reconstruction. In: Workshops at the twenty-sixth AAAI conference on artificial intelligence
62. Pentland A (2014) *Social physics: how ideas turn into actions*. The Penguin Press
63. Bernstein MS, Monroy-Hernández A, Harry D, André P, Panovich K, Vargas GG (2011) 4chan and/b: an analysis of anonymity and ephemerality in a large online community. In: ICWSM
64. Zuckerman E, Roberts H, McGrady R, York J, Palfrey JG (2010) 2010 report on distributed denial of service (DDoS) attacks. Berkman Center Research Publication (2010-16)
65. Resnick P, Kuwabara K, Zeckhauser R, Friedman E (2000) Reputation systems. *Commun ACM* 43(12):45-48
66. Lorenz MO (1905) Methods of measuring the concentration of wealth. *Publ. Am. Stat. Assoc.* 9(70):209-219
67. Gini C (1912) Variabilità e mutabilità. Reprinted in *Memorie di metodologica statistica* (Ed Pizetti E, Salvemini, T) Rome: Libreria Eredi Virgilio Veschi 1
68. Clauset A, Shalizi CR, Newman ME (2009) Power-law distributions in empirical data. *SIAM Rev* 51(4):661-703
69. Vuong QH (1989) Likelihood ratio tests for model selection and non-nested hypotheses. *Econometrica* 57:307-333
70. Olson M, Olson M (2009) The logic of collective action. *Public goods and the theory of groups*, vol 124. Harvard University Press, Harvard
71. Coleman JS (1994) *Foundations of social theory*. Harvard University Press, Harvard
72. Gintis H (2009) *The bounds of reason: game theory and the unification of the behavioral sciences*. Princeton University Press, Princeton
73. Ostrom E (1990) *Governing the commons: the evolution of institutions for collective action*. Cambridge university press, Cambridge
74. Albert R, Jeong H, Barabási A-L (2000) Error and attack tolerance of complex networks. *Nature* 406(6794):378-382
75. Fehr E, Gächter S (2000) Cooperation and punishment in public goods experiments. *Am Econ Rev* 90:980-994
76. Hardin G (1968) The tragedy of the commons. *Science* 162(3859):1243-1248
77. Rapoport A, Bornstein G (1987) Intergroup competition for the provision of binary public goods. *Psychol Rev* 94(3):291
78. Halevy N, Bornstein G, Sagiv L (2008) "In-group love" and "Out-group hate" as motives for individual participation in intergroup conflict a new game paradigm. *Psychol Sci* 19(4):405-411
79. Abbink K, Brandts J, Herrmann B, Orzen H (2010) Intergroup conflict and intra-group punishment in an experimental contest game. *Am Econ Rev* 100:420-447
80. Rutherford A, Cebrian M, Dsouza S, Moro E, Pentland A, Rahwan I (2013) Limits of social mobilization. *Proc Natl Acad Sci USA* 110(16):6281-6286
81. Team web site. <http://shredder-challenge.ucsd.edu/login.php>. Accessed 1 Sept 2014

doi:10.1140/epjds/s13688-014-0013-1

Cite this article as: Stefanovitch et al.: Error and attack tolerance of collective problem solving: The DARPA Shredder Challenge. *EPJ Data Science* 2014 2014:13.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com